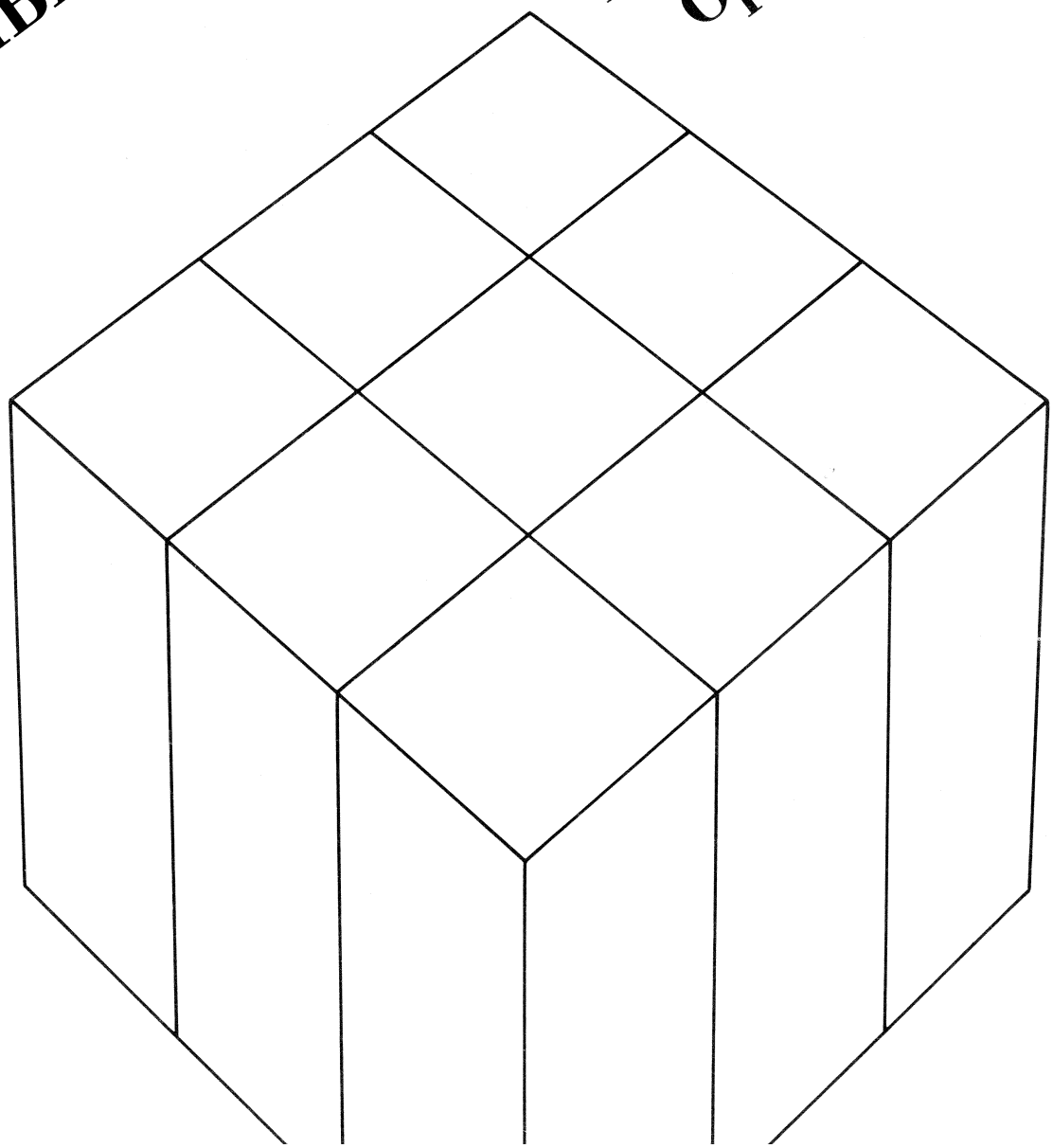


IBM Virtual Storage Extended POWER

Installation and Operations Guide



IBM Virtual Storage Extended/ POWER

Installation and Operations Guide

Version 2 Release 3

Program Number 5666-273

Order Number SH12-5329-06
File No. S370/4300-36

Seventh Edition (June 1987)

This edition is a major revision of SH12-5329-5. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the change. This edition applies to Version 2, Release 3 of IBM Virtual Storage Extended/POWER, Program Number 5666-273, and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370, 30XX and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this document is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

Publications are not stocked at the addresses given below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments is provided at the back of this publication. If the form has been removed, comments may be addressed either to:

IBM Corporation
Dept. 6R1BP
180 Kost Road
Mechanicsburg, PA 17055, USA

or to:

IBM Deutschland GmbH
Dept. 3164
Schoenaicher Strasse 220
D-7030 Boeblingen, Federal Republic of Germany

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1979, 1981, 1983, 1984, 1985, 1987

PREFACE

This manual describes the functions of IBM Virtual Storage Extended/POWER (VSE/POWER for short), a program that schedules the input and output of jobs. The manual guides you in installing and using this program product.

To use the manual for preparing jobs for processing under VSE/POWER, you should be familiar with the operational concepts of the VSE operating system.

To use the manual for installing VSE/POWER or for planning the product's installation, you should be familiar also with the procedures for installing VSE/Advanced Functions, the minimum operating system support in a VSE environment.

A central system operator using the manual should be familiar with the operating procedures as given in the IBM publication VSE/Advanced Functions, Operation.

Throughout this book:

Version and release numbers for VSE/POWER and various other products are shown in the form "n.n." For example, VSE/POWER Version 2, Release 3 is shown as VSE/POWER 2.3.

If not stated otherwise, information given for the IBM 3800 Printing Subsystem applies also to the IBM 3200 Printing Subsystem.

This book contains information for tasks as follows:

- Planning
  - Description and use of functions.
  - The hardware and software requirements for installing and using VSE/POWER; this includes the calculation of the required processor and virtual storage space.
  - Throughput considerations.
- Installation
  - Verification that your VSE supervisor includes the required support.
  - The generation of VSE/POWER control tables.
- Operation
  - How to start, control, and end VSE/POWER operation.
  - How to control the processing of jobs under VSE/POWER.

The description of the major user task application programming has been transferred to a new manual VSE/POWER Application Programming. That manual now contains the following sections, which were formerly part of the level-5 edition of the VSE/POWER Installation and Operations Guide:

- Job Accounting and Segmentation (formerly part of Chapter 5)
- Spool-Access Support (formerly Chapter 7)
- External Device Support (formerly Chapter 8)
- RJE,SNA Data Compaction (formerly Appendix A)
- Reader Exit Routines (formerly Appendix B)
- VSE/POWER SPOOL-Macro Support (formerly Appendix C)

This book is organized in chapters as follows:

Chapter 1 introduces VSE/POWER: how it works, what it can do, and the environment in which it operates.

Chapter 2 gives information about planning. Items of consideration are such things as machine requirements, disk storage for the VSE/POWER files, virtual and processor storage for the various VSE/POWER functions. The chapter includes the procedures you must follow to install VSE/POWER.

Chapter 3 gives examples for the day-to-day use of VSE/POWER as seen from the central system operator's point of view.

Chapter 4 describes the VSE/POWER central operator commands; you can use them for controlling the operation of VSE/POWER from your system's console.

Chapter 5 describes the VSE/POWER job entry control language (JECL) statements; you can use them for defining a job to VSE/POWER and how VSE/POWER is to process the unit-record input and output of the job.

Chapter 6 describes the available autostart statements.

Appendix A explains the purpose of disposition codes; it lists the conditions that cause VSE/POWER to change a disposition code.

Appendix B discusses the VSE/POWER function called data compaction. This function helps to reduce line traffic. You can activate the function as part of VSE/POWER's RJE,SNA support.

Appendix C explains the use of the disk dump program for VSE/POWER files.

At the end of this book, you find:

- A **Glossary** which explains terms used in this manual. For terms not explained in this manual, you may refer to the IBM manual Vocabulary for Data Processing, Telecommunications, and Office Systems, GC20-1699.
- A **Bibliography** which lists related documentation.
- An **Index** to aid you in retrieving information from this manual.

CONTENTS

Chapter 1. Introduction to VSE/POWER 1
Generation of Control Tables 2
Supported Hardware 2
Concepts and Functions 4
Processing Unit-Record I/O Requests 11
System Considerations 12
Throughput Considerations 15
Conventions for Showing Command and Statement Formats 15

Chapter 2. Planning and Installation 19
Migration to this Release 19
Planning for VSE/POWER 21
Hardware Requirements 21
Disk Storage Requirements for VSE/POWER Files 22
Processor- and Virtual-Storage Requirements 29
The Options for Your VSE/POWER 38
Shared Spooling Requirements 38
ACF/VTAM- and ACF/NCP-Generation Requirements 39
Special Device Considerations for VSE/POWER 40
Classes for Processing and for Output 42
Installing VSE/POWER 43
Generating VSE/POWER Control Tables 44
The Generation Macros 45
POWER Generation Macro 47
PLINE Macro 60
PRMT Macro 63
PNODE Macro 72
PCPTAB Macro 76
Examples of VSE/POWER Generation Macros 79
VSE/POWER Generation Procedure 81

Chapter 3. Operating With VSE/POWER 83
Starting VSE/POWER 83
Shared Spooling Considerations 85
Starting VSE/POWER with Autostart 87
Starting VSE/POWER without Autostart 92
Controlling Operation 96
Controlling Operation at the System Console 97
Specific Devices 109
VSE/POWER Support of the IBM 4248 109
Using Multifunction Unit-Record Devices 110
Using an IBM 3540 Diskette Unit 111
Tape Processing with VSE/POWER 114
Processing Spool Tapes 115
Processing for the SYSIN Tape Function 116
Processing for the Offload-Queue Function 117
Terminating VSE/POWER 119

Normal Shut-Down 120
Abnormal End of VSE/POWER 122
Emergency Termination 123

Chapter 4. VSE/POWER Central Operator Commands 125
PACCOUNT: Empty the VSE/POWER Account File 129
PACT: Activate a Transmitter/Receiver 131
PALTER: Alter Job Attributes or Partition Classes 132
PBRDCST: Send Messages 141
PCANCEL: Terminate a Status Display or Cancel a Job 144
PDELETE: Delete Jobs or Messages 146
PDISPLAY: Display VSE/POWER Status 151
PDRAIN: Deactivate (Drain) a Transmitter or Receiver Task . . 176
PEND: End Operation Under Control of VSE/POWER 178
PFLUSH: Cancel a Queue Entry 180
PGO: Reactivate a VSE/POWER Task 184
PHOLD: Place a VSE/POWER Job into the Hold or Leave State . . 187
PINQUIRE: Request Non-Local Status Information 192
PLOAD: Load a Network Definition Table 197
POFFLOAD: Save Queue Entries On Tape / Restore Queue Entries From Tape 199
PRELEASE: Remove VSE/POWER Jobs from Hold/Leave State 204
PRESET: Reset the Job Status 209
PRESTART: Restart Printed or Punched Output 211
PSETUP: Check the Forms Alignment of Printed Output 214
PSTART: Start a Task or a Partition 216
PSTOP: Stop a Task or Release a Partition 227
PXMIT: Route a Command 234

Chapter 5. VSE/POWER Job Entry Control Language Statements 237
Correction of JECL Errors 239
Format of JECL Statements 240
Overview of JECL Statements 241
* $$ CTL: Assign a Default Input Class 244
* $$ DATA: Insert Data into an SLI Library Member 245
* $$ EOJ: Mark the End of a VSE/POWER Job 248
* $$ JOB: Mark the Start of a VSE/POWER Job 249
* $$ LST: Define List-Output Attributes 256
* $$ PUN: Define Punch-Output Attributes 271
* $$ RDR: Adding 3540 Files to Input Jobs 280
* $$ SLI: Add a Library Member to a Job Stream 284

Chapter 6. VSE/POWER Autostart 291
VSE/POWER Autostart Statements 291
DEFINE: Specify User-Defined Output Operands 294
FORMAT: Specify File-Formatting Options 299
PRINTERS: Define Output-Spooling Devices for List Output . . . 301
PUNCHES: Define Output-Spooling Devices for Punch Output . . . 303
READER: Define an Input-Spool Device 305
SET: Set VSE/POWER Control Values 307

Appendix A. VSE/POWER Disposition Codes 309

Appendix B. RJE,SNA Data Compaction 315
Implementing Data Compaction 315
Method of Compaction 317

Appendix C. Disk Dump Program of VSE/POWER 323

Glossary 327

Bibliography 331

Index 333

FIGURES

1. Data Flow Under Control of VSE/POWER 5
2. Minimum Storage Requirements 14
3. Relationship between Queue File and Data File 24
4. Device Type - Queue Record Block Relationship 26
5. Default Sizes of Data Blocks 27
6. Overall Storage Requirements for VSE/POWER Partition . . . 30
7. The VSE/POWER Partition - Storage Requirements 33
8. GETVIS Calculation for RJE,SNA Support Control Blocks . . . 34
9. Values for Calculating VSE/POWER's Fixable and GETVIS Areas 36
10. Job Stream Example for Starting VSE/POWER 89
11. Example of a VSE/POWER Autostart Procedure 90
12. Example of the Statistics Part of a VSE/POWER Status Report After PEND 121
13. VSE/POWER Central Operator Commands 127
14. Examples of the Use of VSE/POWER JECL 238
15. Overview of JECL Statements 242
16. Use of * $$ DATA Statements 246
17. Extract from Keyword Definition Table 296
18. Overview of Disposition Assignments by VSE/POWER 310
19. Mapping of Conditions to Dispositions for Unsuccessful Completion 311
20. Master Character to Compact-Code Character Relationship . . 315
21. Example of a Compact-Code Character Set 316
22. VSE/POWER Compaction Tables 321
23. Control Statement Set for Disk Dump Program Invocation . . 324
24. The Format of a CKD Disk Dump 326

SUMMARY OF CHANGES

Following are the amendments to the publication for this and the preceding release of version 2 of VSE/POWER:

New for VSE/POWER 2.3

- Supported Hardware. IBM processors and devices newly supported by VSE/Advanced Functions 2.1 are supported also by this release of VSE/POWER.

- Spool File Restructure. The organization of the VSE/POWER data and queue file has been significantly improved to increase overall performance.

  A new queue file recovery technique will eliminate any chain errors, since VSE/POWER no longer follows the various class chains and tries to repair them. Instead, it examines all queue records using their sequential organization and rebuilds the various class chains and the free queue record chain.

  Each class chain is divided into a dispatchable and a non-dispatchable class chain. Queue entries with disposition K or D reside in the dispatchable class chain, while all other queue entries reside in the non-dispatchable class chain. This reduces the number of I/O operations necessary to locate the first eligible queue entry.

  A significant reduction of queue file I/O operations is achieved by placing the entire queue file in VIO space while VSE/POWER is active. The VIO space can be seen as an extension of the page data set and is used as system work space.

- Spool File Utilization. The utilization of the queue file is improved by blocking queue records and allowing only one queue record per queue entry. Also, the data file utilization does no longer depend on device characteristics such as track capacity, but can be tailored according to installation requirements (via the DBLK group size).

- Spool Space Enlargement. The number of data file extents has been increased from five to 15.

- Threshold Message. A threshold value, in terms of the percentage of spool space used, can be specified for efficient usage of the spool files. The system operator will be alerted via a message when either the queue file or the data file exceeds the specified value.

- IBM 3800/4248 Output Processing. Performance has been improved when processing the PRESTART command for an IBM 3800 or 4248 printer.

- User-Defined Output Operands. A new interface is provided that allows a user-written application program or an IBM program product to define its own, additional output operands. These output operands can be specified in the * $$ LST or PUN statement and will accompany the queue entry. The operands are transmitted in the network as Output Parameter Text Blocks (OPTBs) within the data set header record. The spool-access support allows to pass such OPTBs directly, together with an output entry, and later to retrieve and/or modify existing OPTBs.

- Queue Manipulation by Creation Date and User-ID. The queue manipulation commands (PALTER, PDELETE, PDISPLAY, PHOLD, and PRELEASE) have been extended to allow an installation to manipulate one or more queue entries based on the creation date or the specified user-ID. Thus, output queue entries can be deleted that have occupied the spool file for some time.

- Additional Spool Devices. The maximum number of output spool devices has been extended to 14 printers or punch devices.

- Job Number Increase. The highest job number that can be assigned to a VSE/POWER job has been increased from 32,767 to 65,535. This reduces the chance of assigning duplicate job numbers.

- POFFLOAD Improvements. The POFFLOAD command has been extended to allow selective reloading of one or more queue entries back to the VSE/POWER queues. Spool tapes (DISP=T) can also be reloaded.

  The PDISPLAY command has been extended to list information about each queue entry on a POFFLOAD tape or a spool tape.

- Additional Operands on * $$ JOB Statement. The * $$ JOB statement has been extended to include new ownership and distribution information such as:

  Programmer name
  Room number
  Building number
  Department number

- Separator Page Improvements. A sequence count, part of the separator page, has been introduced to aid the operator in separating the output produced by individual jobs.

  A new option on the PSTART writer task command allows the operator to either always obtain or always suppress separator pages or cards.

- Skip-to-Channel 1 Option. A new option on the PSTART writer task command allows the operator to insert a skip-to-channel 1 preceding the job output.

- PRESTART Improvement. The highest page/card count to be restarted has been extended from 9,999 to 65,535.

- Temporary Disposition Y. The spool-access support and the external-device support allow to specify a temporary disposition of Y to indicate that the processing of an output queue entry has failed. Disposition Y is also given to predefined entries created by the spool-access support when the communication is terminated abnormally during retrieval of such entries.

- Recovery of Partially Produced Output. A new operand (RBC) on the * $$ LST and PUN statements allows to take checkpoints during spooling of the output. If VSE/POWER terminates abnormally, the incomplete output entry will no longer be lost; VSE/POWER recovery will make it available up to the last recorded checkpoint and flag the entry with a temporary disposition of 'X'. The operator can query such queue entries with the PDISPLAY {LST|PUN},CDISP=X command and, with the PALTER DISP=* command, change the temporary disposition of X back to the queue entry's original disposition.

- Support of Printerless Systems. Dumps of the VSE/POWER partition or the VIO area at abnormal termination time may be directed to tape instead of a printer. Also, the VSE/POWER disk dump utility IPW$$DD can dump selected portions of the spool files to tape.

- New Status Report Statistics. The VSE/POWER status report produced at start-up/termination contains new values showing the virtual storage occupied by the VSE/POWER phases and the maximum GETVIS storage requested. These values help you to tailor the VSE/POWER partition as small as possible.

- Channel-to-Channel Attachment. This support is part of VSE/POWER's networking function and allows data to be exchanged between two virtual machines under control of VM/System Product if:

  - Both of these machines use VSE/POWER with networking, or
  - One machine uses VSE/POWER with networking and the other uses the networking capabilities of RSCS.

Documentation of the positional format of the JECL statements listed below has been removed:

```
* $$ JOB
* $$ LST
* $$ PUN
* $$ RDR
```

Also, the example for the calculation of storage requirements has been removed from Chapter 2.

Note that Chapters 7 and 8, and Appendixes B and C of the previous edition have been removed from this manual and included in a new manual VSE/POWER Application Programming, SC33-6276.

New for VSE/POWER 2.2

- Supported Hardware. IBM processors and devices newly supported by VSE/Advanced Functions 2.1 are supported also by this release of VSE/POWER.

- New Librarian Support. VSE/POWER has been changed to interact with the librarian newly implemented for VSE/Advanced Functions 2.1. This means, for example, that:

  - For the dynamic RJE,BSC or PNET trace option available with VSE/POWER 2.1, the collected trace information (about I/O interrupts and time-outs for a given line), may be dumped into the dump sublibrary of VSE/POWER's partition when the trace area is full.

  - For the SLI support, VSE/POWER allows you to access all sublibraries defined in a search chain for the VSE/POWER partition.

- Virtual Addressability Extension Support. VSE/POWER controlled partitions may reside in any of the available address spaces.

- Source Code Inclusion from VSE/ICCF Libraries. VSE/POWER's source-code inclusion (SLI) support allows users to include source-type library members directly from a VSE/ICCF controlled library.

- Spool-Access Support. The support, implemented primarily for use by IBM program products, is also available to user-written application programs. The support offers access to VSE/POWER services in response to requests, on a macro level, from within those programs. Services that may be accessed are:

  - Submission of a job for inclusion in the VSE/POWER reader or transmit queue.
  - Submission of print or punch output to the VSE/POWER list, punch, or transmit queue.
  - Retrieval of data from the VSE/POWER reader, list, or punch queue.
  - The display and change of job-handling data for jobs in a local queue or in a queue of a system within the same network of systems.
  - The routing of messages to the central operator or to a remote work station user on the same system or on another system in the same network.

  The cross-partition communication support available in the past can coexist with this spool-access support.

- Improved Networking Support.

  - Messages and commands to be transmitted are handled with a higher priority than data.

- The maximum buffer size allowed by VSE/POWER is 1800 bytes for a BSC line and 32,000 bytes for an SDLC line (as opposed to 1000 bytes for either in the past).
- Print files containing composed page-data stream records are tolerated for store-and-forward operations.
- VSE/POWER can start up and operate even if, for any reason, it is unable to start up its networking function.

- Spooled Output to Devices Controlled by Other Applications. This support, if used, shifts device control functions from VSE/POWER to an application program in another partition.

- Improved Remote Job Entry Support. VSE/POWER supports up to 250 remote-IDs and up to 100 BSC lines.

- Improved Accounting Support. VSE/POWER includes requestor identification (node name and user-ID) into the execution-, list-, punch-, and reader-account record.

- Improved Data-File I/O. Maximum size of data blocks is 12K bytes (as opposed to 2008 bytes in the past). To economize processor storage, VSE/POWER maintains the required data buffers in virtual address space.

- Improved Linkage to VM/System Product. If output is to be written to a virtual output device, VSE/POWER (on the virtual VSE system) passes the required control information to VM/System Product.

- Miscellaneous Other Improvements. The release includes minor improvements that contribute to better usability of VSE/POWER:

  - Display of the highest job-control return code by a notify message.
  - Reading SYSIN data for a diskette file definition in the user program.
  - More flexibility by extensions to the queue manipulation commands.
  - Saving, on tape, any queue entry regardless of the entry's "current" disposition.
  - The RJE,SNA-related messages are now contained in one module together with all other VSE/POWER messages. This facilitates translating the messages into another language.
  - The Shared Spooling function has become an integral part of VSE/POWER.

CHAPTER 1. INTRODUCTION TO VSE/POWER

VSE/POWER is a program that improves the throughput of a computer system by separating unit-record[1] input/output from the processing of application programs. This process, which is referred to as spooling, provides for the overlapping the unit-record I/O operations for all controlled partitions with the execution of application programs in these partitions.

VSE/POWER reads input that is to be spooled and stores this input in input queues on disk; it collects output to be spooled and stores this output in output queues on disk or on tape. Optionally, VSE/POWER reads input from a program library and inserts the retrieved data into the spooled input. When a program in a VSE/POWER-controlled partition is ready to accept input, VSE/POWER retrieves the required input from the input queue. When a unit-record device is available to punch or print output, VSE/POWER retrieves output from the applicable output queue and writes this output to the device.

VSE/POWER maintains input and output queues on disk to service the (logical) unit-record devices owned by the partitions under its control. VSE/POWER transfers, on request, spooled output to a subsystem in another partition. The subsystem would then print, display, or punch out this output under subsystem control.

The queues are filled or emptied by the available physical devices. When a program in a VSE/POWER-controlled partition makes a unit-record input request, VSE/POWER provides the next record from the applicable input queue. When the program makes a unit-record output request, VSE/POWER collects the record and places it into the applicable output queue. These operations take place at processor (real) storage or direct access storage speeds and are overlapped with processing in other partitions. As a result, programs complete faster and take better advantage of the system's resources.

VSE/POWER supports job and output classes and priorities as shown below. This helps in scheduling and controlling the processing of jobs and output on your computer system. Assume, for example, that you submit a job with a high priority assigned and a class that is handled by a high-priority partition (but lower than the priority of the VSE/POWER partition). VSE/POWER handles this job almost on a last-in-first-out basis. The classes and priorities supported by VSE/POWER are:

- For job input: 36 classes.

---

[1] input in card-image format from a card reader, diskette, or tape; output in card-image format or to a line or page printer

- For job output: 26 classes.
- 10 levels of priority for each class.

Application programs do not require modifications to take advantage of VSE/POWER.

VSE/POWER supports unit-record input and output devices such as card readers, punches, and printers. With the SYSIN tape reader function, it can also read card-image input from magnetic tape. You can direct VSE/POWER to route printer or punch output to tape rather than to an output queue on disk and have VSE/POWER print or punch this output directly from tape. VSE/POWER is capable of saving spooled data onto tape and to reload it from tape into existing queues again.

VSE/POWER's remote job entry (RJE) function supports remote work stations for entering jobs into the system and for transferring output to them. The function supports both BSC work stations and SDLC work stations.

For further information on remote job entry under VSE/POWER and the types of supported work stations, refer to the publication VSE/POWER Remote Job Entry User's Guide.

Generation of Control Tables

The available generation macros allow you to specify certain options for controlling the operation of VSE/POWER. You can define control values that adapt processing by VSE/POWER to the existing operating environment. Examples of the options that you can specify are the activation of job accounting or the support for operating in a network environment. Details concerning the generation of control tables are described in Chapter 2. Generation of control tables is not required; normally, however, this is one of the steps in preparing VSE/POWER for first start-up and use.

Supported Hardware

Processors Supported

VSE/POWER runs under VSE/Advanced Functions, the minimum operating system support for a VSE controlled computer system. VSE/POWER therefore supports the full range of IBM processors that are supported by VSE/Advanced Functions.

Devices Supported

With some exceptions, VSE/POWER supports the same devices that are supported by VSE/Advanced Functions. For information on the devices supported by VSE/Advanced Functions, refer to the manual VSE/Advanced Functions, Planning and Installation. The exceptions are listed below.

Devices NOT Supported

Optical- and magnetic-character equipment

Display stations

Terminal printers as follows:

IBM 3213
IBM 3284
IBM 3286
IBM 3287
IBM 3288

The IBM 7770 Audio response unit.

Devices Supported With Restriction

IBM 3289, Model 4, a printer - Is supported only as a PRT1 device.
IBM 3501 Card Reader - Is supported on RJE stations.
IBM 3521 Card Punch - Is supported on RJE stations.
IBM 3540 Diskette I/O unit - Is supported for input only.
IBM 4331 Integrated diskette - Is supported for input only.

Supported Telecommunication Devices: In general, VSE/POWER supports all BSC devices and systems that emulate the line discipline of the 2770, the 2780, or the 3780. For more details, refer to the publication VSE/POWER, Remote Job Entry User's Guide.

VSE/POWER supports BSC and SDLC work stations as listed below:

IBM 2770
IBM 2780
IBM 3741, Model 2 in 2780 emulation mode
IBM 3741, Model 4
IBM 3770 in BSC or SNA mode[2]
IBM 3780
IBM 3790 with the RJE facility

---

[2] Non-programmable models only - In BSC mode, 3770s are supported as 2770 or 3780 work stations; in SNA mode, LUTYPE1 protocols must be used.

Concepts and Functions

Figure 1 gives an overview of the data flow under control of VSE/POWER. The illustration shows the possible sources of input and output.

Following below is a discussion of the major functions and services of VSE/POWER.

Control of Program Execution

From the input queue on disk, VSE/POWER transfers the jobs to the designated partitions for processing. The jobs contain all required input; they usually contain JECL statements requesting job-specific handling by VSE/POWER. While a job is being processed, VSE/POWER stores printer and punch output as queue entries in output queues on disk or tape. VSE/POWER deletes or retains job input and output in accordance with disposition specifications which you supply.

Disposition of Jobs and of Output

When you submit a job, you can instruct VSE/POWER how to dispose of the job. You may, for example, instruct VSE/POWER to spool the job and hold it in the queue file unprocessed until your operator releases the job. You provide this kind of an instruction by specifying a "disposition," a one-character code.

If you do not specify a disposition, VSE/POWER assigns a disposition by default to any queue entry that it stores in its queue file for processing. It may change the originally specified (or defaulted) disposition if an unexpected event such as an irrecoverable I/O error occurs.

See Appendix A on page 309 for a list of disposition codes that you can specify and which VSE/POWER assigns by default or for unexpected events. The list shows the action required to release a queue entry that is being held because of its assigned disposition.

Segmentation of Output

Turnaround time for jobs with much printed or punched output can be improved by segmenting the output; this means that part of the output is printed or punched out before the entire job is finished. VSE/POWER supports output segmentation as follows:

By record-count specification (count-driven segmentation)
By specification in the input stream (data-driven segmentation)
By specification in the program (program-driven segmentation).

Segmentation is always done on logical boundaries for count-driven segmentation. For a printer this is a page (indicated by a skip to channel 1 or page overflow); for a card punch, this is a card-image record.

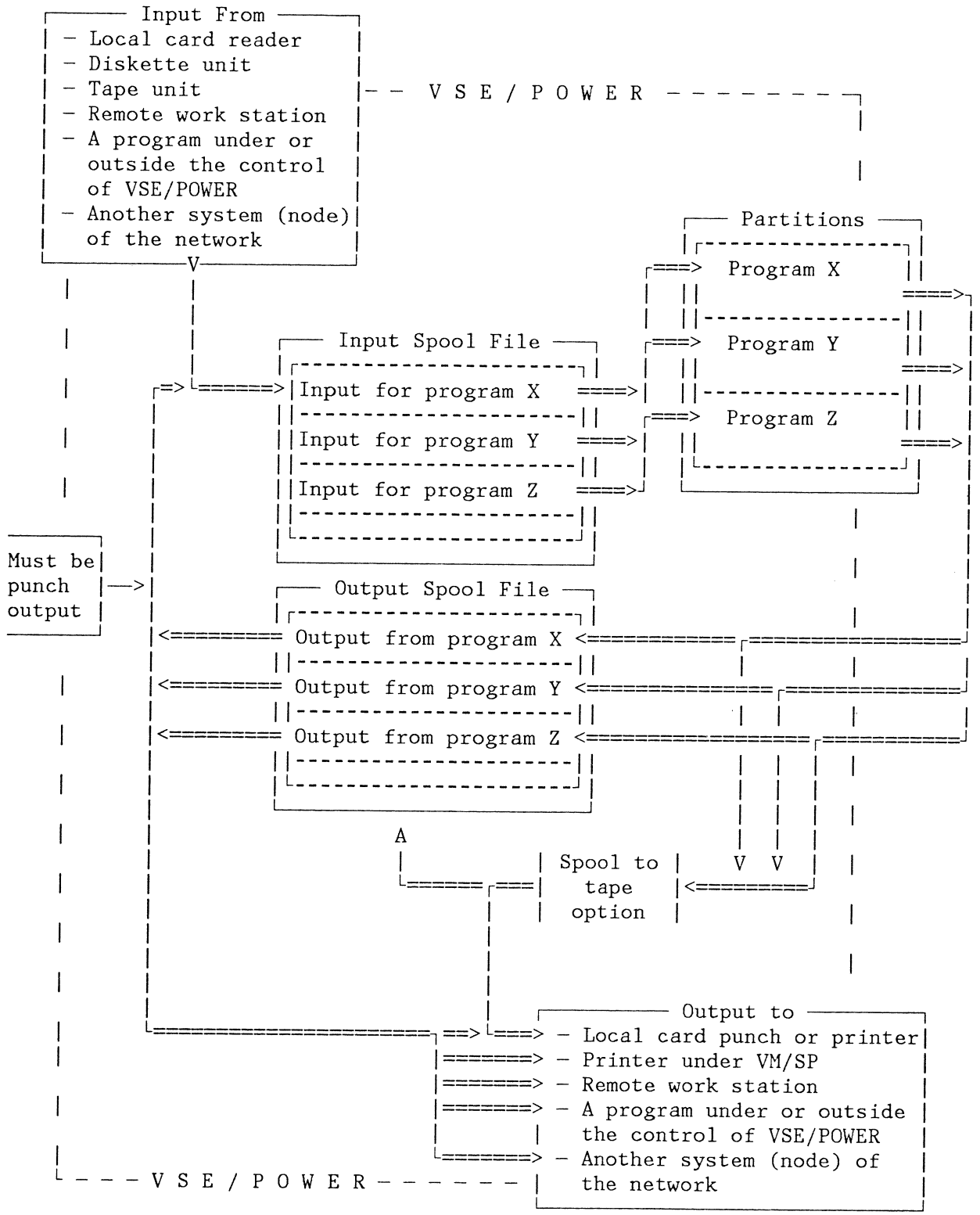


Figure 1. Data Flow Under Control of VSE/POWER

Accounting

VSE/POWER optionally collects accounting information for each job running under its control. It stores this information in an account file on disk. By means of a VSE/POWER command, you can request that this information be written into a separate disk file or to tape. You can also request that the account information be spooled as punched output and queued into the punch queue.

Accounting information written to disk can be saved by copying this information onto tape or into another disk extent.

Tape Support

VSE/POWER provides tape SYSIN support for fixed blocked and unblocked records. Both IBM standard-label and unlabeled tape files may be used. Nonstandard-label tape files are not supported.

The VSE/POWER off-loading function allows you to save, on tape, queue entries contained in the various VSE/POWER queues. You use the function for restoring these queue entries again under the same or another VSE system with VSE/POWER, if this should become necessary.

Tapes may be used for the spooling of job output. You can have these tapes printed or punched out later by a VSE/POWER list or punch task, whichever applies, or reloaded onto disk.

Networking (PNET) Function

This function permits the connection of a VSE system that includes Version 2 of VSE/POWER to nodes within the network with any of the following:

1. VSE/POWER, Version 2
2. SSX/VSE
3. MVS/System Product with JES2
4. MVS/System Product with JES3
5. VM/RSCS supporting the network job-interface (NJI) protocols

For programming support under 4 and 5, above, transmission can be only via a fully synchronous half-duplex multi-leaving BSC line. For the remaining types of network support, transmission can be also via an SDLC session.

The function permits the transmission of any of the following to other nodes within the network:

Jobs
Output
Job-related messages
Central operator commands

Message and command functions are available between all nodes in the network.

This manual includes PNET-related reference information; the publication VSE/POWER Networking User's Guide gives you guidance for using the function.

Remote Job Entry (RJE) Function

This function allows jobs to be entered from remote work stations and output to be obtained either centrally or at a work station.

This manual includes RJE-related reference information; the publication VSE/POWER Remote Job Entry User's Guide gives you guidance for using the function.

Shared Spooling

This function allows two or more (up to nine) VSE systems using VSE/POWER to share a single VSE/POWER spool file and, optionally, a single account file. This sharing of files applies to VSE systems running on different processors or on the same processor under VM/System Product (VM/SP). Once a job of a certain class has been queued into the shared VSE/POWER spool file, this job can run on any VSE/POWER system that is sharing the file and has a partition available for executing a job of that class.

Shared spooling can improve your computing center's efficiency because of the following benefits:

- Job-Load Balancing
  The queueing priority of jobs in VSE/POWER is valid for all sharing systems. If a high-priority job is entered into the spool files, it will execute on the first sharing system that has the job class free.

  Whichever processor has a reader, a writer, or a processing partition available for work can process output or jobs of a common class from the shared spool files.

- Directing Output to Particular I/O Equipment
  Output can be directed to a specific sharing system. For example, one processor could be used for all unit-record I/O operations, thus centralizing the equipment for ease of operator access.

  If only one sharing system has attached to it a specific piece of I/O equipment (for example, an IBM 3800 Printing Subsystem), a job can be run on one sharing system and its output can be directed to the sharing system with the specific equipment.

- Sharing of User Files

If a certain user file is physically accessible from only one sharing system, a job can be read in from another sharing system and processed on the one having access to this user file.

VSE/POWER also supports spooled output to (external) devices controlled by other applications. This support, if used, shifts device control functions from VSE/POWER to an application program in another partition.

Following is a summary of how shared spooling works.

Controlling Spool Files from the Operator's Console: The operator of each sharing system can control all jobs in the shared VSE/POWER queues. He cannot, however, control partitions or devices that do not belong to his own system. The PDISPLAY command provides information about the status of all jobs in the queue, including an indication as to which jobs are active and on which sharing system.

You can define in JECL statements on which sharing system a job is to be processed.

The selection of classes to be used for the various partitions on the different sharing systems is important. Consider providing guidelines in the operating procedures of your computing center. This selection may be influenced by the fact that some of the user files could be available also in shared mode, or that some jobs require tape units that might be available to only one of the processors.

Updating Control Information in Spool Files: To prevent concurrent updating of control information in the VSE/POWER spool files, only one sharing system is allowed update access to this type of information at any one time. Every sharing system, when it has a need to update control information on the spool files, requests exclusive write access to this information.

Note: To prevent one sharing system from retaining this write access indefinitely, a time limit can be specified in the TIME operand of the POWER generation macro. For more information about the various values available for the TIME operand, refer to page 58.

While a given sharing system has write access, it works as if it were the only user of the spool files. While another sharing system is waiting for write access, all functions not requiring an update of control information continue. Normal unit-record I/O processing continues even when a sharing system does not have update access to the spool files, until one of the following occurs:

- When spooling job input, the current DBLK group is filled.
- When processing spooled output, the queue entry is processed.
- During the spooling of output, the current DBLK group is filled.

Sharing VSE/POWER's Job Account File: If the VSE/POWER account file is shared, ensure that the operators of the sharing systems use the PACCOUNT command with care. The command, when issued from one of the systems, causes VSE/POWER tasks in the other sytems to wait if they need to update the control information in the account file. VSE/POWER of the waiting system does not resume processing until execution of the PACCOUNT command is finished. Consider using the PACCOUNT command when such a delay causes the least amount of disruption to processing.

Spool-Access Support

The spool-access support allows a program running under or outside the control of VSE/POWER to access VSE/POWER services. A program using the support can, for example:

- Retrieve queue entries from the local VSE/POWER queues.
- Submit jobs or output data for spooling to the VSE/POWER queues.
- Submit control requests or pass VSE/POWER commands (such as PALTER, PDISPLAY, PHOLD, PXMIT) to control the handling of specific queue entries.

Normally, IBM supplied components make use of this kind of support. For details, refer to VSE/POWER Application Programming.

External Device Support

This support is a dedicated application of the spool-access support. It shifts the control for writing spooled output to a device from VSE/POWER to a subsystem or application program. The subsystem may run in a partition under or outside the control of VSE/POWER. The support allows you, for example, to process output spooled to the LST or PUN queue on a device which is not supported by VSE/POWER. For details, refer to VSE/POWER Application Programming.

Control of Operation

Throughout the various steps of input, execution, and output, the jobs running under VSE/POWER can be controlled as follows:

- Through job-entry control language (JECL)

  Along with the input for a job, you may submit JECL statements that specify how VSE/POWER is to handle the job.

- Through central operator commands

  From the beginning of job input until the end of list and punch output, the central operator may start and stop processing of the

job or its output; the operator may display or alter execution and I/O performance characteristics, or perform similar tasks.

If the networking function is used, the central operator may control the transmission of jobs and output between nodes in a network. The operator may also control the queues at another node within the network.

- Through remote terminal operator commands

  The operator at a remote work station may also perform many of the tasks outlined above for the central operator.

- Via the spool-access support

  This support enables an application program in one of the system's partitions to control the processing of jobs or job output under VSE/POWER. The effects of these controls are similar to those available to the central operator or to a remote operator. However, controlling jobs from an application program is restricted to the use of certain commands as indicated in the PWRSPL macro. For details, see VSE/POWER Application Programming.

Job input as well as job output may be held in the VSE/POWER queues for execution or for printing or punching at a later time. This allows you, for example, to hold a job or output that needs hours for execution or printing. You can release that job or output for processing later when the system is less occupied.

VSE/POWER is structured into a series of asynchronously executing tasks. The operator becomes aware of these tasks when, for example, he requests a display of active tasks. These tasks, routines that share the system's resources together with other programs, are listed below:

- Reader task - It reads input from a unit-record or tape device.
- Punch or list task - It writes spooled output to a unit-record device under control of VSE/POWER.
- Execution task - It makes input spooled by a reader task available to the program which is to process this input; it collects the output of a program running under VSE/POWER control.
- RJE task - It functions as a reader task for remotely submitted jobs and as a writer task for output to a remote work station.
- Transmitter/receiver task - It transmits or receives spooled input or output.
- Spool-access support task - It controls the spooling of input from and output to a program; it accepts program-spooled output for queueing into the applicable output queue.

- Device service task - It controls the transfer of spooled output to a partition for writing this output to a device owned by the subsystem (or application program) running in that partition.

In addition, tasks are available for various other VSE/POWER functions such as saving the account file and shared-spooling timer control.

The operator can request concurrent execution of as many read or write tasks as there are I/O devices available.

Processing Unit-Record I/O Requests

VSE/POWER intercepts input and output requests issued by application programs running in partitions under control of VSE/POWER. VSE/POWER reads input from unit-record devices, remote work stations, and other connected processors; it queues this input into its reader (RDR) queue on disk. When an application program requests such input, VSE/POWER transfers the requested input - record by record - from the RDR queue to the program.

When an application program writes a print or punch record, VSE/POWER transfers this record to the associated output data file (in the list or punch queue, respectively).

When the program's output is complete or one of the defined output segmentation limits has been reached, VSE/POWER transfers this output from the output data file to the specified destination. This destination can be a printer or a card punch; it can be a tape drive, a remote work station, or another connected processing unit.

Following is a summary of the input to and output from VSE/POWER:

Input to VSE/POWER

- Control information:

  Job entry control language (JECL) statements
  Central operator commands
  Remote job entry (RJE) commands
  JECL statements passed from another partition.
  VSE job control statements.

- Application-oriented data from:

  A card or diskette reader
  A tape unit
  A remote work station
  Another partition
  The network.

- Library members:

  VSE/POWER includes, on request, desired library members into the input stream. A member to be included can come from a VSE sublibrary accessible from the VSE/POWER partition or from a specified library controlled by VSE/ICCF.

Output from VSE/POWER

- Application-program output records stored in the output queues.
- Messages.

Notes:

1. VSE/POWER does not support spooling for programs that include any of the following (if used in a program, the affected job might be canceled):

   EXCP real
   CCWs with indirect data addressing (IDA flag set)
   CCWs with data chaining (CD flag set)

2. The CCB-CCW addresses and data areas for the unit-record I/O to be spooled must be located in one of the storage areas listed below:

   - In the same partition as the affected program.
   - In the logical transient area (LTA) if this area is owned by the program.
   - In the shared virtual area (SVA) - applies only to write-request CCWs and I/O areas and not to CCBs.

System Considerations

VSE/POWER operates in virtual mode in one partition of the generated VSE system. It can service as many other partitions as are available.

If your system operates in 370 mode, VSE/POWER must run in a partition defined as shared. The processing priority of the VSE/POWER partition must be higher than the processing priorities of the partitions that are under control of VSE/POWER.

The remainder of this section on "System Considerations" is of interest primarily to users who install their VSE system product by product. If you install VSE/POWER as part of VSE/System Package, you may skip the section.

Programming Requirements

Following is a list of IBM programming support which must be installed in order to install or use VSE/POWER.

- VSE/Advanced Functions 2.1, Program Number 5666-301

  If you plan to use the shared spooling function of VSE/POWER, make sure that your system's supervisor includes the DASD share support. For details about generating a supervisor, refer to VSE/Advanced Functions, Planning and Installation.

- If SDLC sessions are used for remote job entry or networking:

  - Advanced Communications Function for VTAM (ACF/VTAM) 2.1 (or later release), Program Number 5666-280.
  - If the involved terminal equipment is linked via a channel-attached controller: Advanced Communications Function for Network Control Program/VS (ACF/NCP/VS), Program Number 5735-XX9, the release current when Release 3 of Version 2 of VSE/POWER becomes available.

- For communication with a node using VM/System Product, the following at that node: VM/370 Remote Spooling Communications Subsystem Networking 1.2 (or later release), Program Number 5748-XP1.

- For communication with an MVS node, either of the following at that node:

  - OS/VS2 MVS/System Product-JES2 1.3 (or later release), Program Number 5740-XYS.
  - OS/VS2 MVS/System Product-JES3 1.3 (or later release), Program Number 5740-XYN.

VSE/POWER can operate with the above listed programming support also on a virtual VSE machine under VM/System Product.

Machine Requirements

VSE/POWER has no machine requirements beyond those of VSE/Advanced Functions, except as follows:

- The compare-and-swap hardware feature is required if this is not already part of the standard configuration.

- A minimum of processor and virtual storage as listed in Figure 2 on page 14. Normally, a partition size of 1,024K to 1,536K bytes with a processor (real) storage allocation of 100K bytes is sufficient.

- Space on disk storage for its files; generally, a total space of 30M bytes is sufficient.

	Requirements (in K bytes)		
	Processor storage	Virtual storage	Partition size (incl. GETVIS)
Basic requirements			
without options	20	312	380
with options	20	328	396
Networking function	-	66	66 *
Remote job entry function			
BSC	2	18	20 *
SNA	-	34	34 *
Shared-spooling function	-	4	4 *
* The value has to be added to the basic requirement.			

Figure 2. Minimum Storage Requirements

Maximum Resource Support

The maximum number of resources supported by VSE/POWER are:

- Up to 14 logical printers, 14 logical punches, and one logical reader per partition.
- Up to 100 BSC work stations or BSC connected nodes concurrently. Through switched operation, more work stations may be in the network non-concurrently.
- Up to 250 work stations, SNA in synchronous data link control (SDLC) mode and BSC. 100 of these work stations may be BSC. Through switched operation, more work stations may be in the network non-concurrently.
- For each SNA work station: one console, one logical reader, one logical punch, up to three logical printers, and up to six simultaneous sessions.
- The maximum real storage that VSE/POWER can use is 512K bytes.

Throughput Considerations

By providing resources in addition to the minimum requirements, you can improve throughput as indicated below:

- Additional processor storage to:
  - Allow simultaneous operation of VSE/POWER tasks.
  - Improve input/output buffering.
- Additional virtual storage - To ensure efficient use of functions such as

  Job accounting
  Networking support
  Source code inclusion (SLI)
  VSE/ICCF notify support
- Additional disk storage for VSE/POWER queue files - To increase your system's operational flexibility.

Conventions for Showing Command and Statement Formats

The conventions listed here apply also to the formats of macros:

- Uppercase letters, numbers, and punctuation marks must be coded exactly as shown.
- Operands can start in any column after the operation code and one blank space.
- Continuation lines must start in column 16, except for job entry control language (JECL) statements. Continuation lines for JECL statements may start at any column from 6 through 16.
- Column 72 of each operand line, except the last, must contain a non-blank continuation character.
- Lowercase letters and words represent information that must be supplied by the person coding the command, statement, or macro.
- Options between braces { } and separated by an OR symbol (|) represent alternatives, one of which must be selected. The braces themselves may not be coded.
- Brackets [ ] indicate that the element they enclose is optional. Two or more elements between brackets and separated by an OR symbol (|) represent alternatives, of which only one may be selected. The brackets themselves may not be coded. Default values used by VSE/POWER are underlined.

The use of brackets and braces is shown by the examples below.

Operation	Operands
* $$ CTL	[CLASS={A|class}]

If you submit the statement with an operand, then you must supply a class specification, which is then used by VSE/POWER. If you submit the statement without an operand, VSE/POWER uses class A. You may, of course, submit the statement with an operand which explicitly specifies class A.

Options between braces and brackets may be stacked (for ease of information retrieval) as shown below:

Braces	Brackets
{*abc|	[,jobname[,jobnumber[,jobsuffix]]|
ALL[,listaddr]|	,*abc|
FREE}	,class]

- A comma at the beginning of a line merely indicates that this separator is to be coded with the associated operand. A continuation line cannot begin with a comma.

- Keyword operands consist of a keyword, followed by an equal sign followed by a value or a list of values. If you specify a list of values, this list must be enclosed in parentheses, and the specified values must be separated from each other by commas.

  For keyword operands, do not code a comma in front of the first operand. For example, if you omit the BURST operand in the * $$ LST statement, do not include a comma before the CHARS operand.

- Positional operands must be coded in a specific location relative to other operands. A comma must be coded for each omitted (optional) positional operand. Trailing commas may not be coded.

- Specification of a User Identifier

  Various statements and commands require the specification of a user identifier, either as originator (of a job, for example) or point of destination (of output). In the statement and command descriptions, this is indicated by the word userid. A user in this context can be one of the following:

Type of User	Explanation / Format of your Specification
A remote user	For userid, specify the number of the destination (RJE) work station as defined to VSE/POWER. Specify this in the form R000 - R250.
A VSE/ICCF user	For userid, specify the four-character identifier by which the destination user is known to VSE/ICCF.
A VM/SP CMS user	For userid, specify the identifier used to define the destination user to VM/SP.
A user at another node	For userid, specify the identifier used to define the destination user to the controlling system at the other node.
A logical user	For userid, specify the logical name which is defined in the applicable subsystem or at the destination node. Do not use R000 to R250 as logical names.

An identifier or logical name specified for userid cannot be any longer than eight (alphameric) characters.

- An alphameric character is defined as follows:
  - Any letter of the alphabet
  - Any of the numerals 0 through 9
  - The pound (#) sign
  - The dollar ($) sign
  - The commercial at (@) sign
  - The punctuation marks hyphen (-), period (.), and slash (/).

To use the table-generation macros documented in this publication, you should be familiar with the rules for coding programs or routines in assembler language. For details about these rules, see the publication OS/VS-DOS/VSE-VM/370 Assembler Language.

CHAPTER 2. PLANNING AND INSTALLATION

This chapter aids you in planning for VSE/POWER and in installing the product.

Planning for VSE/POWER: Before you install VSE/POWER, consider topics such as migration to this release of VSE/POWER, VSE/POWER's requirements, and restrictions regarding the use of the available hardware and software. These topics are discussed under "Migration to this Release," below, and under "Planning for VSE/POWER" on page 21. The section on planning includes the information you need to make VSE/POWER available in your VSE environment; it includes information about devices which need special considerations with VSE/POWER.

Installing VSE/POWER: After having restored VSE/POWER from the distribution tape to an online library, you most likely will have to define, assemble, and link-edit some or all of VSE/POWER's (table) generation macros. The section "Generating VSE/POWER Control Tables" on page 44 describes the generation macros.

Note: Throughout this chapter, the information regarding the IBM 3800 Printing Subsystem applies also to the IBM 3200 Printing Subsystem, if not specifically stated otherwise.

MIGRATION TO THIS RELEASE

Jobs that have been processed successfully under control of VSE/POWER 2.1 or 2.2 should run successfully also under control of VSE/POWER Version 2.3. However, each migration back and forth between VSE/POWER 2.3 and a previous VSE/POWER release requires a "cold start" of the queue and data file due to format changes. If necessary, you can offload (save) the queues onto tape under the old release and reload them again after having performed a cold start under Version 2.3 of VSE/POWER.

VSE/POWER print or punch files that have been saved or spooled onto magnetic tape under control of a previous VSE/POWER release can be printed or punched under control of VSE/POWER 2.3; the reverse, however, is not possible.

VSE/POWER Version 2.3 cannot coexist with previous VSE/POWER levels in a shared spooling environment; all systems must be at the same level.

As compared to VSE/POWER 2.2, changes in Version 2.3 of VSE/POWER include the following:

- The PDISPLAY queue command will show first the queue entries that are dispatchable (starting with priority 9 running down to 0), followed by the queue entries that are non-dispatchable.

- The RDR, LST, PUN, and SPOOL account records contain a two-byte field for the actual number of DBLK groups used. This field replaces the current number-of-tracks field. (The layout of the account records is described in VSE/POWER Application Programming.)

- The TRACKGP and BLOCKGP operands of the POWER generation macro are no longer supported; they are replaced by the DBLKGP operand.

- Values for the DBLK operand of the POWER generation macro of less than 1,000 bytes are no longer valid.

- The PRELEASE and PHOLD commands will no longer leave the manipulated queue entries at their original position in the class chain; instead, each modified queue entry is removed from the non-dispatchable/dispatchable class chain and added to the dispatchable/non-dispatchable class chain, respectively. This results in jobs being scheduled for processing in the sequence in which they are PRELEASEd rather than submitted.

If your currently used VSE system includes Version 1 Release 2 of VSE/POWER, you should note the following:

- If you have used your VSE/POWER with the Shared Spooling feature, then there is no need for this feature after migration. This release of VSE/POWER has the feature incorporated as a standard function.

- Functions that have been available with the VSE/POWER RJE feature are part of VSE/POWER. Support similar to the VSE/POWER RJE 1.2 multi-leaving (MLI) is included in the networking (PNET) function of VSE/POWER. A system with VSE/POWER using the MLI option of VSE/POWER RJE 1.2 cannot communicate with a system with VSE/POWER Version 2.

As compared to VSE/POWER 1.2, changes in Version 2 of VSE/POWER include the following:

- The system does not check for sufficient partition GETVIS space; the area is used dynamically.

- VSE/POWER includes the spool-access support for accessing VSE/POWER services from within a program. IBM recommends to use this support rather than the cross-partition communication (SPOOL-macro) support that has been available in the past and is retained to ensure program compatibility.

PLANNING FOR VSE/POWER

Before you can install VSE/POWER on your VSE system, you should plan for the program product's requirements in the following areas:

- Hardware Requirements - see "Hardware Requirements."
- Disk storage requirements - see "Disk Storage Requirements for VSE/POWER Files" on page 22.
- Processor and virtual storage requirements - see "Processor- and Virtual-Storage Requirements" on page 29.
- The options for using VSE/POWER - see "The Options for Your VSE/POWER" on page 38.
- Shared Spooling requirements - see "Shared Spooling Requirements" on page 38.
- ACF/VTAM and ACF/NCP generation requirements - see "ACF/VTAM- and ACF/NCP-Generation Requirements" on page 39.
- Special device considerations for VSE/POWER - see "Special Device Considerations for VSE/POWER" on page 40.
- Processing and Output Classes - see "Classes for Processing and for Output" on page 42.

The following sections discuss these topics beyond the scope of coverage under "System Considerations" on page 12.

Hardware Requirements

The processor and virtual storage sizes given under "System Considerations" allow a minimum set of VSE/POWER tasks to be loaded into virtual storage and at least one task to be active. For performance reasons, however, and for the concurrent operation of several VSE/POWER tasks, you should allocate more processor storage.

If you plan to use the RJE,SNA (remote job entry with SNA) function or the PNET,SNA (networking with SNA) function, reserve one partition for ACF/VTAM.

The attachment of SNA work stations or the use of SDLC lines for communication with another system, requires one of the following:

An IBM 3704 Communications Controller.
An IBM 3705 Communications Controller.
An IBM 3725 Communications Controller.
A communication adapter for an IBM 4321, 4331, or 4361.

The attachment of BSC work stations to your computer system or the use of BSC lines for communication with another system, requires at least one of the following:

- An IBM 2701 Data Adapter Unit (DAU) with Synchronous Data Adapter, Type II, equipped with EBCDIC or ASCII code and transparency.
- An IBM 2703 Transmission Control Unit (TCU) with Synchronous Attachment, Synchronous Base, Synchronous Terminal Control (EBCDIC or ASCII and Synchronous Line Set), or 3705 Communication Controller in 270x emulation mode.
- For IBM System/370 Model 135 and 138 processors, an ICA with a synchronous data adapter, Type II, and transparency.
- A communication adapter for an IBM 4321, 4331, or 4361.
- The compare-and swap feature on the processor operating under control of your VSE system.

Disk Storage Requirements for VSE/POWER Files

As an initial planning value, IBM recommends that you reserve a total disk space of at least 30M bytes for VSE/POWER's queue, data, and account files.

To make a fairly accurate estimate of the space required for the VSE/POWER queues, you should have an understanding of their structure. These queues are organized as one queue file and one data file. Within these two files, VSE/POWER arranges your jobs and your job output in logical queues; it sets up chains of queue pointers for the various classes and for priorities within these classes.

Queue File to Data File Relation

The first step in showing this relation is a brief discussion of the organization of the two files.

Organization of the Queue File: The file, named IJQFILE, consists of only one extent on one or more tracks or FBA blocks. It can reside on any supported disk device, but not on split cylinders. VSE/POWER accesses this extent using the programmer logical unit SYS001. The file contains, besides internal records, one queue file record per VSE/POWER job[3].

[3] Job input spooled for execution or job output spooled for output to a device; may be either, queued for transmission.

The queue file is organized in queue record blocks whose size depends on the device being used (see Figure 4 on page 26). Normally a queue record block contains 16 queue records of 256 bytes each.

If the queue file resides on an FBA device, the queue record block comprises eight FBA blocks. Thus, the number of FBA blocks allocated for the queue file should be a multiple of eight; otherwise the remaining FBA blocks are wasted.

Copy of Queue File in VIO Space: To reduce the number of I/O operations to the queue file, a copy of the entire queue file is placed in virtual storage (VIO space) while VSE/POWER is active. The VIO space can be seen as an extension of the page data set and is used as system work space. Once the queue file is placed in VIO space, most of the updates of a queue record are made only in the virtual-storage copy of the queue file. VSE/POWER updates the disk file only when necessary.

For details on estimating the required VIO space, see "The Size of the Queue File" on page 25.

Organization of the Data File: The file, named IJDFILE, consists of up to 15 extents, and all extents of the file must reside on devices of the same type:

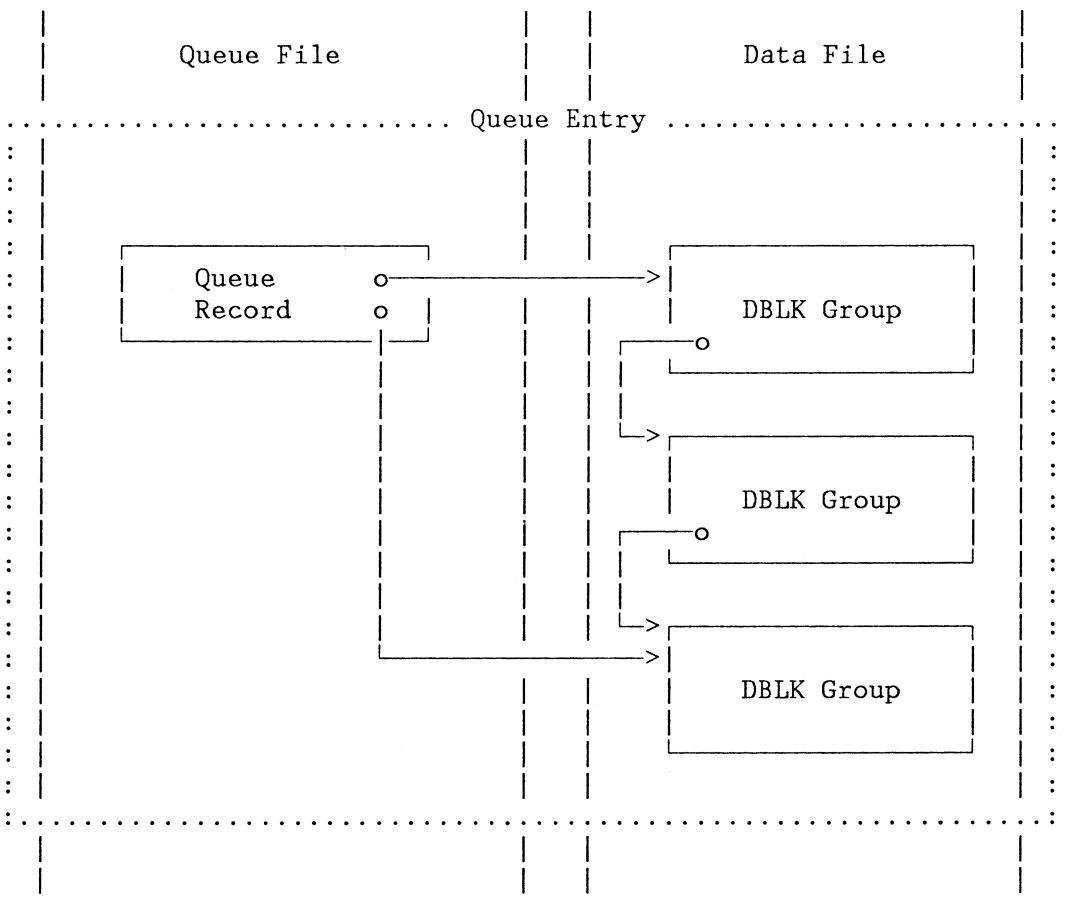
- If the file resides on CKD devices, split cylinders are not supported. If you use an IBM 3340, all of the file's extents must reside on the same type of 3348 data module.
- If the file resides on an FBA device, an extent can start on any block boundary.

The extents of the data file can be on the same volume or on different volumes. If they are on different volumes, the logical unit assignments for these extents must be in ascending sequence of the volumes being used. VSE/POWER uses the logical units SYS002 to SYS016 for accessing the data file extents.

The space available on the data file is arranged in **DBLK groups**. A DBLK group is the smallest unit of space that you can allocate to a VSE/POWER job. Each DBLK group contains an integer number of DBLKs (data blocks). The smallest DBLK group consists of one DBLK. You can define the number of DBLKs per DBLK group in the DBLKGP operand of the POWER generation macro.

A DBLK group can cross track, cylinder or even extent boundaries, since the entire data file is seen as one contiguous space, divided into DBLKs (which are grouped into DBLK groups).

File to File Relation: This is shown by Figure 3.



Notes

- A queue entry contains only one queue record, which describes a complete job or output. Thus the total number of queue records is identical to the maximum number of jobs to be queued by VSE/POWER at any one time.
- The queue record points to one or more DBLK groups allocated to the affected queue entry. The DBLK groups are chained together. Each DBLK group contains one or more DBLKs (data blocks).
- VSE/POWER determines the number of DBLK groups to be allocated to the queue entry. You determine the size of the DBLK groups.
- The device type used for the queue file need not be the same as is used for the data file.

Figure 3. Relationship between Queue File and Data File

As the illustration shows, a queue record always points to one or more DBLK groups which are chained together. These DBLK groups contain the spooled data associated with the VSE/POWER job which the queue record represents.

Organization of the Account File

This (optional) file, named IJAFILE, consists of one extent; it may reside on any supported disk device. VSE/POWER uses the logical unit SYS000 to access the file.

VSE/POWER maintains the account file as a sequential, variable-length unblocked file. The records of the file do not span track (or block) boundaries.

If the VSE/POWER account file resides on an FBA device, account records are blocked within a control interval of 2,048 bytes.

You may want to save your accounting information in a separate disk file if an account-file-full condition should occur. If so, define this file as one extent of the same size as the extent for your VSE/POWER account file.

Estimating Disk Space for the VSE/POWER Files

You must supply DLBL and EXTENT statements for the VSE/POWER files, and you define the sizes of these files in the EXTENT statements. A VSE/POWER file may be placed on a disk volume that contains other files. However, for performance reasons you should not place a VSE/POWER file on a disk volume that contains one or more other heavily used files.

The disk space required for the various VSE/POWER files depends on:

- The number of VSE/POWER jobs (queue entries) that may be queued at any one time.
- The number of DBLK groups needed by VSE/POWER to store the associated spool data in the data file.

Only if you have a good feel for the values applicable to your operation can you arrive at a fairly accurate estimate.

The Size of the Queue File: The size of the queue file depends on the maximum number of job or output queue entries existing at any one time. Normally, a queue file accommodating 500 queue records should meet the operational requirements. You may need a larger queue file if you plan to share your system's spool files or to process jobs from other nodes in a network.

Based on the selected maximum number of queue records you can now determine (1) the number of tracks needed for the queue file and (2) the necessary amount of VIO space as follows:

1. Divide the chosen number of queue records by the number of queue records per block (see Figure 4 for the appropriate values). Add one to the quotient (for the master record which occupies at least one queue record block) to obtain the necessary number of 'queue record blocks'.

2. Divide the calculated number of queue record blocks by the number of queue record blocks per track (see Figure 4) and round up the quotient if the remainder is not zero. This gives you the necessary number of tracks for the queue file.

3. Multiply the number of queue record blocks obtained under 1 (without the master record block) by the queue record block size (see Figure 4), which gives you the necessary VIO space in bytes, rounded up to the next multiple of 32K/64K, depending on the mode of operation.

For example, assuming the total number of allocated queue records on a 3330 is 500, then

1. The number of queue record blocks is: 500 : 16 + 1 = 32 (rounded down)

2. The number of tracks is: 32 : 3 = 11 (rounded up)

3. The amount of VIO space to hold the entire queue file is: 32 x 4K = 128K (rounded up).

For an FBA device, the number of queue record blocks would also be 32, assuming a total of 500 queue records. Since a queue record block comprises eight FBA blocks, the total number of FBA blocks required for the queue file would be 256.

Device Type	Queue Record Block Size	Number of Queue Records per Block	Number of Blocks per Track
2314/2319	6,144	24	1
3330/3333	4,096	16	3
3340/3344	4,096	16	2
3350	4,096	16	4
3375	4,096	16	8
3380	4,096	16	10
FBA	4,096	16	-

Figure 4. Device Type - Queue Record Block Relationship

The Size of the Data File: The size of the data file depends on the amount of space needed for each job or output spooled. To do a more accurate estimate, consider the following:

- The maximum number of VSE/POWER jobs in the system at any one time and the use of output segmentation.

  As a meaningful approach, IBM recommends that you estimate separately the number of I/O card images and the number of line images spooled to disk in a typical eight-hour shift. Choose a file size large enough to hold half this amount of data. This approach should avoid that VSE/POWER runs out of file space. File extents can be re-defined if they prove to be too large or too small. By regularly checking the status report (which VSE/POWER provides on request), you can get a feel for the amount of data that is being spooled by VSE/POWER at your location.

- The size of your DBLK groups.

  This size depends, to some extent, on the size which you defined for a data block. The sizes of the data blocks and the DBLK groups are discussed below.

Data-Block Size: You define the size of a data block, the physical record that VSE/POWER writes into a DBLK group, in the DBLK operand of the POWER generation macro. This means that the defined size is also the size of VSE/POWER's data buffers. If you do not define this size, VSE/POWER chooses a default size that suits the characteristics of the disk device assigned to the data file. The default values used by VSE/POWER for the various devices are given in Figure 5.

Device Type	Default DBLK Size	Approx. Number of Cards per Block *	Approx. Number of Lines per Block **
2314/2319	1,304	16	9
3330/3333	1,744	22	13
3340/3344	1,540	19	11
3350	1,952	24	14
3375	2,016	25	15
3380	2,016	25	15
FBA	2,032	24	14

* VSE/POWER suppresses trailing blanks; hence these figures represent the minimum values.

** Based on 132 print positions per line.

Figure 5. Default Sizes of Data Blocks

The smaller you define the size of your data blocks, the less processor storage is needed by VSE/POWER to have a given number of tasks active. On the other hand, the larger your data blocks are, the fewer disk I/O operations are required to access the same number of spool records. Choose a data block size which is the optimum of the following:

- The amount of storage you have available as data buffers for the concurrently active VSE/POWER tasks.
- A suitable number of data blocks contained in a DBLK group with little or no disk space wasted.

  If the data file resides on an FBA disk and your specification is less than an integer multiple of an FBA block (512 bytes), some space of the last FBA block is wasted.

DBLK Group Size: Once you know the size of your data blocks, you can set the size of your DBLK groups (in the DBLKGP operand of the POWER generation macro).

If you do not specify a DBLK group size, VSE/POWER assumes a default of 10 data blocks (DBLKs) per DBLK group.

The space available on your data file is arranged in DBLK groups. A DBLK group is the smallest unit that VSE/POWER allocates to a job. This allocation is completely independent of any device characteristics. Therefore you can tailor your DBLK groups entirely to the type of environment in which your VSE/POWER jobs run, without regard for the type of device on which your data file resides.

For example, if your installation runs many small jobs that produce minimal output, the DBLK group may be quite small - perhaps one DBLK. However, a specification of DBLKGP=1 can cause performance impacts due to increased VSE/POWER overhead.

If you run very large jobs producing large output, you can set the DBLK group size relatively high, up to the equivalent of several cylinders. However, if the DBLK group size is extremely large, fragmentation on the spool volume(s) can increase. The system may then run out of spool space, although, in fact, plenty of spool space still exists.

The FULL=YES operand of the PDISPLAY command shows the number of allocated DBLK groups for each queue entry. The Q operand of the PDISPLAY command shows the number of free DBLK groups currently in the system. This information can be used to analyze if the selected number of DBLKs per DBLK group is properly chosen. If there are many queue entries with one allocated DBLK group, you should reconsider the DBLK group size - it might be an indication that the DBLK group size is too high.

As a further help, specification of the SPLIM= operand in the POWER macro causes the operator to be notified when the indicated percentage of used spool space (default 90%) is exceeded.

The Size of the Account File: For each VSE/POWER job, VSE/POWER provides at least one of the following types of account records:

Reader account record
Execution account record
List or punch account record

To give you a feel for the account file disk-space requirements: a disk space of about 250K bytes can accommodate the account records for about 300 VSE/POWER jobs with an average of three account records per job.

Processor- and Virtual-Storage Requirements

Virtual and real address space must be allocated to the partition in which VSE/POWER is to run. The size of the partition is the sum of the following, rounded to page boundary:

Processor storage needed by VSE/POWER for fixing pages
Pageable storage
GETVIS storage

The use of virtual storage by your VSE/POWER partition is shown in Figure 6 on page 30. As a help in estimating your storage requirements, use the status report statistics that VSE/POWER provides after a PEND command (see Figure 12 on page 121). In this status report, the following lines (with sample values) are of interest for estimating storage requirements:

```
FIXABLE STORAGE USED BY VSE/POWER                        nnn K-BYTES
VIRTUAL STORAGE OCCUPIED BY VSE/POWER PHASES             nnn K-BYTES
MAXIMUM NUMBER OF K-BYTES FIXED IN PRESENT SESSION        nn K-BYTES
MAXIMUM GETVIS STORAGE REQUESTED IN PRESENT SESSION       nn K-BYTES
```

In addition, consult the virtual storage map produced by entering the job control or attention routine command MAP. For details about this command and the output which it produces, see VSE/Advanced Functions System Control Statements.

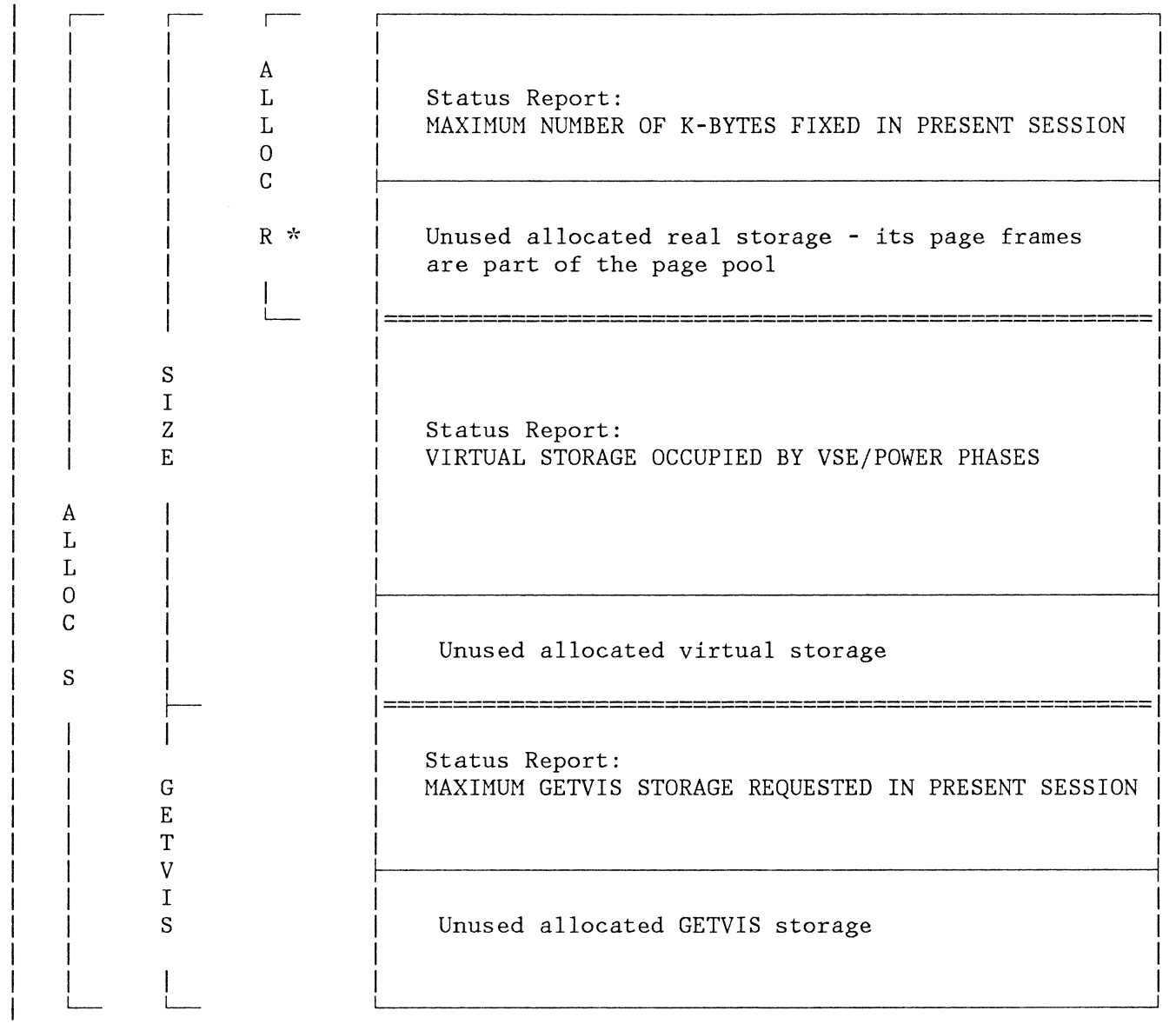


Figure 6. Overall Storage Requirements for VSE/POWER Partition

Referring to Figure 6, proceed as follows:

1. From the MAP command display (or the BG ASIPROC startup values), look up the K-byte values for:

   - ALLOC S (V-SIZE + GETVIS in MAP output)
   - ALLOC R (R-SIZE in MAP output)

- SIZE operand on // EXEC POWER statement
- GETVIS (GETVIS in MAP output)

2. Collect the 'used' K-byte values for the above status report lines of some typical VSE/POWER session.
3. Evaluate the unused portions of the allocated real, virtual, and GETVIS storage.
4. Let the unused portion of the allocated virtual storage approach zero (and leave reasonable buffers for the unused portions of real and GETVIS storage) by
   - Reducing the ALLOC S and ALLOC R values, and
   - Adjusting the GETVIS value by an appropriate SIZE specification.

This gives you the minimum storage requirements for the VSE/POWER partition.

Notes:

1. The status report value for 'MAXIMUM GETVIS STORAGE REQUESTED IN PRESENT SESSION' is only an approximation - it may actually be larger because of
   - GETVIS area control information
   - GETVIS storage fragmentation
   - RJE/SNA or PNET/SNA utilization
   - GETVIS storage requirements by VSE/Advanced Functions services.
2. The value for 'VIRTUAL STORAGE OCCUPIED BY VSE/POWER PHASES' depends on the functional support generated in the specific VSE/POWER phases.

VIO Space

In addition to allocating space for the partition in which VSE/POWER is to run, you must reserve a certain amount of virtual storage to be used as VIO space. This VIO space is needed to accommodate the virtual-storage copy of your queue file while VSE/POWER is active. The VIO area uses space on disk immediately behind the page data set.

For details on how to estimate the VIO space required for your queue file, see "The Size of the Queue File" on page 25. You define the VIO area during system start-up either in the IPL ASI procedure or in response to message 0I03D during system start-up. For details, see VSE/Advanced Functions: Planning and Installation, SC33-6193.

Detailed Storage Requirements

Figure 7 on page 33 shows the contents of the partition and gives the storage requirements in number of bytes. These byte values are supplemented by Figure 8 on page 34 and Figure 9 on page 36. The requirements given in Figure 9 are in addition to those given in Figure 7.

Following are the Notes referred to in Figure 7:

1. Must be allocated as fixable area.

2. Must be allocated as fixable area. For the actual requirement, refer to the "Fixable Area" column of Figure 9.

3. The requirement depends on the size of your exit routine.

4. See Figure 8 on page 34 for GETVIS requirements.

5. Required if you anticipate dynamic loading of a network definition table (see also "PLOAD: Load a Network Definition Table" on page 197).

6. 12K bytes are required for each 3800 printer setup being processed concurrently.

7. The size depends on the number of messages stored at any one time in the queues of the subsystems that use the support. Each message in a queue requires 128 bytes or more.

Purpose / Function	Program Space	GETVIS Space
VSE/POWER nucleus (including control tables)	20K (See Note 1)	
Fixable area	(See Note 2)	
Pageable area: basic functions	308K	
Areas for optional functions:		
Accounting support - CKD	6K	
Accounting support - FBA	6K	4K
User RDREXIT phase	(See Note 3)	
RJE,BSC support	18K + trace area size	
Shared spooling support	4K	
Cross-partition spool manager support (CTLSPOOL/GETSPOOL/PUTSPOOL)	8K	
Source (library member) inclusion	2K	
RJE,SNA support		
VSE/POWER SNA support	34K	
Remote control block		
SNA unit control block		
Work area control block		(Note 4)
LOGON tables		
Compaction table		
Logical unit table		
Networking support	66K + trace area size	(Note 5)
3800 printer support		
Work area for printer setup processing		12K (Note 6)
Notify support		(Note 7)

For the referenced notes, see the preceding page.

Figure 7. The VSE/POWER Partition - Storage Requirements

Type of Storage or Control Block	GETVIS Requirements in Bytes
Remote control block for all work stations	2,048 + r x 32
SNA unit control blocks for all work stations	(192 + 160 x n1) x n2 rounded to next higher page boundary
Work area control blocks plus two I/O buffers	(n1 + 1) x 1K x n2 rounded to next higher page boundary
LOGON storage for all work stations	2,048
Compaction table storage per work station	(0.5r + 0.5) x 2K rounded to next higher page boundary
Logical unit name storage	2,048

Legend:
r = Number of remote work stations (remote-IDs or number of PRMT macros with TYPE=LUT1).

n1 = Total number of LOGONs allowed as a result of the SESSLIM=value specifications in your PRMT generation macros.

n2 = Number of work stations: Either your SNA=wscount specification in the POWER generation macro or the number of PRMT generation macros. The maximum is 250.

Figure 8. GETVIS Calculation for RJE,SNA Support Control Blocks

Legend for Figure 9 :

A = 96 if printing/punching from spool tape; otherwise A = 0

B = n x 16 (rounded to the next multiple of 32)
n = number of spooled devices

C = 96 if spooling to tape; otherwise C = 0

D = 32 x n where $n = \frac{DBLK + 39}{32}$ (rounded to the next higher integer)

E = 32 x e where $e = \frac{(R+8) \times 26}{32}$ (rounded to the next higher integer)

R = logical record length as specified for the diskette I/O unit

F = 256 (needed only for building control records

G = ((1 + tr x bt) + (1 + rv x br)) x bs

H = ((1 + tr x bt) + (1 + rv x br)) x bs
where for G and H:
tr = Number of active job/output transmitters
bt = Number of buffers per active job/output transmitter
rv = Number of active job/output receivers
br = Number of buffers per active job/output receiver
bs = For G: Buffer size + 28 (rounded up to next multiple of 32)
bs = For H: Buffer size + 128 (rounded up to next multiple of 128)

J = Size of the spool-access support buffer as defined for a specific function request (up to a maximum of 64K bytes)

N = DBLK + 16 rounded up to the next multiple of 128 bytes
where DBLK = The value specified in DBLK=n of the POWER generation macro

P = 32 x n where $n = \frac{DBLK + 39}{32}$ (rounded to the next higher integer, but not to exceed 2048)

S = 256 if shared spooling is specified, otherwise S = 0

V = 256 if any connection is started using SDLC line discipline;
otherwise V = 0

X = 3200 if SLI statements are to be processed; otherwise X = 0

Y = 128 for each additional SLI nesting level; otherwise Y = 0

Z = n x 40 + 116 (rounded up to the next multiple of 128)
where n = Number of nodes defined in the network definition table

VSE/POWER Component or Task	Requirements in Bytes	
	Fixable Area	GETVIS Area
Dynamic control blocks (always required)	6,144	
Each data file extent	D	
PNET control blocks for networking	192	S+V+Z
SNA control block for RJE,SNA support	192	
Each local writer task		
With one buffer	608+P+A	512+N
With two buffers	608+2P+A	512+N
With four buffers	640+2P+A	512+2N
Each local punch task	608+P+A	256+N
Each local card reader task		
With one input buffer	480+P	256+F+N
With two input buffers	480+2P	256+F+N
Diskette I/O unit connected	640+P+E	256+F+N
Each local 3540 reader task	608+E	256+F+N
Each print status task	1120	512
Each temporary command processor task	480	2,048
Each local tape reader task	512	4,352+N

Figure 9 (Part 1 of 2). Values for Calculating VSE/POWER's Fixable and GETVIS Areas

VSE/POWER Component or Task	Requirements in Bytes	
	Fixable Area	GETVIS Area
Each off-load task		
Save (backup) function	448	256+N
Load function	448	12,544
Select function	448	12,800
Cross-partition support		
Reader task (PUTSPOOL)	480	256+F+N
Writer task (GETSPOOL)	608	256+F+N
Control task (CTLSPOOL)	416	
Each device-service task	768	256+N+J
Each spool-access-support connection		
CTL function	352	768+J
GET function	576	1024+N+J
PUT function	480	1024+N+J
RJE,SNA tasks		
Each reader task	416	256+F+N
Each writer task	544	256+N
RJE,BSC tasks		
Each reader task	416	256+F+N
Each writer task	544	256+N
Each execution reader (one per partition)	448+B	512+N+X+Y
Each execution writer (up to 16 per partition)	352+C	768+N
First RJE,BSC line	352+1856	
Each additional RJE,BSC line	1856	
Each job/output transmitter	352	1,536+N
Each console transmitter	352	1,280
Each job/output receiver	352	2,560+N
Each session (PNET,BSC)	1,152+G	2,176
Each session (PNET,SNA)	960	3,200+H

Figure 9 (Part 2 of 2). Values for Calculating VSE/POWER's Fixable and GETVIS Areas

The Options for Your VSE/POWER

VSE/POWER as shipped provides support in accordance with default values set by IBM for the various table-generation options. Review the description of the VSE/POWER generation macros to see if these defaults meet your requirements. For this description, turn to "Generating VSE/POWER Control Tables" on page 44. If these defaults meet your requirements, there is no need for any further planning. In this case, it is sufficient to assemble the macros without any operands specified. Most likely there will be a need for specifying some of the operands.

Shared Spooling Requirements

Before using VSE/POWER shared spooling, do the following:

- Declare the DASD that is to be used for the VSE/POWER spool files as sharable when VSE is initialized.
- Define a VSE lock file during VSE initialization. This file must be on a shared volume. For performance reasons, it should not be located on the same physical device as any of the VSE/POWER spool files.

  If VSE systems sharing the same lock file are not to share the same VSE/POWER files, these files must reside on different volumes.

  For additional information on how to define a lock file refer to the VSE/Advanced Functions System Management Guide.
- Before any VSE system with shared spooling is initialized, make sure that the label information for the spool files is the same in all systems sharing the files.

Sharing systems must be on the same release level. However, they do not have to be generated identically. In this case, values for DBLK and DBLKGP are provided by the first sharing system to be initialized with spool-file formatting (cold start).

It is also possible to share the VSE/POWER generation when a shared library is used. In this case autostart must be used in order to change the SYSID field by use of the SET SYSID= statement.

If several VSE systems are running on virtual machines, all under control of VM/SP, the CP command SET CPUID must be used to provide a unique VM/SP CPUID for every VSE system. Shared spooling by systems under control of VM/SP requires specific mini-disk definitions so that VM/SP can use the correct RESERVE/RELEASE commands. For details refer to Virtual Machine Running Guest Operating Systems, GC19-6212.

ACF/VTAM- and ACF/NCP-Generation Requirements

If you plan to use the RJE,SNA or PNET,SNA support of VSE/POWER, then ACF/VTAM must be included in your system.

Your generated version of ACF/VTAM must meet the following requirements:

- VSE/POWER must be defined to ACF/VTAM as an application program. You do this by using the APPL definition statement.

  If RJE,SNA is used, define VSE/POWER with the name which you specified in the SNA operand of your POWER generation macro.

  If PNET,SNA is used, APPLID definition statements must be included. The statements must specify the names which you used in the APPLID operand of your PNODE generation macros.

- The password specified in the PRTCT operand of the APPL macro, if included, must match the password specified in the SNA operand of your POWER generation macro.

- The names of the SNA logical units (luname) must be defined to ACF/VTAM by way of LU macros. These names must be the same as those you supply with the LU=(namelist) operand of your PRMT macros (if any) and in the VSE/POWER RJE commands where the value 'luname' is to be specified.

  As a logical unit name, you cannot use EOJ or ALL.

  If the buffer limit defined (to ACF/VTAM) for a logical unit is exceeded, the session with that logical unit ends immediately without an error message. A situation such as this can occur in an overloaded system.

You may plan to make use of the USS[4] tailoring services in order to support a simplified LOGON procedure. However, using these services depends on the type of work station you have; you can use them, for example, for a 3770 but not for a 3790.

For more information about ACF/VTAM and ACF/NCP generation requirements, refer to the publication Network Program Products Planning.

---

[4] USS = Unformatted system services

Special Device Considerations for VSE/POWER

IBM 2560, 5424, and 5425 Multifunction Card Devices

VSE/POWER supports these multifunction card devices for input and output; it handles 96-column cards as are used on the IBM 5424 and 5425.

If SYSPCH is assigned to these multifunction card devices, VSE/POWER supports:

Program-controlled stacker selection
Punch and interpret
Card print
Punch/print

The use of these devices is subject to restrictions as follows:

- VSE/POWER cannot update or interpret a card input file.
- VSE/POWER ignores stacker selection for input cards.
- Read column binary is not supported.
- VSE/POWER uses hopper 1 for input.
- VSE/POWER uses the stacker you specify for output if no separator cards are specified; it uses stacker 2 for output if separator cards are specified.
- VSE/POWER spools both punch and read records for a given partition only if a unique physical device address is specified for each I/O function (00C for the read function and 01C for the punch function, for example). You may use a dummy assignment for one or both of the functions.

IBM 3525 Printing Card Punch

VSE/POWER supports the basic read and punch functions of the device.

If SYSPCH is assigned to the device, VSE/POWER supports the following functions:

Program-controlled stacker selection
Print
Punch/print (multiline)
Automatic line positioning
User-controlled line positioning
Print overflow

The assignment of a logical unit to an IBM 3525 card reader or card punch is subject to the same rules that apply to the assignment of a logical unit to a multifunction card device.

IBM 3540 Diskette Input/Output Unit

VSE/POWER supports the 3540 only as a unit-record input device: like a card reader (SYSIN mode) or as a supplement to a card reader (data mode). VSE/POWER can do either of the following:

- Read the job control statements from a card reader (or any other input medium) and then switch to the diskette unit to read data from the mounted diskette.
- Read control statements and data (records of a length of up to 128 characters) from the diskette unit, using no other input medium.

IBM Magnetic Tape Units

The following tape processing functions for VSE/POWER tasks are supported:

- For the execution writer, spooling to tape rather than disk; this is the tape-spooling function.
- For the list or punch processor, a tape rather than disk can be used as input device; this is the spool-tape print or punch function.
- For the reader, a tape rather than a card reader (or diskette I/O unit) can be used as input device; this is the tape SYSIN reader function.
- For queue entries of VSE/POWER queue and data files, a save and load function can be performed; this is the offloading queue function.
- For saving the VSE/POWER account file, a tape can be used as the output device.

IBM 4248 Printer

The 4248 printer operating in native mode provides the following functions:

- FCB-controlled print speed (2,200 or 3,000 or 3,600 lines per minute).
- FCB-controlled intermixing of vertical spacing (six or eight lines per inch).
- Horizontal-copy printing.

- Print-band verification.
- Stacker-level control.

For more details on the VSE/POWER support of the 4248 printer, see "VSE/POWER Support of the IBM 4248" on page 109.

Device-Independent FCB Image Names

VSE/POWER supports device-independent specification of FCB-image phases in the * $$ LST statement. If you specify the name of the phase in the form $$$$xxxx (where xxxx is a string of characters of your own choosing), then VSE/POWER replaces the dollar signs by a character string as follows:

FCB1	For a 3800 printer
FCB2	For a PRT1 printer
FCB3	For a 3203-1 printer
FCB4	For a 5203 printer
FCB5	For a 4248 printer

It follows then that for printing you can use a printer of a type other than that specified for spooling the output (during program execution). It is your responsibility, however, supply FCB images with phase names according to the above conventions.

If the output file is directed to an IBM 3800 printer, the first four characters of the FCB phase name must be FCB1.

Classes for Processing and for Output

VSE/POWER allows you to group your jobs and their output by classes. You may, for example, want to group your jobs by required resources such as certain libraries that are defined as accessible from certain partitions.

Consider grouping the output of these jobs by devices to which this output is to be written eventually. For example, output from different jobs and using the same form number can be kept together in the list queue by matching class and priority specifications. This helps reduce forms mount effort by the central operator.

You can set up classes, one character definitions, during VSE/POWER start up in the PSTART command as follows:

1. When starting a partition under VSE/POWER

   You can assign up to four different classes to a partition. For example: if you assign to a partition the classes A, B, C, and D, then VSE/POWER routes all jobs of classes A through D to this partition for execution.

One of the four classes may be partition dependent. The numbers 0 through 9 are available for this purpose as follows: 0 = partition BG, 1 = partition F1, 2 = partition F2, and so on.

For the assignment of classes to partitions, you can use the letters of the alphabet and the numbers 0 through 9.

2. When starting an output-writer task

   You can start a task for processing spooled output of up to four classes, each one designated by a letter of the alphabet.

   Assume a VSE system with three printers for output: one IBM 3800 and two PRT1 printers. The IBM 3800 is used for high-quality mass production, while the PRT1 printers are used for standard quality output on seven different forms. In that case, you might start list-writer tasks for classes as follows:

   - The task for the IBM 3800: class M.
   - The task for the first PRT1 printer: classes CDEF.
   - The task for the second PRT1 printer: classes FXYZ

   For the assignment of output classes, you can use the letters of the alphabet.

IBM considers it a planning responsibility to set up local conventions for the assignment of classes to the available partitions, to the jobs that are to be processed in these partitions, and to the output of those jobs.

This concludes the discussion of planning VSE/POWER requirements. The next section discusses the installation of the product.

INSTALLING VSE/POWER

You receive VSE/POWER on magnetic tape. The product as shipped includes two pre-assembled VSE/POWER (table) generations which you may consider when planning for VSE/POWER; for more information about these pre-assembled generations, see "Generating VSE/POWER Control Tables" on page 44.

VSE/POWER may reside on the system residence volume (SYSRES) in the system library or on any other volume in a private library.

Installing VSE/POWER as Part of VSE/System Package

If you install VSE/POWER as part of VSE/System Package, you may ignore this section and continue reading the next section, "Generating VSE/POWER Control Tables."

Installing under VSE/Advanced Functions

Installation of VSE/POWER requires that you first install VSE/Advanced Functions, the minimum operating system support for a VSE system. VSE/Advanced Functions must be operational.

You install VSE/POWER under control of the Maintain System History Program (MSHP), a program of VSE/Advanced Functions. The publication VSE/Advanced Functions, Planning and Installation includes a sample job for the installation of a program product on top of VSE/Advanced Functions.

After completion of the installation process, you can generate the required VSE/POWER control tables. This is discussed under "Generating VSE/POWER Control Tables," the next section.

GENERATING VSE/POWER CONTROL TABLES

Generation macros are available for the generation of required control tables. This section of the manual describes these macros and gives you a guideline for their assembly; it includes examples of source code for an assembly.

VSE/POWER is shipped with two pre-assembled generations of VSE/POWER tables as summarized below. If one of these generations meets your requirements, there is no need for you to do a VSE/POWER generation. The generations are:

1. A phase named POWER - It provides support in accordance with the default values documented under "POWER Generation Macro" on page 47.

2. A phase named IPWPOWER - It provides the support as 1, above, plus:

```
ACCOUNT=YES
SPOOL=YES
SUBLIB=P
MEMTYPE=P
NTFYMSG=100
```

Both versions relate only to the POWER macro; other macros, if required, must be coded, assembled, and link-edited.

The Generation Macros

These macros are available for generating the required control tables such that VSE/POWER meets your location's requirements. For a full understanding of remote job entry (RJE) and networking (PNET) requirements, you may want to consult the publications VSE/POWER Remote Job Entry User's Guide and VSE/POWER Networking User's Guide, respectively.

The VSE/POWER generation macros are:

POWER — The macro defines the scope of local support; it provides for activation of support for:

- Networking (PNET)
- Remote job entry using SNA (RJE,SNA),
- VSE/ICCF notify function
- Shared spooling

For a description of the macro and its operands, turn to "POWER Generation Macro" on page 47.

PLINE — The macro defines the hardware characteristics of a BSC line and is required for RJE,BSC and for PNET,BSC support. For a description of the macro and its operands, turn to "PLINE Macro" on page 60.

PRMT — The macro defines the hardware characteristics of an RJE work station. For a description of the macro and its operands, turn to "PRMT Macro" on page 63.

PNODE — The macro generates an entry of the network-definition table, which is required by the networking function. For a description of the macro and its operands, turn to "PNODE Macro" on page 72.

PCPTAB — The macro generates a compaction-table as may be required by the RJE,SNA support. For a description of the macro and its operands, turn to "PCPTAB Macro" on page 76.

For the assembly, the macros have to be submitted in the sequence: POWER followed by PLINE followed by PRMT. The macros PNODE and PCPTAB are to be assembled separately. The macros are discussed in this section in the sequence as they are to be submitted.

The table below helps you select the generation macros and operands related to the available RJE, PNET, and shared spooling support.

Legend: X = The macro or operand applies.

To Have Support for:					Code the Macro (or Operand of the Macro) as Shown
Remote Job Entry		Networking		Shared Spooling	
RJE,BSC	RJE,SNA	PNET,BSC	PNET,SNA		
X	X	X	X	X	POWER
		X	X		PNET
		X	X		RDREXIT
X					RJEBSC
				X	SHARED
	X				SNA
				X	SYSID
				X	TIME
X		X	X		TRACESZ
X		X			PLINE
X		X			PSWRD
X		X			TRNSP
X	X				PRMT
		X	X		PNODE
	X				PCPTAB

POWER Generation Macro

The macro and its operands define the support for VSE/POWER options such as job accounting, remote job entry, networking, and the shared spooling function.

Format of the Macro

Name	Operation	Operands
[name]	POWER	[ACCOUNT={NO|YES}] [,CLRPRT={YES|NO}] [,COPYSEP={YES|NO}] [,DBLK={0|n}] [,DBLKGP={10|n}] [,FEED={YES|NO}] [,JLOG={YES|NO}] [,JSEP={(0,0)|(n,m)}] [,LTAB=linetab] [,MEMTYPE=membertype] [,MRKFRM={YES|NO}] [,MULT12={YES|NO}] [,NTFYMSG=nnn] [,PAUSE={NO|YES}] [,PNET=phasename] [,PRI={3|n}] [,RBS={(0,0)|(n,m)}] [,RDREXIT={NO| localexit| ([localexit],pnetexit)}] [,RJEBSC={YES|NO}] [,SHARED={NO|(Q[,A])}] [,SNA={YES| ([prmtno|wscount], [password], [applid|POWER])}] [,SPLIM={90|nnn}] [,SPOOL={YES|NO}] [,STDCARD={(0,0)|(n,m)}] [,STDLINE={(0,0)|(n,m)}] [,SUBLIB={NO|sublib}] [,SYSID=n] [,TIME={(5,0,60)|([t1],[t2][,t3])}] [,TRACESZ={12|nnn}]

If you code the macro with a name, the assembler inserts this name into the PHASE statement, which is being generated as the first statement of the resulting object output. You can specify a name of one to eight alphameric characters, the first of which must be alphabetic. If you do not specify a name, the assembler inserts POWER as the phase name.

Description of Operands

ACCOUNT={NO| YES}
Specify ACCOUNT=YES if you want VSE/POWER to maintain job accounting information. For this specification to be effective, start up your system with JA=YES specified in the IPL command SYS.

If you specify ACCOUNT=NO, no VSE/POWER accounting information is maintained.

Accounting information collected by VSE/POWER is stored in VSE/POWER's account file, IJAFILE.

CLRPRT={YES|NO}
This operand applies to IBM 3800 printers.

If you specify CLRPRT=YES, VSE/POWER clears the printer's page buffer at end of job and waits until all printing for the current job is complete. This may slow down your 3800 because the printer must empty its page buffer completely before VSE/POWER can transfer new data into this buffer.

If you specify CLRPRT=NO, VSE/POWER begins to fill the page buffer with the next job's output without waiting for completion of the printing for the previous job. But this can be risky. If an error occurs on the 3800 during the printing of the previous job, the output may be incomplete, and this job may already be deleted (if it had a disposition of D). VSE/POWER is then unable to reprint any lost data. As a bypass, you can specify a disposition K in the * $$ LST statement. This causes VSE/POWER to hold the output in the LST queue after processing. After having verified that the output is complete, you can delete the output job from the queue by a PDELETE command.

COPYSEP={YES|NO}
Specify COPYSEP=YES or omit this operand if VSE/POWER is to produce separator pages or cards between each copy of an output file. However, VSE/POWER produces separator pages or cards only if either:

- A nonzero value is specified in the JSEP operand (see below), or
- A specific request is made in the applicable JECL statement, either * $$ LST or * $$ PUN.

If you specify COPYSEP=NO, VSE/POWER produces printed or punched output without separator pages or cards between copies of this output. The COPYSEP specification can be overridden by a specification in the JECL statements * $$ LST or * $$ PUN.

For more information about separator pages or cards, see the description of the JSEP operand.

DBLK={0|n}

The operand specifies the size of a data block, the unit of transfer which VSE/POWER uses for both disk and tape. It specifies, therefore, the size of each data buffer. Data buffers are used as I/O areas by the execution processors, reader and writer tasks, and RJE routines.

The value for n must be a number from 1,000 to 12,288.

If you omit the operand or specify 0, VSE/POWER uses the default values as shown in Figure 5 on page 27. Whenever VSE/POWER performs a cold start, it uses the DBLK value that it finds already in existence in the queue file.

DBLKGP={10|n}

Specifies the number of data blocks (DBLKs) grouped together in a DBLK group. A DBLK group is the smallest unit of space that can be allocated to a VSE/POWER job on the data file. The smallest DBLK group consists of one DBLK, that is, the minimum value for n is 1. The maximum value that you can specify for n is 32,767. If you omit the operand, VSE/POWER sets the DBLKGP value to 10 DBLKs.

For more information about DBLK groups refer to "The Size of the Data File" on page 27.

FEED={YES|NO}

Specify FEED=YES if, on end of file, VSE/POWER is to eject the previously used diskette and feed a new diskette.

Specify FEED=NO or omit the operand if this automatic eject and feed operation is not desired. You can override this specification by way of the JECL statement * $$ RDR.

JLOG={YES|NO}

Specify JLOG=YES or omit the operand if VSE/POWER is to display job-related data on the system console whenever one of the following occurs:

- A new VSE/POWER job is started by a supported partition.
- A job or output is transmitted to another node in the network.
- A job or output is received from another node in the network.

The job-related data is:

Job name and number
Originating information

User-information (if present) is included in the job start message only.

Specify JLOG=NO if no such logging is desired.

JSEP={(0,0)|(n,m)}
Specify JSEP=(n,m) if VSE/POWER is to include extra pages or cards between jobs. In the operand:

n = The number of separator pages to be printed across perforations.
m = The number of separator cards.

For n and m, you may specify any integer from 0 to 9. VSE/POWER produces no separator pages or cards if your specification is 0 or if you omit the operand.

You can override a JSEP specification by an * $$ LST statement for printed output and by an * $$ PUN statement for punch output. You can also override a JSEP specification for any VSE/POWER writer task by coding the SEP|NOSEP operand of the PSTART command.

For printed output, VSE/POWER prints one separator page more than is specified. Each separator page contains the following information:

- VSE/POWER job name and number.
- User information from the * $$ JOB statement or from the * $$ LST statement.
- A two-digit sequence number in block letters, starting with 1 and increasing by one for each job (for local tasks only). If the sequence number exceeds 99, it starts again with 1. The sequence number is increased also for jobs not requesting separator pages.
- The node and user identifier of the target destination.
- The node and user identifier of the originator.
- The name of the execution node.
- The job number on the originating node.
- The device address of the printer used for output printing, or the line address of the RJE station used for printing, or the external device name to which the output was passed.
- The form number.
- The name of the FCB-image phase.
- The total number of lines for the output or output segment.
- The output class.
- Programmer information (if available):
  - Name of programmer
  - Building number
  - Room number
  - Department number
  - The SYSID of the executing system (if the shared spooling function is used).
  - The identifier of the partition in which the originating job was executed, if this information is available.

- START or END indication for heading or trailing separator pages.
- Job name, job number, and segment number of the output.

  Note: On separator pages that delimit the last or only segment of printed output, the segment number reads LAST or ONLY, respectively.

- The date and time the output was printed.
- The identification of the spooling system, including version and release level.

If the page length defined for your output is less than 45 lines, VSE/POWER cannot print a job identification using expanded letters.

For punched output, VSE/POWER provides:

- Depending on your specification of m, two to eight cards with 12-11-0-8-9 punches in all columns are placed as start separator cards before the job's punched output:

m	Cards
1	2
2	2
3	2
4	3
5	4
...	...
9	8

  Note: Two blank cards are used as end separator cards.

- One card with the VSE/POWER job name punched such that it can be read on the back of the card.

Note: Stacker selection is ignored if VSE/POWER is requested to include separator cards. In that case, VSE/POWER uses the default stacker.

If punched output is transferred to an RJE work station without the transparency feature, job separator cards are not readable. (The transparency feature is defined through the TRNSP operand of both the PRMT macro and the PLINE macros.)

LTAB=(d0,d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12)

The operand describes the carriage control tape or the forms-control buffer (FCB) image for printed output. VSE/POWER uses the values you specify as a default for spooling list output. You can override these values by specifying, in the * $$ LST statement, the name of an FCB-image phase or a different LTAB

specification. If you specify this operand, all entries in the list (d0 through d12) must be present.

Each entry of the list specifies the distance (in number of print lines) from the first carriage control tape channel punch to the other channel punches or to the end of the page:

d0 Specifies the distance from the last line on the page (channel-12 punch) to the first line on the next page (channel-0 punch).

d1 Specifies the distance between the first line (channel-0 punch) and the first skip-stop position (channel-1 punch).

d2 Specifies the distance between the channel-1 punch and the channel-2 punch.

d3 Specifies the distance between the channel-1 punch and the channel-3 punch.

And so on.

If channel 1 is the first punch, express the distance d1 as 00. If a channel is not represented by a skip, specify 00 for this channel position. Give an LTAB specification of all zeros if you want to avoid a channel-12 indication.

If you omit the operand, VSE/POWER uses a default as follows:

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

The default defines a 66-line carriage control tape (or FCB) with a distance of 10 lines between the last line on the page and the first line of the next page. The distance between channel punches 11 and 12 is 6 lines, the distance between all other channel punches is 5 lines.

For output on a printer with an FCB, ensure that your specification in this operand matches the printer's default FCB image. This ensures proper handling of output to this printer for which this default image is to be used. See also the discussion of the operands FCB and LTAB under "* $$ LST: Define List-Output Attributes" on page 256.

MEMTYPE=membertype

Specify this operand if you want VSE/POWER to support source code inclusion (SLI). The previously available SUBLIB operand is still supported for compatibility reasons.

For membertype, specify a one- to eight-character name which VSE/POWER is to use as the default member type in searching for a member if you have included an SLI statement of the form

```
* $$ SLI MEM=membername
```

in a VSE job stream. You may not use PHASE or DUMP for membertype.

If you have specified the MEMTYPE operand (but no SUBLIB operand) and you use an SLI statement of the form

```
* $$ SLI bookname
```

VSE/POWER will use A as default for sublib in the SLI statement (interpreted as membertype) with one exception: If you have specified a one-character membertype in the MEMTYPE operand and no SUBLIB operand is present, then this value is taken as default.

MRKFRM={YES|NO}

This operand applies to 3800 printers.

Specify MRKFRM=YES or omit the operand if the mark-form function of your IBM 3800 is to be used for producing start-job separator pages. In this case, VSE/POWER produces only one page and then issues the Mark-Form channel command. This causes the 3800 printer to reproduce three or five copies of the separator page and to print a line of dark marks on the perforation between these pages. These pages are followed by a sufficient number of blank pages to advance the last line of black marks beyond the printer's fuser station.

The MRKFRM option is activated only when job separator pages are to be printed.

VSE/POWER uses the mark-form function of the 3800 only for producing separator pages at the beginning or between copies of an output. At the end of an output, VSE/POWER produces the separator page(s) according to your JSEP specification.

If you specify MRKFRM=NO, VSE/POWER includes job separator pages according to the value specified in the JSEP operand or in the JECL statement * $$ LST.

MULT12={YES|NO}

Specify MULT12=YES if you plan to make use of multiple channel-12 posting. Once channel 12 has occurred, VSE/POWER signals the channel-12 condition to your program for each subsequent line until channel 1 on the next page has been reached.

If you specify MULT12=NO, explicitly or by default, the channel 12 condition is indicated to your program only once: when channel 12 has been reached.

Note: Multiple channel-12 punches may be specified in an FCB-image phase; however VSE/POWER recognizes only the first channel-12 location. Therefore, if you specify MULT12=YES for a printer with an FCB, your program's list output under VSE/POWER may be different from this output when your program runs outside VSE/POWER control.

NTFYMSG=nnn

Specify this operand if notify support for VSE/ICCF is to be used.

For nnn, specify the maximum number of messages that VSE/POWER is to hold in the VSE/ICCF notify queue. This number can be any value from 1 to 999. However, sufficient GETVIS space must be available to VSE/POWER to store the messages.

If you specify this operand and also the NTFY operand in the JECL statement * $$ JOB, VSE/ICCF receives a notify message when:

- Execution of a job submitted by a user of VSE/ICCF is complete.
- Output, destined for a VSE/ICCF user, has arrived from the network.
- A job or output owned by a VSE/ICCF user has been transmitted to the next node on its path to the ultimate destination.

PAUSE={NO|YES}

Specify PAUSE=YES if message 1Q40A is to be displayed before VSE/POWER starts to write punched output to the defined punch device. This gives your operator a chance to take cards, if any, from the stacker of the punch device. VSE/POWER does not start processing the affected output until your operator issues a PGO command.

Specify PAUSE=NO or omit the operand if you do not want VSE/POWER to wait.

PNET=phasename

Specify this operand if the networking function (PNET,BSC or PNET,SNA) is to be supported. For phasename, insert the name you assigned to the applicable set of PNODE macros.

You can override the name by using the autostart statement SET PNET.

PRI={3|n}
The operand specifies the default priority of a job entry when no priority is specified in the * $$ JOB statement. For n, you can specify any digit from 1 to 9, where 9 is the highest priority.

RBS={(0,0)|(n,m)}
This operand is valid only for output that is spooled to disk.

Specify RBS=(n,m) if VSE/POWER is to begin an output segment after having processed n list pages or m card-image records for a job. For n and m, you can specify any number up to 999999. If you specify 0 or omit the operand, no record-count controlled segmentation takes place.

Note: For segmentation on an IBM 3525, and when using print commands, either all segments must be punched out in sequence or separator cards must be specified. This avoids losing the last two print lines.

RDREXIT={NO|localexit|([localexit],pnetexit)}
Specify this operand if a user-written reader exit routine is to be loaded into the VSE/POWER partition. In the operand, specify for

localexit
The name of the routine that is to be used for processing local input;

pnetexit
The name of the routine that is to be used for processing input from the network.

Input received from the network is not passed to the exit routine whose name you specify for localexit. Likewise input received from a local input reader is not passed to the exit routine whose name you specify for pnetexit.

The names specified for localexit and pnetexit may be identical.

Appendix B of VSE/POWER Application Programming discusses the coding of a reader-exit routine and gives a coding example.

RJEBSC={YES|NO}
Specify RJEBSC=YES or omit this operand if VSE/POWER should run with RJE,BSC support active.

Specify RJEBSC=NO if you plan to use the PNET,BSC function but want to avoid having RJE,BSC phases loaded. This saves virtual storage.

For this operand to take effect, submit at least one PLINE generation macro.

SHARED={NO|(Q[,A])}
This operand applies only to the shared spooling function.

Specify Q,A if the account file is to be shared in addition to the VSE/POWER queues (RDR, LST, PUN and XMT). If you specify only Q, then a separate account file is to be defined for each of the sharing systems that requires VSE/POWER accounting support.

SNA={YES|([wscount|prmtno],[password],[applid|POWER])}
The operand specifies control values for the RJE,SNA support as follows:

SNA=YES
RJE,SNA support is to be initialized in accordance with those of your PRMT generation macros which include TYPE=LUT1. Specify SNA=YES if this support is required but none of the other (optional) operands need be specified.

wscount
For wscount, specify the number of work stations that may be logged on concurrently. You can specify any value from 0 to 250. When you supply this value, consider the following:

- If you supply an invalid specification, VSE/POWER takes the default action. It sets this value to the total number of SNA work stations defined by PRMT macros or to 250, whichever is the smaller.
- If you specify a value smaller than the total number of work stations defined by PRMT generation macros, VSE/POWER uses the specified value. This can prevent a work station user from logging on to your system.
- If you specify a value greater than the number of work stations defined by PRMT generation macros, VSE/POWER sets the value equal to the number of defined work stations.

prmtno
The default if wscount is not specified; for prmtno, give the number of PRMT macros that specify TYPE=LUT1.

password
The password which VSE/POWER is to use when it issues an OPEN call for an ACF/VTAM ACB. This password must be identical to the one you specified in the PRTCT operand of the ACF/VTAM APPL statement. If you did not specify a password in the APPL statement, VSE/POWER ignores the password specified here.

applid
For applid, specify the name by which you defined VSE/POWER to ACF/VTAM as an application (when you set up the ACF/VTAM API).

Omitting Specifications: If you want to omit one or more specifications within the parentheses, be sure to code the delimiting comma, except when the omission occurs at the end of the list. Examples:

```
SNA=(128,,SPOOL)
SNA=(128)
SNA=(128,SPAPPLPW)
```

SPLIM={90|nnn}

Specifies the percentage of used spool space which, when exceeded, causes the central operator to be notified via message 1QF0I (at most every 60 seconds.) If this message occurs frequently, the operator can take appropriate action (for example, delete or offload queue entries).

You can obtain the percentage of spool space currently in use by issuing the PDISPLAY Q command.

For nnn, you can specify any value between 0 and 100. If you specify 0, no notification will take place.

SPOOL={YES|NO}

Specify SPOOL=YES if any of your application programs uses the XECB-based SPOOL-macro support with the macros CTLSPOOL, GETSPOOL, and PUTSPOOL.

STDCARD={(0,0)|(n,m)}

In the operand, n specifies the number of cards that may be punched out before VSE/POWER issues a warning message to the operator; m specifies the additional number of cards that may be punched out before VSE/POWER reissues the same message. For n and m, you may specify any number from 0 to 999999.

A specification of 0 in the first entry means that no warning will be given. If one of the values is omitted, the specified value is used for both n and m; for example:

```
STDCARD=(,800)  has the same effect as STDCARD=(800,800)
STDCARD=(10000) has the same effect as STDCARD=(10000,10000)
STDCARD=12000   has the same effect as STDCARD=(12000,12000)
```

STDLINE={(0,0)|(n,m)}

In the operand, n specifies the number of lines that may be printed before VSE/POWER issues a warning message to the operator; m specifies the additional number of records that may be printed before VSE/POWER reissues the same message. For n and m, you may specify any number from 0 to 999999.

The rules for specifying STDCARD=(n,m) apply accordingly to this operand.

SUBLIB={NO|sublib}
This operand is available to ensure compatibility with previous releases of VSE/POWER. Consider using the MEMTYPE operand instead.

For sublib specify a one-character name which VSE/POWER is to use as the default for sublib in the SLI statement (interpreted as member type) in searching for a member if you have included an SLI statement of the form

```
* $$ SLI bookname
```

in a VSE job stream.

If you have specified the SUBLIB operand (but no MEMTYPE operand) and you use an SLI statement of the form

```
* $$ SLI MEM=membername
```

VSE/POWER uses SOURCE as default for membertype.

If you do not specify the MEMTYPE operand and specify SUBLIB=NO (or omit this operand), no SLI support is generated.

SYSID=n
This operand applies only to the shared spooling function; it is required if the SHARED operand is specified.

For n, specify the identification, a one-digit number from 1 to 9, of the system on which your VSE/POWER is to run. VSE/POWER uses this identification in all account records and in its responses to PDISPLAY requests.

Each of the sharing processors must have a unique system identification assigned.

You can change the specified identification by SET SYSID=sysid in the autostart procedure.

TIME={(5,0,60)|([t1],[t2][,t3])}
This operand applies only to the shared-spooling function.

The specified values need not be the same for all of the sharing systems. Define these values in consideration of characteristics such as the operating speeds of the sharing processors, their I/O devices, and their job mix. Specify for:

t1 The active time in number of seconds. This is the time your VSE/POWER is given update control of the spool files. You can specify a value from 1 to 99.

t2 The idle time in number of seconds. This is the time your VSE/POWER must wait before it can issue another update-control request for the spool files. You can specify a value from 0 to 9. Normally, t2 should be specified as 0.

This specification may be useful if processors of different operating speeds are involved. You can use the specification to give another VSE/POWER (on the slower processor) a chance to obtain update control of the spool files.

t3 The "polling time" (in number of seconds) for your VSE/POWER when none of its tasks are active. VSE/POWER on another processor may have queued jobs that must be processed by your VSE/POWER. The tasks of your VSE/POWER must be activated in order to search for work to do. You can specify a value from 1 to 999.

The time required to do this search is very short; if there is no work, your VSE/POWER returns to the wait state until the time interval t3 expires again.

TRACESZ={12|nnn}

The operand specifies the number of K-bytes (up to 252) to be reserved for the telecommunication-trace function. If you omit this operand but specify RJEBSC=YES or PNET=phasename, VSE/POWER reserves an area of the default size.

PLINE Macro

The macro and its operands define the hardware characteristics of a BSC line - a control unit such as an integrated communication adapter, a transmission control unit, or an IBM 370x in emulation mode.

The macro must be coded if support for remote job entry (RJE) or networking (PNET) is required and one or more of the connections are via BSC lines. The macro does not apply if only SDLC lines are used (RJE,SNA or PNET,SNA) or if networking is done via the channel-to-channel attachment (CTCA) support under VM. One PLINE macro must be coded for each BSC line that is to be supported by VSE/POWER.

VSE/POWER supports up to 100 BSC lines, which may be RJE,BSC or PNET,BSC or any combination of the two. The PLINE macro(s) must follow the POWER macro and precede the PRMT macro(s).

Format of the Macro

Name	Operation	Operands
	PLINE	ADDR=X'cuu'
		[,CODE={EBCDIC|ASCII}]
		[,INTRPT={NO|YES}]
		[,MODSET={AA|AB|BA|BB}]
		[,PSWRD=password]
		[,SWITCH={NO|YES}]
		[,TIMEOUT={NO|n}]
		[,TRNSP={YES|NO}]

Description of the Operands

ADDR=X'cuu'
: The operand specifies the line address (channel and unit number) of the control unit. This address is the same as the one used to identify the control unit to VSE.

CODE={EBCDIC|ASCII}
: The operand specifies the transmission code, which may be either EBCDIC or ASCII.

INTRPT={NO|YES}
: Specify YES if a selector channel is to run in interrupt mode. The specification causes VSE/POWER to free the channel as soon as possible. However, if VSE/POWER gets a time out during the initial I/O, any dial-in connection is lost and the connection has to be dialled again.

MODSET={AA|AB|BA|BB}
The operand defines the characteristics of an IBM 2701 control unit by a two-letter combination as shown.

First letter:
It specifies interface A or B when a dual-communication interface is used.

Second letter:
It specifies code A or B, when the dual-code feature is installed.

For more details about the interfaces and the codes, refer to the component description of the IBM 2701 Data Adapter Unit.

PSWRD=password
The operand defines the password which a remote operator must specify for sign-on of a remote work station via this line. The password must be an alphameric string of up to eight characters. If the macro defines an RJE,BSC line, the specified password may be overridden (until the next PSTOP command) by the central operator in a PSTART command.

This operand applies also to the processors in a networking environment. The defined password must be the same as the password which the other system sends during the sign-on processing. If the passwords do not match, no connection is established.

VSE/POWER sends the defined password to the other node if no line password is specified in the PSTART PNET,... command.

SWITCH={NO|YES}
Specify YES if a switched-line connection is used. Specify NO or omit the operand for leased-line operation.

TIMEOUT={NO|n}
For an RJE,BSC line, specify for n the number of minutes the work station or line is allowed to remain idle before sign-off is forced. The specification becomes effective after sign-on is complete and when no job or output data is being transferred.

You can specify any time-out value from 1 to 255. If you specify 0 or NO, or if you omit this operand, the work station or line is allowed to remain idle until the line is stopped using the PSTOP or PEND command.

Note: Before sign-on, VSE/POWER disconnects if no sign-on occurs within about one minute:

- After the first data byte has been received via a leased line.

- After connection has been established via a switched line.

In either case, VSE/POWER restarts the line set-up procedure right after completion of the disconnect request. For a switched line, however, the connection is dropped and has to be set up anew.

A forced sign-off can save line charges on switched lines. It can save processor charges for a remote user because, without a forced sign-off, the system processes timeouts every three seconds. Moreover, an idle work station keeps a switched line busy.

The specification of a timeout can be useful if there is a line problem. It causes a line to be stopped automatically if, for this line, only timeout counts are processed until the elapse of the specified time.

For a PNET,BSC line, specify for n the number of minutes during which this node attempts to contact the other node. You can specify any time-out value from 1 to 255. The specification is effective during the processing of a PSTART request.

For leased lines, the counting of the specified time begins when the PSTART for the other node is issued; for switched lines, the counting begins when this node (your own) is being called.

If you specify 0 or NO, or if you omit the operand, VSE/POWER attempts to contact the other node every three seconds until PSTOP or PEND is issued. For a switched line, VSE/POWER does this after having been called.

For switched lines you should always specify a TIMEOUT limit. This avoids that VSE/POWER tries to contact a caller over and over again in case of a wrongly dialled connection.

When sign-on is complete, VSE/POWER uses a time-out value of 90 seconds to control forcing off idle users.

TRNSP={YES|NO}

Specify YES or omit the operand to ensure that all data is transmitted as presented to the line (a way of transmitting data called transparency). The transmission of a language translator output, for example, requires transparency.

TRNSP=YES is required if PNET,BSC support is to use the line. To use the transparency feature for RJE,BSC, you must specify TRNSP=YES also in the PRMT generation macro.

PRMT Macro

The macro defines the hardware characteristics of an RJE work station; it identifies the user; it describes to where output is to be routed.

At least one PRMT macro must be supplied for each remote work station, and all PRMT macros must be submitted behind the PLINE macro(s), if present.

The PRMT macros must be submitted in ascending order of the work-station identifiers, and your definitions for RJE,BSC work stations must precede the ones for RJE,SNA work stations. Therefore, the lowest RJE,SNA identifier must be higher than the highest RJE,BSC identifier.

If an operand is specified for a work station which does not have the specific hardware feature, this operand is ignored.

Format of the Macro

Name	Operation	Operands	Used for	
[name]	PRMT	REMOTE=nnn,	SNA	BSC
		TYPE={2770|	SNA	BSC
		2780|		
		3741|		
		3780|		
		LUT1}		
		[,ABE=YES|,BE=YES]		BSC
		[,CMPACT=name]	SNA	
		[,CONSOLE={YES|<u>NO</u>}]	SNA	
		[,CS={YES|NO}]		BSC
		[,CSALST={<u>DC1</u>|		BSC
		DC2|		
		DC3|		
		0}]		
		[,CSAMSG={DC1|		BSC
		DC2|		
		DC3|		
		0}]		
		[,CSAPUN={DC1|		BSC
		<u>DC2</u>|		
		DC3|		
		0}]		
		[,HFC={YES|NO}]		BSC
		[,LIST=n]		BSC
		[,LSTROUT={remid|	SNA	BSC
		<u>own-nnn</u>}]		
		[,LU=(name,name,...)]	SNA	
		[,MRF={YES|<u>NO</u>}]		BSC

Format of the Macro (cont'd):

Name	Operation	Operands	Used for
		[,MSG=nnn]	BSC
		[,MSGEJCT={YES|NO}]	BSC
		[,MSGSPCE={YES|NO}]	BSC
		[,PSWRD=password]	SNA
		[,PUN={80|nnn}]	BSC
		[,PUNROUT={nnn| own-nnn}]	SNA BSC
		[,SCE={YES|NO}]	BSC
		[,SESSLIM={n|1}]	SNA
		[,TRNSP={YES|NO}]	BSC
		[,TURNEOJ={YES|NO}]	BSC
		[,XLATE={YES|NO}]	SNA

Format of the Short Form

Use the available short form of the macro if the characteristics of a work station to be defined are identical to those specified in a preceding PRMT macro (see also the explanation for the REF=nnn operand). The format of this short form is:

Name	Operation	Operands	Used for
[name]	PRMT	REMOTE=nnn,	SNA BSC
		,REF=nnn	SNA BSC
		[,LSTROUT={remid| own-nnn}]	SNA BSC
		[,PUNROUT={remid| own-nnn}]	SNA BSC

Description of Operands

REMOTE=nnn

For nnn, specify the remote identifier for the RJE work station whose hardware characteristics are to be defined. As an identifier, you can specify any number from 1 to 250.

TYPE={2770|2780|3741|3780|LUT1}

The operand specifies the type of work station.

For an IBM 3770, specify TYPE=2770 or 3780 depending on the buffer size.

For an SNA work station, specify LUT1.

ABE=YES

The operand specifies additional buffer expansion for the 2770 and 3741. The size of the buffer is then 512 bytes.

BE=YES

The operand specifies buffer expansion for an IBM 2770. If you specify BE=YES, the size of the buffer is 256 bytes.

BE=YES will be overridden by a specification of ABE=YES.

CMPACT=name

The operand applies only to an SNA work station.

For name, specify the name of the compaction table set to be used if no such set is specified in an * $$ LST statement. The name you specify is the one you use in the name field of the PCPTAB macro for generating the table set.

Data compaction can be requested by way of an * $$ LST statement even if no compaction is defined in this macro.

CONSOLE={YES|NO}

The operand applies only to an SNA work station.

Specify YES if the work station has a line printer in addition to the normal work station console. This should always be specified for an IBM 3790.

The specification causes the transmission of printed output to be interrupted when messages are generated for the remote operator and no free session is available with the work station. A free session with the work station can be available only in a multiple logical unit (MLU) environment; in this environment, messages are always sent when a free session is found, even if CONSOLE=NO is specified.

For a single logical unit, specifying NO prevents VSE/POWER from interrupting the transmission of print output just for the transmission of messages. Instead, VSE/POWER collects the messages and submits them to the operator at the end of this printer flow.

If the work station does not have an additional line printer and you specify CONSOLE=YES, the printout at the work station will have interspersed operator messages.

CS={YES|NO}

This operand applies to a BSC work station other than IBM 2780 and 3741.

The operand specifies component selection. Its effect depends on your specifications for the operands CSALST, CSAMSG, and CSAPUN as discussed later. The default for CS is:

YES for an IBM 2770
NO for an IBM 3780

You define the work station component that is to be selected as output device by specifying "component select characters" in the operands CSALST, CSAMSG, and/or CSAPUN (see below). You may, if this is acceptable, use the defaults instead.

Normally, your specification selects an output device as follows:

DC1 - The printer
DC2 - The punch
DC3 - The diskette

However, the device that is actually selected depends on the hardware configuration at the work station.

Your specification for this and the associated CSxxx operands should not result in one of the following:

Print output to be routed to a punch.
Message output to be routed to a punch.
Punch output to be routed to a printer.

Component selection does not take place if you specify one of the following:

CS=YES and 0 in CSALST, CSAMSG, and CSAPUN.
CS=NO.

In either case, VSE/POWER routes the output according to the existing hardware settings (for example, to the diskette if the DISK switch on the IBM 3776 is ON).

CSALST={DC1|DC2|DC3|0}
This operand is valid only if CS=YES is specified. The operand specifies a component select character for a printer attached to the work station. For information about specifying a select character, refer to the discussion of the CS operand above.

CSAMSG={DC1|DC2|DC3|0}
This operand is valid only if CS=YES is specified. The operand specifies a component select character for messages to be sent to an output device. For information about specifying a select character, refer to the discussion of the CS operand above.

If you do not specify this operand, the component select character specified in or defaulted by the CSALST operand is used.

CSAPUN={DC1|DC2|DC3|0}
This operand is valid only if CS=YES is specified. The operand specifies a component select character for a punch device attached to the work station. For information about specifying a select character, refer to the discussion of the CS operand above.

HFC={YES|NO}
This operand applies only to a BSC work station.

Specify YES to avoid the transmission of blanks within a print line by using horizontal tab (HT) characters.

For an IBM 2770 or 2780 device, the horizontal format control (HFC) feature is optional, and the default is NO. For the IBM 3780, the feature is standard and the default is YES. The HFC feature is not supported by the IBM 3741.

If you specify also SCE=YES, VSE/POWER ignores your HFC=YES specification.

LIST=n
This operand applies only to a BSC work station.

For n, specify the number of print positions per line. You can specify any value from 80 to the maximum number of print positions or 150, whichever is larger. For the devices listed below, VSE/POWER accepts maximum values as indicated and uses default values as shown:

IBM Device Type	Max. Value	Default
2770	144 (see Note 1)	120
2780	144	120
3741	132 (see Note 2)	126
3780	144	120

Notes:

1. To use 132 or more print positions per line on an IBM 2770, specify also BE=YES or even ABE=YES.

2. To use more than 126 print positions per line on an IBM 3741, specify also ABE=YES.

LSTROUT={remid|own-nnn}
The operand tells VSE/POWER where to route by default the list output of jobs submitted at the work station. If you omit this operand, VSE/POWER routes the output to this work station.

LU=(name,name,...)
This operand applies only to an SNA work station.

The operand specifies a list of secondary logical unit names that may log on using the remote identifier specified in REMOTE=nnn. The names you specify must be in accordance with your definitions for ACF/VTAM.

If you use this operand, VSE/POWER rejects any LOGON request from the defined work station if this request uses a logical unit name not defined in this operand.

If you omit this operand, no logical unit name test is performed during the LOGON procedure.

Note: The number of names specified for one work station may be larger than the number specified in the SESSLIM operand. This number is limited only by the VSE assembler restrictions.

MRF={YES|NO}
This operand applies only to a BSC work station of the type IBM 2780.

The operand specifies that the multiple record feature of the IBM 2780 is to be used. As a result, the maximum number of records per block is seven instead of two.

MSG=nnn
This operand applies only to a BSC work station.

In the operand, nnn specifies the number of print positions per line for the device selected by the CSAMSG operand. You can use any number from 80 to 150 as in the LIST operand. If you omit this operand, VSE/POWER uses the value specified in or defaulted by the LIST operand.

MSGEJCT={YES|NO}
This operand applies only to a BSC work station.

MSGEJCT=NO suppresses a skip to the channel-1 line position between the last printed output and the first (or only) message if both output and messages are to be printed on the same output device of the work station.

MSGSPCE={YES|NO}
This operand applies only to a BSC work station.

MSGSPCE=NO suppresses the transmission of four contiguous Space-3 commands following the transmission of a message to the work station.

PSWRD=password
This operand applies only to an SNA work station.

The operand defines the password that has to be specified at the remote work station during LOG ON as part of the user data of the LOGON command.

If you specify an invalid password (longer than eight characters), the assembler sets the password field to blanks and issues an MNOTE (message) in the assembly listing.

PUN={80|nnn}

For nnn, specify the maximum length of the records that are to to be transmitted to the device selected by the CSAPUN operand. You can specify any number from 80 to 150.

Note: To use more than 126 punch positions per punch record on an IBM 3741, specify also ABE=YES; otherwise the PUN value will be reset to its default.

PUNROUT={remid|own-nnn}

The operand tells VSE/POWER where to route by default the punch output of jobs submitted at the work station. If you omit this operand, VSE/POWER routes the output to this work station.

REF=nnn

The operand, which applies only to the macro's short form, specifies the identifier given in another PRMT macro for REMOTE=nnn. It indicates that this (short form) macro defines a work station whose characteristics are the same as those defined by the referenced PRMT macro. The value for nnn in the REF operand must be lower than the value for nnn in the REMOTE operand of your short-form macro.

Operands other than REMOTE, LSTROUT, and PUNROUT are ignored if you supply a PRMT macro with the REF operand

Note: Check the referenced PRMT macro instruction for correct assembly. No warning is issued in the referring macro if the referenced macro is in error.

SCE={YES|NO}

The operand applies to an IBM 2770 or 3780. To use this operand for an IBM 3741, ABE=YES is also required.

SCE=YES specifies space compression/expansion. If you specify SCE=YES, you cannot specify also TRNSP=YES.

SESSLIM={n|1}

The operand applies only to an SNA work station.

For n, specify how many sessions may be logged on at the SNA work station that is being defined. You can specify up to six sessions per work station.

TRNSP={YES|NO}

This operand applies only to punch output destined for RJE,BSC work stations.

The operand specifies that the transparency feature is available at the work station. The feature is required for the transmission of punch output that includes "non-printable" characters (character values lower than X'40').

If you specify TRNSP=YES, you cannot specify also SCE=YES.

Specify TRNSP=NO or omit the operand if neither object decks nor "non-printable" characters are to be transmitted to the work station. The specification results in all characters between X'00' and X'40' to be converted to:

Zeros - If the receiving work station is an IBM 3741.
Blanks - If the receiving work station is not an IBM 3741.

To use the transparency feature, you must specify TRNSP=YES also in the PLINE generation macro.

Note: For print data which includes character values lower than X'40', the same rules as for TRNSP=NO apply, regardless of whether YES or NO is specified.

TURNEOJ={YES|NO}

The operand applies only to a BSC work station.

Specify TURNEOJ=YES if line turnaround (to switch from write to read at the end of an output operation) is to take place. The specification causes output to continue if no further input is available.

Specify TURNEOJ=NO if no line turnaround should take place.

The default value depends on the type of the work station as shown:

Type of Work Station	A Default of
IBM 2770	YES
IBM 2780	YES
IBM 3741	NO
IBM 3780	YES

XLATE={YES|NO}

The operand applies only to an SNA work station.

Specify XLATE=YES if print output sent to the work station is to be transmitted with characters X'01' through X'3F' translated to blanks. This may be required by certain printers attached to the work station.

If YES is specified, you may not code your own single character string (SCS) characters in the print output.

PNODE Macro

The macro defines an entry of your system's network definition table (NDT). Code a separate PNODE macro for your own node and for each of the nodes in the network to which your system transmits data or from which your system receives data. Defining an NDT is discussed in more detail in the publication VSE/POWER Networking User's Guide.

Format of the Macro

Name	Operation	Operands	Used in
phasename	PNODE	NODE=nodeid1	All
		[,APPLID=name]	SNA only
		[,AUTH={NOJOB|JOB|NET}]	Non-local
		[,BUFSIZE={size|400}]	Non-local
		[,LOCAL={YES|NO}]	Local
		[,MAXBUF={(recv,trans)|(3,3)}]	Non-local
		[,PWD=password]	All
		[,ROUTE1=nodeid2]	Non-local
		[,ROUTE2=nodeid3]	Non-local

For phasename, specify the name you want to be assigned to your NDT table. Use this name in the first of your set of PNODE macros and, if you have only one NDT, also in the PNET operand of the POWER generation macro. The name you use must be different from that specified in any of the macro's NODE operand.

The name you supply can have up to eight alphameric characters. The first character of the name must be one of the letters A through Z or one of the characters $, #, and @.

Description of Operands

NODE=nodeid1

For nodeid1, specify the name of the node to which the specifications in this PNODE macro apply; it is the name by which this node is known in the network.

If you use shared spooling, all of your sharing systems must be defined to VSE/POWER with the same node name in the PNODE macro with LOCAL=YES.

The name you supply can have up to eight alphameric characters; the first character must be one of the letters A through Z or one of the special characters $, #, and @.

APPLID=name

The operand applies to and is required for a node using SNA support.

For name, specify the application identifier that is to be used to establish an ACF/VTAM controlled session. The name you specify can have up to eight alphameric characters; embedded blanks are not allowed.

In the macro defining your own node (LOCAL=YES is specified), your specification for name identifies VSE/POWER to ACF/VTAM.

In the macro defining another node (LOCAL=NO is specified or defaulted), this specification identifies the ACF/VTAM application with which your VSE/POWER is to communicate.

Notes:

1. The operands APPLID and ROUTE1 cannot be specified together.

2. If APPLID is not specified for the local node, VSE/POWER uses as application identifier the name specified for nodeid1 in the macro's NODE operand.

3. The APPLID operand may not specify the same name as the SNA operand of the POWER macro; it should not be POWER because this is the default if you omit the SNA operand of the POWER macro.

AUTH={JOB|NET|NOJOB}

The operand applies if the macro defines another node (LOCAL=NO is specified or defaulted); it defines for the operator at the affected node the level of authorization for controlling operations at your own node. An attempt of the other node's operator to enter a VSE/POWER command outside the defined authorization level is rejected. The operator at your own node always has the right to control his/her system and all jobs and output in its queues.

AUTH=JOB

If specified (or defaulted) authorizes the operator at the other node (defined by NODE=nodeid1) to alter job characteristics of queue entries at your own node as follows:

- Jobs originated at this other node.
- Output destined for this other node.
- Output produced by a job originated at this other node.

AUTH=JOB allows the operator at the other node to request general display functions to be performed.

AUTH=NET
Allows the operator at the other node to:

- Issue networking commands.
- Manipulate all queue entries in the queues of your own node, regardless of ownership.

AUTH=NOJOB
Avoids that the operator at the other node manipulates any jobs or output in the queues of your own node.

For more information about command authorization for networking, refer to the publication VSE/POWER Networking User's Guide.

BUFSIZE={size|400}
The operand applies if the macro defines another node (LOCAL=NO is specified or defaulted).

For size, specify (in number of bytes) the length of each of the VSE/POWER transmission buffers. You can specify a value from 300 up to:

1,800 bytes	For transmission via a BSC line or with channel-to-channel attachment (CTCA) support.
32,000 bytes	For transmission via an SNA link.

If you specify an odd value, VSE/POWER rounds it to the next higher even number. For more information about these buffers and their use, refer to the VSE/POWER Networking User's Guide.

During the sign-on procedures, communicating systems agree on a common buffer size. If the sizes of their buffers disagree, the systems use buffers of the smaller size.

LOCAL={YES|NO}
Specify LOCAL=YES if the macro defines your own node, that is, an NDT entry for your own system. Only one specification of LOCAL=YES is allowed for the generation of a network definition table.

Specify LOCAL=NO or omit the operand if the macro defines another node.

Note: The local node name may not be changed without offloading the queues and a subsequent cold start.

MAXBUF={(recv,trans)|(3,3)}
The operand applies if the macro defines another node (LOCAL=NO is specified or defaulted). The operand specifies the number of buffers which may be used by each PNET-receiver and transmitter task.

For recv, give the number of input buffers for each of the PNET-receiver tasks servicing the applicable other node. The minimum value is 2, the maximum value is 255.

For trans, give the number of buffers for each of the PNET-transmitter tasks servicing the applicable other node. The minimum value is 1, the maximum value is 255.

For more information about these buffers, refer to the publication VSE/POWER Networking User's Guide.

PWD=password

If LOCAL=NO, specify for password the (optional) node password to be used for checking a connection request from the node that is defined by the macro. The password you specify can have up to eight alphameric characters.

If LOCAL=YES, specify for password the node password that is to be transmitted to all other nodes during the connection process (except if a node-password is given in the PSTART command). For more information about processing passwords, refer to the publication VSE/POWER Networking User's Guide.

ROUTE1=nodeid2

The operand applies if the macro defines another node (LOCAL=NO is specified or defaulted) and no direct link exists to that node. The operand specifies the name of the first node that is to be used for transmitting data or messages from your own node to this other node.

The node whose name you specify for nodeid2 must itself be defined by an entry in the NDT as a direct link (without any of the operands ROUTE1 or ROUTE2).

This operand and the operand APPLID and cannot be specified together.

ROUTE2=nodeid3

The operand specifies the name of a node as a route alternate to the primary one specified in the ROUTE1 operand. VSE/POWER uses this alternate route for transmission if the primary route is not signed on. The line discipline for the ROUTE2 node may be the same as or different from that of the ROUTE1 node.

The node whose name you specify for nodeid3 must itself be defined by an entry in the NDT as a direct link and without the operand ROUTE1.

PCPTAB Macro

Use the macro to define a compaction table set for the RJE,SNA support. Any number of compaction table sets may be generated for a specific VSE/POWER, and each set is identified and referred to by its name.

Required compaction table sets need not be generated during VSE/POWER table generation; this can be done at any time as the need arises. For more detailed information about building a compaction table set see Appendix A.

You can suppress compaction for an individual job by supplying an * $$ LST statement which specifies CMPACT=NO.

Format of the Macro

Name	Operation	Operands
[name]	PCPTAB	MASTER=(mc1,mc2,mc3[,...,mc16]) [,NOMAST1=(nmc(1),...,nmc(n))] [,NOMAST2=(nmc(n+1),...,nmc(p))] [,NOMAST3=(nmc(p+1),...,nmc(q))]

For name, specify the name of the compaction table set. This name can have up to four alphameric characters, and its first character must be a letter of the alphabet or one of the characters $, #, and @. The name you use becomes the phase name of the table set. If you do not specify a name, the phase name will be PCPT by default.

Description of Operands

MASTER=(mc1,mc2,mc3[,...,mc16])
: The operand specifies your set of master characters (mc). You can specify from 3 to 16 master characters in any sequence and separated by a comma. You may specify a master character as a character or as a hexadecimal value. Example:

  MASTER=(A,B,40,D,7F).

  For more details about the selection of master characters, refer to Appendix A.

NOMAST1=(nmc(1),...,nmc(n))
: The operand specifies your set of non-master characters (nmc).

  The number of non-master characters to be specified depends on the number of master characters defined in the MASTER operand.

If:

m = 16, then nm = 0
m < 16, then nm = 256 - m(m + 1).

where: m = Number of master characters.
nm = Number of non-master characters

You can specify non-master characters in any sequence and separated by a comma. You may specify a non-master character as a character or as a hexadecimal value. Example:

NOMAST1=(X,N,3A,1C,B0,H).

The maximum length of a character string that can be specified in a NOMSTn operand is 255 bytes; this number includes the separating commas and the enclosing parentheses. However, you may continue the non-master character specifications of the NOMAST1 operand, for example, by using the NOMAST2 operand, even if you have not coded the maximum string length in the NOMAST1 operand.

You may, if this is convenient for you, code an operand on two or more lines, following the assembler rules for macro continuation.

For more details about the selection of non-master characters, refer to Appendix A.

NOMAST2=(nmc(n+1),...,nmc(p))
See the description of the NOMAST1 operand, above.

NOMAST3=(nmc(p+1),...,nmc(q))
See the description of the NOMAST1 operand, above.

Rules for Character Specifications

A number of rules apply to the character specifications of the MASTER and NOMASTn operands. These rules are:

- A master or non-master character may not be specified more than once.
- The following characters must be coded within apostrophes when specified in character notation:

Left parenthesis:	'('	or hexadecimal 4D
Right parenthesis:	')'	or hexadecimal 5D
Comma:	','	or hexadecimal 6B
Blank:	' '	or hexadecimal 40

- The following characters, if specified in character notation, must be specified twice but count as one:

Ampersand:	&&	or hexadecimal 50
Quote:	''	or hexadecimal 7D

- For performance reasons, forms-feed and new-line characters (FF or X'0C' and NL or X'15', respectively) should be specified also as non-master characters. Select-channel characters (SELn or X'04' with X'81' through X'89' and X'7A' through X'7C') and the carriage-return (CR or X'0D') character should be specified as required.

Link-edit the resulting object module into a sublibrary which is defined as accessible to VSE/POWER.

Examples of VSE/POWER Generation Macros

The following examples show macro sequences for generating your VSE/POWER control tables. The examples are for:

- VSE/POWER only.
- VSE/POWER with both RJE,BSC and RJE,SNA support.
- VSE/POWER with RJE,SNA support only.
- VSE/POWER with PNET,BSC support only.

In the examples, the required continuation characters in column 72 are not shown in order to have room for explanations.

Example 1: VSE/POWER Only

```
POWEX   POWER   ACCOUNT=YES,      Accounting support
                DBLKGP=8,         DBLK group size
                SPLIM=80,         Spool limit value
                JSEP=(2,2),       Job separators
                NTFYMSG=100       VSE/ICCF notify support
        END
```

Example 2: VSE/POWER with RJE,BSC and RJE,SNA Support

```
NAME1   POWER SNA=(3)             Up to three work stations may
                                  be logged-on concurrently
        PLINE ADDR=X'cuu'         Defines BSC line
        PLINE ADDR=X'cuu'         Defines BSC line
        PLINE ADDR=X'cuu'         Defines BSC line
        PLINE ADDR=X'cuu'         Defines BSC line
        PRMT  REMOTE=1,           Defines BSC remote user
              TYPE=2780
        PRMT  REMOTE=2,           Defines BSC remote user
              TYPE=2770
        PRMT  REMOTE=3,           Defines BSC remote user
              TYPE=3780
        PRMT  REMOTE=4,           Defines BSC remote user
              TYPE=2770
        PRMT  REMOTE=5,           Defines SNA remote user
              TYPE=LUT1,
              CONSOLE=YES
        PRMT  REMOTE=6,           Defines SNA remote user
              TYPE=LUT1,
              CONSOLE=YES
        PRMT  REMOTE=7,           Defines SNA remote user
              TYPE=LUT1,
              CONSOLE=YES,
              SESSLIM=4,
              CMPACT=TEXT
        END
```

Example 3: VSE/POWER with RJE,SNA Support

```
NAME2   POWER SNA=(3,,POW)
        PRMT  REMOTE=1,
              TYPE=LUT1,
              CONSOLE=YES,
              SESSLIM=5
        PRMT  REMOTE=2,
              TYPE=LUT1,
              CONSOLE=YES,
              SESSLIM=2,
              CMPACT=TEXT,
              LU=(LU01,LU02)
        PRMT  REMOTE=3,
              TYPE=LUT1
        END
```

Example 4: VSE/POWER with PNET Support

```
NAME4   POWER  ACCOUNT=YES,      Accounting support
               DBLKGP=8,         DBLK group size
               SPLIM=80,         Spool limit value
               JSEP=(2,2),       Job separators
               PNET=NEWNDT,      PNET with NDT NEWNDT
               RJEBSC=NO         RJE,BSC support not required
        PLINE  ADDR=X'cuu'       Defines BSC line
        PLINE  ADDR=X'cuu'       Defines BSC line
        END
```

The macros below must be assembled and cataloged separately:

```
NEWNDT  PNODE  NODE=NODEA,       The NDT entry for the own node
               LOCAL=YES,
               PWD=NODEAPW
        PNODE  NODE=NODEB,       The NDT entry for node B
               AUTH=JOB,         Command authorization
               BUFSIZE=800,      800-byte transmit buffers
               PWD=NODEBPW
        PNODE  NODE=NODEC,       The NDT entry for node C
               AUTH=NET,         Command authorization
               BUFSIZE=800,      800-byte transmit buffers
               MAXBUF=(4,6),     4 receive and 6 transmit buffers
               PWD=NODECPW
        PNODE  NODE=NODED,       The NDT entry for node D
               AUTH=JOB,         Command authorization
               BUFSIZE=800,      800-byte transmit buffers
               ROUTE1=NODEB,     Primary routing node is NODEB
               ROUTE2=NODEC,     Secondary routing node is NODEC
               PWD=NODECPW
        END
```

VSE/POWER Generation Procedure

After having restored the VSE/POWER distribution tape, you can begin generating VSE/POWER control tables. The steps for generating these tables are:

1. Code VSE/POWER generation macros and operands as required.

2. Assemble these macros – This creates an object module that contains the code needed by VSE/POWER for start-up and operation.

3. Link-edit the module to the sublibrary to which you restored VSE/POWER during installation.

After generation, you can initialize VSE/POWER using the procedures described under "Starting VSE/POWER" on page 83.

CHAPTER 3. OPERATING WITH VSE/POWER

This chapter discusses the operation, at the console, of a VSE system with VSE/POWER. The chapter describes:

- The procedures for starting VSE/POWER - See "Starting VSE/POWER."
- The use of some of the VSE/POWER central operating commands - See "Controlling Operation" on page 96.
- The use of the IBM 4248 printer, the IBM multipurpose unit-record devices, and diskette I/O units - See "Specific Devices" on page 109.
- Tape processing - See "Tape Processing with VSE/POWER" on page 114.

For efficient operation of a VSE system in a networking environment, you should familiarize yourself with the information contained in the VSE/POWER publication Networking User's Guide.

Throughout this chapter, the information given for the IBM 3800 applies also to the IBM 3200 if not explicitly stated otherwise.

STARTING VSE/POWER

VSE/POWER may be started at any time like an application program. Normally, however, you would have VSE/POWER started as part of automated system initialization (ASI). You can do this if your system library includes a JCL ASI procedure that makes use of VSE/POWER's autostart function.

This section describes how to prepare the VSE/POWER related statements of an ASI procedure for VSE/POWER start-up; it shows also how to start VSE/POWER interactively from the console without making use of the autostart function.

Regardless of the method you use, you should consider the following:

- Formatting VSE/POWER queues

  Formatting or reformatting of the queues is required if, for example:

  - The extents for the VSE/POWER files are changed between executions of VSE/POWER.
  - A new version or release of VSE/POWER is installed and this new version or release requires the queue file to be reformatted.

  Formatting is not required when the information existing in the VSE/POWER files can be used as is. A start-up under these conditions is called a "warm start." If you perform a warm start after

an abnormal end of VSE/POWER, the next job number will be one higher than on termination. This prevents duplicate job numbers for accounting purposes.

- Defining the devices to be spooled in each partition

  VSE/POWER intercepts I/O requests for specified devices, regardless of the logical units assigned to these devices. If the I/O requests for a device are intercepted, the assignment for this device is in fact a "dummy" assignment; the device is not used by the application program. Therefore, logical units in different partitions under VSE/POWER control may be assigned to the same unit-record device.

  Dummy device assignments are required for multifunction card devices.

  A writer-only use of VSE/POWER and the use of VSE/POWER on a cardless system normally requires dummy device assignments.

- Device assignments for the VSE/POWER partition

  The partition must have a sufficient number of logical unit names available. Include a job control NPGR command in your job control start-up procedure, if necessary; the IBM set default - 30 programmer logical units - may not be enough.

  Reserve the logical units SYS000 through SYS016 for the VSE/POWER files by including assignments for them in your VSE/POWER start-up procedure.

A VSE/POWER start-up procedure can include a variety of commands:

- VSE commands - They are described in the publication VSE/Advanced Functions System Control Statements.

- VSE/POWER central operator commands - They are described in Chapter 4.

- Autostart statements - For information about these statements see "VSE/POWER Autostart Statements" on page 291.

The procedures described in this section assume that the disk volumes needed to accommodate the VSE/POWER files have been mounted and that their disk drives have been made ready.

Shared Spooling Considerations

Sharing systems can be started in any sequence.

The first sharing system to be started must format the VSE/POWER spool files, if formatting is necessary.

The operators of all sharing systems should synchronize their system clocks to the nearest minute for control purposes.

Starting of a sharing system may be delayed because:

- Another sharing system is being started.
- Another sharing system is currently performing queue-file recovery.
- Another sharing system is currently processing a POFFLOAD BACKUP function.
- Another sharing system is displaying the VIO storage copy of the queue file.
- The operator of a previously started sharing system has issued the PRESET command.
- Another previously started sharing system is processing the PACCOUNT command.
- Another sharing system came to an abnormal end while having exclusive control of a queue file. In this case, the VSE system operator must issue a VSE UNLOCK command to free all resources that were owned by the failing system, or this system must be restarted.

Part of the start-up process of a sharing system is a check to see if any of the sharing systems previously ended abnormally. If this is the case, the first sharing system to be started performs recovery of the items in the spool files for itself and the sharing system that ended abnormally. Recovery of the spool files is explained in more detail below.

Start-Up With Spool File Formatting (Cold Start)

If VSE/POWER is started in one sharing system with a request for formatting the shared spool files, VSE/POWER must not be active in any of the other sharing systems. The VSE operator who starts VSE/POWER is prompted by message 1QB2D to specify whether any sharing system is active. If the response is YES, the start-up is canceled; if the response is NO, the start-up continues and the specified spool files are formatted. An incorrect operator response of NO, when another sharing system is in fact active, produces unpredictable results in the sharing system.

The account file alone can be formatted even when other sharing systems are active. However, this results in deletion of the current account file, similar to processing of a PACCOUNT command with the DEL operand.

Start-Up Without Spool File Formatting (Warm Start)

Whenever a sharing system is being started and the operator response to the FORMAT= message is NO (or pressing ENTER), VSE/POWER attempts a warm start. This means that all jobs in the VSE/POWER shared spool files are ready for processing.

Part of the start-up process is a check to see whether any of the sharing systems ended abnormally. If no previous abnormal termination occurred, the warm-start continues to completion.

If a previous abnormal end did occur, the warm-start procedure is interrupted while the VSE/POWER spool file recovery routines attempt recovery of the shared spool files.

The following example explains what is meant by spool file recovery. In this example there are three sharing systems: System A, System B, and System C. System B ended abnormally; Systems A and C ended normally. All three systems are to be restarted with warm start in the sequence System A, System B, System C.

1. During the start-up of System A, the abnormal termination of System B is detected.

2. System A resets the status of all jobs that belong to System B in the shared spool files to 'dispatchable'. Any jobs that were in process on System B will start over again after System B is started. Any jobs being printed or read in on System B will be printed or read in from their beginning after System B is started.

3. Start-up of System A continues to its completion.

4. System B's warm start proceeds normally, as if its previous abnormal termination had not occurred.

5. System C's warm start proceeds normally.

If any sharing system is still active during a warm start of another sharing system, recovery is necessary only if (1) the sharing system that abnormally ended is being restarted and (2) recovery has not been performed for the restarting sharing system by the active sharing system. This is determined automatically by VSE/POWER and recovery is made if necessary.

The active sharing system checks for an abnormal end of another sharing system while that sharing system had write access to the control information in the spool files. When an abnormal end is detected, the active sharing system automatically performs recovery of the spool files. In

this case no recovery of the spool files is necessary when the abnormally ended sharing system is restarted.

Note: A sharing system may perform a warm start for spool files that have been used previously by a non-shared VSE/POWER system of the same version/release level (a previous normal or abnormal termination will also be taken into account). Therefore, the operator must ensure that the non-shared VSE/POWER system is not active while a sharing system is warm-started - otherwise, unpredictable results may occur.

Starting VSE/POWER with Autostart

This section describes how to start VSE/POWER during start-up of the VSE system. The method discussed here makes use of VSE/POWER's autostart function together with the VSE system's ASI (automated system initialization) function. The method requires that a suitable ASI JCL procedure is cataloged in the system sublibrary.

The publication VSE/Advanced Functions, System Management Guide describes how to write and catalog an ASI JCL procedure set. The statements needed for a VSE/POWER start-up procedure are discussed below.

1. Allocation of virtual and processor storage for VSE/POWER

   Assume that VSE/POWER is to run in partition F2. Your allocation for the partition (in ALLOC statements) might look like this:

   ```
   ALLOC R,...,F2=100K,...
   ALLOC S,...,F2=704K,...
   ```

   The ALLOC statements are included in the ASI JCL procedure for BG.

2. Starting the VSE/POWER partition

   To start VSE/POWER in partition F2, include in the ASI JCL procedure for BG the following VSE commands:

   ```
   START F2
   STOP
   ```

   This causes partition F2 to become active and partition BG to be stopped; it causes the ASI JCL procedure for partition F2 - the actual VSE/POWER start-up statements - to be called.

3. Make a list-output device available (to obtain a status report, if this is desired)

   This is done by assigning SYSLST to a printer, for example:

   ```
   ASSGN SYSLST,00E
   ```

If the start-up is a warm start, the assignment causes VSE/POWER to provide a status report immediately after completion of the start-up procedure. This report is similar to the one produced after a PEND statement. Figure 12 on page 121 gives an example of a PEND status report.

If you do not want a status report, specify ASSGN SYSLST,UA.

4. Define VSE/POWER files and invoke VSE/POWER

   Figure 10 on page 89 shows an example of the VSE job control statements required for this purpose. The numbers within parentheses in front of the statements refer to explanations given below the example.

5. Supply the VSE/POWER autostart statements

   An example of a set of autostart statements is shown and discussed in Figure 11 on page 90. The numbers within parentheses in front of the statements refer to explanations given below the example.

```
      (1) // ASSGN SYS000,260
      (2) // DLBL IJAFILE,'VSE.POWER.ACCOUNT.FILE',99/365,DA
      (2) // EXTENT SYS000,POWER1,1,0,12,60
      (1) // ASSGN SYS001,260
      (2) // DLBL IJQFILE,'VSE.POWER.QUEUE.FILE',99/365,DA
      (2) // EXTENT SYS001,POWER1,1,0,72,48
      (1) // ASSGN SYS002,133
      (1) // ASSGN SYS003,134
      (2) // DLBL IJDFILE,'VSE.POWER.DATA.FILE',99/365,DA
  (2) (3) // EXTENT SYS002,POWER2,1,0,95,855
  (2) (3) // EXTENT SYS003,POWER3,1,1,3800,1045
      (4) // EXEC POWER,SIZE=552K
```

(1) SYS000, SYS001, and SYS002 may be assigned to disk devices of different types.
However, SYS002 and SYS003 (the data file extents) must be assigned to devices of the same type.

(2) DLBL and EXTENT statements may have been stored permanently in your system's label information area. In this case, they need not be included in your ASI procedure that starts the partition for VSE/POWER.

The VSE/POWER account file and the queue file occupy one extent each. If the account file is to be saved on a tape with standard labels, you must provide an applicable TLBL statement.

(3) Up to 15 extents can be specified if the entire data file does not fit into one extent. Performance is slightly improved if you use additional extents.

(4) The example assumes that VSE/POWER runs in a 704K partition. The specification SIZE=552K causes 152K of the partition to be reserved as GETVIS space.

Figure 10. Job Stream Example for Starting VSE/POWER

```
(1)  SET SYSID=2
(2)  SET PNET=NETTBLE3
(3)  DEFINE L,PAGEDEF,1F,1,8,C
(3)  DEFINE L,FORMDEF,1D,1,8,C
(4)  FORMAT=NO
     PSTART BG,GST
     READER=00C
     PRINTERS=00E
     PUNCHES=00D
     PSTART F3,CEGF
     READER=00C          See part 2
     PRINTERS=01E
     PUNCHES=00D
     PSTART LST,00E
     PSTART LST,01E
     PSTART PUN,00D
     PSTART RJE,060
     PSTART RDR,00C
     /+
```

(1) SET SYSID=2
The statement implies that the shared-spooling function is used. It defines that VSE/POWER is to run on the processor designated as number 2 (see also "Note" below).

(2) SET PNET=NETTBLE3
The statement defines the name of the network-definition table to be used by VSE/POWER. The definition in this statement overrides the name that was specified in the PNET operand of the POWER generation macro (see also "Note" below).
Note: If a SET statement is in error, you have to correct your procedure; the error cannot be corrected at the console.

(3) The statements define the PAGEDEF and FORMDEF keywords as valid operands of the * $$ LST statement.

(4) FORMAT=NO
Specify FORMAT=NO if no VSE/POWER file is to be formatted, that is, if you perform a VSE/POWER "warm start." Else provide, instead of NO, formatting instructions as needed. For example: FORMAT=D causes the queue and data files to be formatted, but not the account file.
If you do not supply a FORMAT statement in your procedure, VSE/POWER prompts the operator for formatting instructions.

Figure 11 (Part 1 of 2). Example of a VSE/POWER Autostart Procedure

```
        SET SYSID=2                    |
        SET PNET=NETTBLE3              |
        DEFINE L,PAGEDEF,1F,1,8,C      |  See part 1
        DEFINE L,FORMDEF,1D,1,8,C      |
        FORMAT=NO                      |
  (5)   PSTART BG,GST
  (5)   READER=00C
  (5)   PRINTERS=00E
  (5)   PUNCHES=00D
  (5)   PSTART F3,CEGF
  (5)   READER=00C
  (5)   PRINTERS=01E
  (5)   PUNCHES=00D
  (6)   PSTART LST,00E
  (6)   PSTART LST,01E
  (6)   PSTART PUN,00D
  (6)   PSTART RJE,060
  (6)   PSTART RDR,00C
        /+
```

(5) The statements place the partitions BG and F3 under control of VSE/POWER. PSTART BG,GST defines, for example, that jobs with the input classes G, S, and T are to be processed in partition BG. PSTART commands that place a partition under VSE/POWER control can be submitted in any sequence. However, the spool-device specifications for a partition must follow immediately the PSTART command for the partition and in the sequence as shown. If the spool device specifications are incorrect or incomplete, then VSE/POWER prompts the operator for acceptable specifications. This is the same as for a start-up without autostart.

(6) PSTART LST,00E starts a list task for the printer at 00E.
Statements to start VSE/POWER tasks can be included in any sequence behind the required SET and FORMAT statements. However, to avoid conflicting device assignments, place PSTART statements for tasks behind the PSTART statements for partitions.
Your procedure may include PSTART PNET,nodeid,... commands for both BSC and SNA nodes. It may include the required PACT commands; however, a PACT command may not precede its corresponding PNET command if this PACT is given for an SNA node.
The first PSTART PNET command for an SNA node causes VSE/POWER to attempt linking to ACF/VTAM; If ACF/VTAM is not active, VSE/POWER queues the PSTART commands for SNA nodes until ACF/VTAM becomes active.

Figure 11 (Part 2 of 2). Example of a VSE/POWER Autostart Procedure

Starting VSE/POWER without Autostart

In rare case you, the operator at the system's console, may have to start up VSE/POWER in the middle of a shift. The sample procedure given here assumes that VSE/POWER is to reside in partition F1 and controls partitions BG, F2, and F3. The example further assumes that the following devices are available for spooling input and output:

Card reader with device address 00C

Card punch with device address 00D

Printer with device address 00E

Printer with device address 01E

Printer with device address 118

Disk storage device with device address 260

Disk storage device with device address 133

Disk storage device with device address 134

Terminal device connected to a communication controller with device address 060

The disk storage devices are used for the VSE/POWER files; the data-transmission terminal for remote job entry.

Step 1: Make Devices Available

Most likely, the spool devices to be used by VSE/POWER have been used by other partitions. Before you can make them available for VSE/POWER use, you have to unassign them in the partition that owns them. To do this, enter for example:

```
ASSGN SYSIN,UA
ASSGN SYSLST,UA
ASSGN SYSPCH,UA
ASSGN SYS002,UA (if SYS002 was in use)
```

Alternatively you may enter, for each of the affected devices, the sequence

```
DVCDN cuu
DVCUP cuu
```

Step 2: Set Up Partition F1

In rare cases, you may have to allocate storage to partition F1. If so, stop one or more of the other partitions in your system by issuing the VSE command UNBATCH for them. After successful storage allocation, you may start those VSE partitions again. A typical allocation example would be:

```
ALLOC R,F1=64K,...
ALLOC S,F1=768K,...
```

Note: If VSE/POWER has been in operation, you must end VSE/POWER operation by a normal shut down before you do any reallocation of partitions. How to do this is described under "Normal Shut-Down" on page 120.

Step 3: Start the VSE/POWER Partition

Start the VSE/POWER partition (F1 in the example) by issuing the command:

```
START F1
```

Step 4: Assign SYSRDR

Omit this step if you intend to enter the required control statements via SYSLOG (see also step 6); otherwise assign SYSRDR to the device from which the statements for starting VSE/POWER are to be read.

Note: SYSRDR is not unassigned automatically if VSE/POWER terminates and the SYSRDR file resides on a disk or a diskette device.

Step 5: Prepare for a Status Report

Assign SYSLST if your installation requires a status report after a warm start. Figure 12 on page 121 shows an example of such a status report.

Step 6: Define the VSE/POWER Files and Invoke VSE/POWER

Place the required VSE control statements into the card reader to which you assigned SYSRDR in step 4. An example of these statements, which vary from one location to another, is given in Figure 10 on page 89.

If you did not assign SYSRDR, you have to enter the statements on the console, one after the other.

Step 7: Format VSE/POWER Queues

When VSE/POWER displays the message

```
1Q11D FORMAT QUEUES=
```

respond as instructed by your system administrator (see "FORMAT: Specify File-Formatting Options" on page 299). Your response (behind the equal sign) can be one of the following:

{D|Q} Format the queue file and the data file.

A Format the account file.

D,A Format all VSE/POWER files.

NO No formatting is required (instead of entering NO, you may simply press END/ENTER). For VSE/POWER, this response is a "warm start" indication. If you assigned SYSLST (in step 5), you get a status report.

When VSE/POWER start-up is complete, you get the message

```
1Q12I VSE/POWER 2.3 INITIATION COMPLETED
```

If, at that point, SYSIN is assigned to a disk device, this assignment is retained; otherwise SYSIN is unassigned by VSE/POWER together with SYSLST and SYSPCH.

You can now start processing under control of VSE/POWER as outlined in the next step.

Step 8: Start VSE/POWER Tasks

Use the PSTART command to place partitions under control of VSE/POWER as required and to start VSE/POWER tasks as are needed.

- Placing a partition under the control of VSE/POWER

  For example, the command:

  ```
  PSTART F2,2A
  ```

  places the partition F2 under control of VSE/POWER. This partition must have lower priority than the VSE/POWER partition.

  In response to the command, VSE/POWER prompts you to specify the devices that are to be spooled for the partition. The prompting messages are (for the above example) as follows:

  ```
  1R86I  PLEASE SPECIFY DEVICES TO BE SPOOLED
  1R50D  F2 READER=
  ```

```
1R50D  F2 PRINTERS=
1R50D  F2 PUNCHES=
```

SYSLOG stops after each prompting message (1R50D). This allows you to specify a device address (in the form cuu) or to enter NO, if one of the device types is not used, as shown in the example below:

```
1R50D  F2 READER=C
1R50D  F2 PRINTERS=E,1E
1R50D  F2 PUNCHES=NO
```

Per partition, you can specify:

One device to be used as a reader.
Up to 14 devices to be used as printers.
Up to 14 devices to be used as punches.

If your response is READER=NO, but you specify one or more addresses for printers or for punches (or both), VSE/POWER operates as a writer-only spooling program for the partition that is being started.

VSE/POWER spools only the input if your response is READER=cuu, PRINTERS=NO and PUNCHES=NO.

A device can be specified only once, even if it can both read and write.

After a partition has been placed under control of VSE/POWER, I/O requests from that partition to the specified devices are intercepted by VSE/POWER. If the reader queue contains no entry for processing in this partition, VSE/POWER displays the message:

```
1Q34I F2 WAITING FOR WORK
```

Use similar PSTART commands to place partition BG or one of the other foreground partitions under control of VSE/POWER.

- Starting a VSE/POWER task

For example, the command:

```
PSTART RDR,00C,A
```

causes a VSE/POWER reader task to spool input from the card reader at 00C. The task assigns class A to all reader queue entries from that reader, except when another class is specified in the JECL for a job input.

Use similar PSTART commands to start other VSE/POWER tasks as required.

CONTROLLING OPERATION

Using VSE Controls: Certain commands and controls can be used in a VSE/POWER controlled environment only with restrictions:

- The LFCB command

  If issued for a printer before or during the execution of a VSE/POWER LST task, this command will have no effect on the FCB of the printer that is used for output from the LST task.

- The STOP command

  If STOP is used for a partition controlled by VSE/POWER, a subsequent PSTOP or PEND command will not be successful until the partition has been restarted.

- The PRTY command

  The command can assign to a VSE/POWER controlled partition only a processing priority lower than that of the VSE/POWER partition.

- The SYSBUFLD program

  Do not use the SYSBUFLD program for loading an FCB or UCB buffer image or for a band-id verification in a partition that is supported by VSE/POWER. Use an * $$ LST statement with the FCB= or UCS= operand instead.

Controlling Jobs: The processing of jobs under VSE/POWER and also the retrieval of output from these jobs can be controlled by VSE/POWER commands. These commands may be entered in one of three ways:

- At the system console by the central operator

  Use the VSE/POWER central operator commands described in Chapter 4. Operations at the system console are discussed in this section.

- At a remote work station by the operator at this station

  A set of commands similar to the VSE/POWER central operator commands is available. For information on the commands and how to operate at BSC and SNA work stations, refer to the publication VSE/POWER Remote Job Entry User's Guide.

- From within an application program

  You can invoke certain VSE/POWER services from an application program running in a partition other than the one of VSE/POWER. To do this, issue a CTL request using the support as described under the spool-access support in VSE/POWER Application Programming.

The remainder of this section discusses operations at the system console; it includes sections on the operation with miscellaneous devices and on the operation with tapes.

Controlling Operation at the System Console

In using the VSE/POWER central operator commands at the system console, you may request a variety of functions. Examples are:

- Change the processing priority for one or a specific group of VSE/POWER jobs, which may be spooled job input or job output.
- Delete jobs in the input queue or output in the output queues.
- Hold certain jobs or output in the queues for later processing.
- Interrupt printer or punch output (for example, to free a device urgently required by some other task).
- Restart printer or punch output, either from the beginning or from a certain point within the data stream (to restart printer output on a page other than the first, each page must start with a skip to channel 1).
- Specify the number of copies to be printed or punched for a specific job output.

You request VSE/POWER to perform the desired function by entering the suitable VSE/POWER central operator command. To enter a command, proceed as if you were entering a VSE attention-routine command.

Whenever you enter a command related to a queue entry (a job or the output of a job), try to have the command correctly identify the job by:

- The name under which it was logged by VSE/POWER and, possibly, also
- The number that VSE/POWER assigned to it (a number is required, for example, if two or more VSE/POWER jobs in a queue were logged under identical names).

If you are in doubt about the correct identification of a VSE/POWER job, use the PDISPLAY command to have VSE/POWER display this identification. For more details about this command, see the section "PDISPLAY: Display VSE/POWER Status" on page 151.

This section discusses various operator tasks such as altering partition-related VSE/POWER controls, starting and stopping VSE/POWER tasks, and Controlling and monitoring remote job entry (RJE) at the system console.

Altering Partition-Related VSE/POWER Controls

Changing Partition Priorities: To place a partition under control of VSE/POWER, this partition's processing priority has to be lower than that of the VSE/POWER partition. To change the partition's priority, use the VSE command PRTY; this command is described in VSE/Advanced Functions, System Control Statements.

Generally, there is no need for changing the priority of a partition once it is under control of VSE/POWER. Consider altering instead the class and priority assignments of a queue entry or a group of queue entries.

VSE/POWER rejects a PSTART command for a partition whose priority is not lower than the priority of the VSE/POWER partition.

How to place a partition under the control of VSE/POWER is described under "Step 8: Start VSE/POWER Tasks" on page 94.

Changing the Class Specification for a Partition: When placing a partition under VSE/POWER control, you can specify one or more classes to ensure that only jobs with one of these classes are executed in this partition. Use the PALTER command to change this class specification. For example:

```
PALTER  BG,ABC
```

tells VSE/POWER that only jobs with a class assignment of A, B, or C are now eligible for selection in partition BG. The command takes effect on completion of the job that is being processed in the partition.

Stopping and Restarting a Partition: You may have I/O requirements which are not supported by VSE/POWER (for example, input from a magnetic-ink character reader), or you may have changed real storage allocation for the partition. This requires you to:

1. Stop a partition under VSE/POWER.
2. Start the partition under control of the VSE supervisor.
3. Run the job in question.
4. Bring the partition back under VSE/POWER control after this job is finished.

Proceed as follows:

1. Issue a PSTOP command for the partition. For example, the command:

   ```
   PSTOP F2
   ```

   causes the job being processed in partition F2 to be completed and the partition to be returned to VSE control. SYSRDR and all assignments to spool devices for this partition are also released. VSE/POWER acknowledges the command by a message.

Do not use the VSE STOP command because this does not release the partition from VSE/POWER control.

2. Stop any VSE/POWER reader or writer task that uses devices which may be required by the job to be run. Examples for stopping a task are given under "Stopping a VSE/POWER Task" on page 101.

3. Assign the necessary devices for the partition.

4. Make the devices ready and start the partition using the ENTER key.

5. When the job has finished, stop the partition using the VSE STOP command.

6. Issue the PSTART command to place the partition under control of VSE/POWER again. How to do this is described as part of the start-up procedure under "Step 8: Start VSE/POWER Tasks" on page 94.

Loading Print Buffers

Loading a Printer's FCB via VSE/POWER JECL: By specifying FCB=phasename in your * $$ LST statement, you cause VSE/POWER to get the named FCB-image phase from the defined sublibrary. VSE/POWER uses the control values in that phase for spooling print output.

When the spooled output is to be printed, VSE/POWER causes the printer's FCB to be loaded with the FCB image phase used for spooling the output.

VSE/POWER loads the default FCB at the beginning of output for which no FCB image was specified.

Loading the FCB within a VSE/POWER Job: The job step to load an FCB under VSE/POWER may be placed anywhere within a VSE/POWER job.

When VSE/POWER prints the output of a job and finds the load FCB request, the current forms are aligned to line 1 and the buffer is loaded. Note that line 1 and the channel-1 line may be different.

When the printer's FCB has been loaded, VSE/POWER displays message 1Q40A. In response to the message, proceed as follows:

1. Place the forms indicated in the message on the printer and align them using the PSETUP command. This command allows you to adjust forms alignment during printing.

2. Enter the command PGO cuu (where cuu = the printer's channel and unit number).

Loading a UCB under VSE/POWER: The procedure for loading the UCB under VSE/POWER is the same as that for loading the FCB. The system requests you by message to mount the correct print train (chain or belt) for the UCB-image phase just loaded.

An image loaded into a printer's UCB remains in the buffer until a user program makes another load-UCB request. No defaults are assumed.

Loading an FCB and a UCB under VSE/POWER: If there is a need for loading a printer's UCB and FCB together, you can do this in one job. Example:

```
* $$ JOB JNM=UCB/FCB
* $$ LST DISP=D,CLASS=A,FNO=A123,UCS=IJBTRT11,FCB=SPEC3
// JOB   DUMMY
/&
* $$ EOJ
```

Controlling VSE/POWER Tasks

Starting a Reader Task: A reader task reads card-image records from the specified unit-record device or from a tape. It writes these records onto intermediate storage on disk.

You start a reader task by issuing a PSTART command. You can do this whenever you have a suitable input device available, regardless of the number of reader tasks currently active.

The command:

```
PSTART RDR,00C,2
```

starts a reader task with a partition-dependent class specification. The task reads job input from the device at the address 00C. Unless a different class is specified in the input, the VSE/POWER jobs spooled by this task can be executed only in partition F2.

The command:

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

starts a reader task with a partition-independent class. A VSE/POWER job spooled by this task is assigned class C, unless a different class is specified in the job's input.

The command:

```
PSTART RDR,00B,,'INPUT',2
```

starts a reader task to read a file named INPUT from two diskette volumes on the device at the address 00B.

The command:

```
PSTART RDR,480
```

starts a reader task to read from a tape on the drive at the address 480. VSE/POWER assigns (default) class A to the input from this tape.

Starting a VSE/POWER List or Punch Task: The task writes data from intermediate storage to the printer or punch, respectively.

You start a list- or punch task by issuing a PSTART command. You can do this whenever you have a printer or punch available, regardless of the number of list- or punch tasks currently active.

The command:

```
PSTART LST,00E,AB
```

starts a list task to print spooled output that has classes A and B assigned. The task uses the printer at the address 00E.

The command:

```
PSTART LST,118,T,2
```

starts a list task with two print buffers. The task prints spooled output that has class T assigned; it uses the printer at the address 118.

The command:

```
PSTART PUN,00D,X'285'
```

starts a punch task for writing output to the device at the address 00D. This output was spooled onto a tape mounted on the drive at the address 285.

The command:

```
S LST,00E,W,,VM
```

starts a list task for printing spooled list output to the virtual printer with the address 00E.

Stopping a VSE/POWER Task: The PSTOP command is available to stop a VSE/POWER task or to stop the transmission of data via an RJE line.

Specify EOJ as the second operand if you want to stop a reader task. This ensures that the currently processed VSE/POWER job is properly placed into the reader queue. If you do not specify EOJ, the task stops immediately, and the entire input for the currently processed VSE/POWER job has to be resubmitted.

The command:

```
PSTOP 00C,EOJ
```

causes the associated reader task to stop after having finished reading the currently processed VSE/POWER job into the reader queue.

The command:

```
PSTOP 280,EOJ
```

causes the associated tape reader task (reading from SYSIN tape or queues written to tape by POFFLOAD) to be stopped on end of job. The tape is mounted on the tape drive at the address 280.

To stop a list task, you can use the PSTOP command without a second operand, or you can use it with EOJ or RESTART specified. The command:

```
PSTOP 00E
```

causes the associated list task to stop output processing immediately. The currently processed queue entry is retained in the output queue. When the task is started again (by a PSTART command), VSE/POWER processes the output for that queue entry beginning with the first output line in the currently processed entry.

The command:

```
PSTOP 00E,RESTART
```

causes the associated list task to complete the page that is being printed and then to stop. When the task is started again (by PSTART), VSE/POWER processes the output for the queue entry as follows:

- If the printer is not an IBM 3800

  Beginning with the page after the last one processed before the PSTOP command was issued.

- If the printer is an IBM 3800

  Beginning with the next page within the copy group that was processed when the PSTOP was given. For more information on copy grouping refer to the publication DOS/VS IBM 3800 Printing Subsystem, Programmer's Guide.

The command:

```
PSTOP 00E,EOJ
```

causes the associated list task to stop after having finished printing the currently processed output queue entry. When the task is started again (by PSTART), it processes the output of the next entry in the list queue.

The information provided above for stopping a list task applies accordingly to stopping a punch task. Examples of PSTOP commands to stop a punch task:

```
PSTOP 00D
PSTOP 00D,RESTART
PSTOP 00D,EOJ
```

Account-File-Full Condition: If the account file is full and a task of VSE/POWER must write another account record, this task waits until you issue a PACCOUNT command. Your programmer will instruct you to do one of the following:

- Save the account records on tape (normally the preferred action).
- Save the account records on disk if a disk extent has been defined for this purpose. This disk file must have been defined as a sequential output file consisting of one extent.
- Have the contents of the account file spooled to the punch queue and punched out by starting a punch-writer task with class P.

Output Segmentation

VSE/POWER job output can be segmented, that is, part of the output from a job can be printed or punched before the entire job is finished.

Four types of segmentation are possible based on the event that initiates the segmentation:

- Count-driven output segmentation.

  You can specify the number of print pages or of punched cards in the output file after which physical output should be started. You define this either in the RBS operand of either the POWER macro during table generation or in the * $$ LST or * $$ PUN statement for a given job.

  VSE/POWER prints a suffix on the separator pages as follows:

  ```
  0001 for the first segment of an output.
  0002 for the second segment of an output.
  And so on
  LAST for the last segment of an output.
  ```

  This helps to identify the sequence of the output.

  All parts of a segmented output have the same job name and job number as the VSE/POWER job. In addition, each segment is given a unique segment number in ascending sequence, displayed as S=nnn after a PDISPLAY FULL=YES command.

The maximum number of segments for printed or punch output is 127.

- Data-driven output segmentation.

  You may submit two or more * $$ LST or * $$ PUN statements within the boundaries of one VSE/POWER job. Each of these statements causes an individual list or punch queue entry to be created.

  Data-driven segmentation in a writer-only partition requires certain precautions. VSE logs all JCL and VSE comments on SYSLST. Since VSE/POWER JECL statements are treated as VSE comments, they are also logged on SYSLST. This causes the * $$ JOB statement to be placed into the queue as output. The * $$ LST statement segments this output causing multiple queue entries to be created for each job in a writer-only partition. The following example shows how to prevent logging of extraneous * $$ JOB statements:

```
// NOLOG        To set temporary override
* $$ JOB
   ... ... ...
* $$ EOJ
// JOB name     To reset temporary override
```

  The // SETPRT VSE job control statement can also cause list output segmentation.

  Each list or punch queue entry will have its unique job number.

- Multivolume tape segmentation.

  When end-of-volume occurs during tape spooling of list output (DISP=T), VSE/POWER segments the output at a logical boundary (last skip to channel 1 or full page) and requests another tape to be mounted. To print the output on these tapes, ensure that they are mounted in correct sequence.

- Program-driven output segmentation.

  In an application program the VSE/POWER SEGMENT macro may be used to separate the output. For details refer to VSE/POWER Application Programming.

Creation and Handling of Disposition-X Queue Entries

A queue entry is assigned a disposition of X if one of the following conditions is true:

- The system or VSE/POWER fails while the entry is being processed, and SET NORUN=YES was specified in the autostart procedure.

- A checkpointed output is being submitted via the spool-access support; during this processing, the system, VSE/POWER, or the accessing application program fails.

- The system or VSE/POWER fails, but the partial, checkpointed output on the LST queue is not discarded (if the RBC operand is specified in the * $$ LST or PUN statement); instead, the output is added to the appropriate class chain with disposition X.

Assignment of a disposition of X makes the queue entry unavailable for normal processing. For further information on dispositions, see also Appendix A on page 309.

Queue entries that have a disposition of X should be handled as follows:

1. Get a list of the affected queue entries by submitting the command

   ```
   PDISPLAY ALL,CDISP=X
   ```

2. For a queue entry under your control (not submitted via spool-access support), change the disposition from X to the original one. Example:

   ```
   PALTER RDR,JOBXYZ,DISP=*
   ```

   This makes the entry eligible for processing under control of VSE/POWER.

   For a queue entry not under your control (submitted via spool-access support), do not take any action other than reporting its name to the owner or administrator of the submitting program or subsystem.

Creation and Handling of Disposition-Y Queue Entries

A spool-access support user or a device-driving subsystem can issue a 'quit-and-lock' request during the retrieval of a queue entry. This request causes the queue entry currently accessed to be requeued in the appropriate class chain with a temporary disposition of Y to:

- Indicate that some problem occurred during output processing, and

- Prevent that the output entry is handled again until the subsystem has taken some special action.

When cross-partition communication is abnormally terminated while a spool-access support user is retrieving a predefined 'protected' output entry, that queue entry is also assigned disposition Y. (A protected output entry is one that has been created via the spool-access PUT output service with SPL field SPLDMOHP set on to signal: 'Hold when print/punch fails'. For details, see VSE/POWER Application Programming.)

Assignment of a disposition of Y makes the queue entry unavailable for normal processing. For further information on dispositions, see also Appendix A on page 309.

Queue entries that have a disposition of Y should be handled as follows:

1. Get a list of the affected queue entries by submitting the command

   ```
   PDISPLAY ALL,CDISP=Y
   ```

2. For a queue entry under your control (not submitted via spool-access support), change the disposition from Y back to the original one. Example:

   ```
   PALTER LST,JOBXYZ,DISP=*
   ```

   This makes the entry eligible for processing under control of VSE/POWER.

   For a queue entry not under your control (submitted via spool-access support), do not take any action other than reporting its name to the owner or administrator of the accessing program or subsystem.

Remote Job Entry (Work Station) Support

Starting an RJE,BSC Line: You do this by starting a VSE/POWER RJE task. This task processes job streams from a work station and builds a reader queue entry for each VSE/POWER job transmitted from that station. On request from the work station operator, an RJE print or punch task makes output available for transmission to the work station. VSE/POWER allows you to start up to 100 RJE,BSC lines for concurrent operation.

The command:

```
PSTART RJE,060,RJEOK09
```

for example, causes the line with the device address 060 to be started. When VSE/POWER has finished processing this command, a work station operator can sign on at a work station linked via this line. The operator's SIGNON command is accepted only if it specifies the password RJEOK09.

Note: A password specified in the PSTART command for an RJE,BSC line overrides the password defined in the PLINE macro for this line.

Starting an RJE,SNA Session: VSE/POWER's RJE,SNA support uses ACF/VTAM, an IBM program product needed for RJE,SNA operation. For an SNA work-station user to log on to VSE/POWER, you must set up VSE/POWER's interface to ACF/VTAM. You do this by a PSTART command in the format:

```
PSTART RJE,SNA
```

When VSE/POWER has finished processing the command, SNA work stations defined to ACF/VTAM (as logical units) and to VSE/POWER can log on. VSE/POWER indicates successful completion by message 1V04I.

Commands for Controlling RJE Operation: The following central operator commands can be used to control RJE operation from the system console at the central location:

PALTER — To change one or more job attributes of one or a group of RJE jobs. You may change, for example, the class or disposition of a job or the destination or copy count of an output.

PBRDCST — To send a message to any of the following:

- An individual work station
- All work stations
- Another node
- Work stations linked to another node

PDELETE — To delete one or more entries from the specified job queue; alternatively to delete a certain ALLUSERS-type message or delete all of such messages.

PDISPLAY — To display the status of RJE-type queue entries or list all ALLUSERS-type messages.

PHOLD — To place into the hold state all input originated with the specified remote-identifier, or all output directed to this remote-identifier.

PINQUIRE — For RJE,BSC lines – To display the status of one or all supported RJE lines.

For RJE,SNA sessions – To display the status (PROCESSING, LOGGED ON, or LOGGING ON) of one logical unit or to display the logical unit names of all active logical units together with the remote-IDs of the users operating them.

PSTOP — To end a specific user session or to deactivate the ACF/VTAM interface, which causes all user sessions currently active to be terminated. The command is further discussed below.

Stopping an RJE,BSC Line: The following procedure is recommended for stopping an RJE line.

1. Determine the current status of the system's RJE lines; you do this by issuing the command

   ```
   PINQUIRE ALL
   ```

2. Issue a PSTOP command for a line which is inactive. Specify EOJ as the second operand in your PSTOP command; remote job entry may have

been started between VSE/POWER's response to your PINQUIRE command and the processing of your PSTOP command. For example, the command

```
PSTOP 060,EOJ
```

allows the transmission of input or output to continue until the end of the queue entry that is being processed via this line.

If a line must be stopped at once, use the command

```
PSTOP 060
```

This command, referred to as an immediate stop, causes transmission of data on line 060 to be stopped at once. VSE/POWER attempts to stop the line in accordance with the existing line protocol and to send a line-stopped message to the connected work station.

VSE/POWER may be unable to stop the line by processing a command as shown above. This is indicated if the line-stopped message (1R02I) is not displayed within a minute or two. In that case, request a forced line stop by issuing the command

```
PSTOP 060,FORCE
```

Stopping RJE,SNA Sessions: To stop an individual user session, use a PSTOP command with a format as shown by the example below:

```
PSTOP RJE,SNA,PRLL03,EOJ
```

In the command, PRLL03 is the logical unit name that was used to define the affected work station to ACF/VTAM. EOJ in the command ensures that all input or output for the currently processed VSE/POWER job will be completed. If you omit EOJ - and you should do this only in case of an emergency - VSE/POWER stops all processing for the named session immediately. As a result, input data may get lost.

To stop all currently active RJE,SNA sessions, issue the command:

```
PSTOP RJE,SNA,EOJ
```

The command causes VSE/POWER to end all sessions. Since EOJ is specified, each of the active sessions is allowed to continue until the end of the currently processed VSE/POWER job is reached. If you issue the command without EOJ - and again, you should do this only in case of an emergency - VSE/POWER stops all processing for all active sessions immediately.

SPECIFIC DEVICES

This section discusses the use of the IBM 4248 printer, of multifunction unit-record devices, and of diskette I/O units under control of VSE/POWER.

VSE/POWER Support of the IBM 4248

A program that writes to an IBM 4248 printer operating in native mode can run under control of VSE/POWER. In general, there is no need for you to change the program. VSE/POWER handles IBM 4248-specific I/O requests as follows:

- Print band verification:

  Define the identifier of the required print band in an * $$ LST statement. You do this by way of the statement's UCS operand. Example: UCS=4199.

  Note: Do not use the SYSBUFLD program to define a print band needed for printing output spooled by VSE/POWER.

  When the spooled output is to be printed, VSE/POWER compares the specified identifier with that of the mounted band. If the identifiers match, processing continues. If they do not match, VSE/POWER

  1. Issues a 'band-needed' message to the operator.
  2. Causes the identifier of the required print band to be displayed on the printer. Example:

     BAND RQ 4199

  Then either delay the printing of the spooled output or mount the required print band at once.

- Forms-mount request:

  When a new form is to be mounted, VSE/POWER issues a forms-mount message. In addition, it causes the required form to be displayed on the printer. Example:

  FORM RQ A003

- Horizontal-copy printing: If horizontal copying is not set in the loaded FCB, VSE/POWER ignores any horizontal-copy related print order that may have been spooled.

  VSE/POWER turns off horizontal copying for the printing of job- and copy-separator pages.

- User-written channel programs: Some of the IBM 4248-specific I/O commands cannot be processed such that they achieve the expected result. These commands are listed in VSE/POWER Application Programming.

Using Multifunction Unit-Record Devices

Multipurpose unit record devices (a card read/punch device with a card-print feature, for example) cannot be used by a reader and a punch task at the same time. If, for example, it is necessary to start a reader task for a multipurpose device while a punch task is still using this device, you must first stop the punch task using the PSTOP command.

Assuming that your multipurpose device's address is 00D, then the command

```
PSTOP 00D,EOJ
```

causes punched output on the device to be stopped at the end of the punch queue entry currently processed by the associated punch task, and the command:

```
PSTOP 00D,RESTART
```

causes punched output on the device to be stopped immediately.

In either case, VSE/POWER confirms your request by a message.

You may now start the reader task using the command:

```
PSTART RDR,00D
```

When the reader task has finished reading the new input, stop this task and restart the interrupted punch task. For the above example, the commands would be:

```
PSTOP 00D
```

and, following VSE/POWER's confirmation message:

```
PSTART PUN,00D
```

If you specify RESTART in the PSTOP command as shown above, VSE/POWER continues output processing at the point of interruption.

Using an IBM 3540 Diskette Unit

VSE/POWER can read data from a diskette in SYSIN mode or in data mode. These modes of data input are discussed below.

The Diskette Unit in SYSIN Mode

A VSE/POWER reader task can read diskette records of a length of up to 128 bytes. The actual size of the records to be read is taken from the diskette file's HDR1 label. If the input records are 81 characters long, only the last 80 bytes are copied to the VSE/POWER data file.

When a reader task reads an * $$ RDR statement, then VSE/POWER attempts to assign a free diskette unit to read the requested file. If no diskette unit is available, VSE/POWER issues a message and cancels the currently processed job input.

If you define a diskette input unit in your * $$ RDR statement, then SYSIN records can be read only by an application program that reads from a card reader specified at partition start-up as a unit-record spool device. However, if you specify the DSKTIN operand in the * $$ JOB statement, your program can use DTFDU[5] to access these SYSIN records. In this case, VSE/POWER provides for the SYSIN data to be read from the named diskette unit when the job is being executed.

SYSIN-Mode Example 1: Job control statements and data are both in one diskette-SYSIN file. The operator enters a PSTART command to start a reader task on a diskette unit. For example:

```
PSTART RDR,00B,B,'FILE-ID',8
```

Up to eight diskettes of the file called FILE-ID can be read from the diskette I/O unit at 00B. Reading stops after eight diskettes or after reading a diskette that does not have a continuation indicator in its label.

One diskette file may contain just one VSE or VSE/POWER job, it may contain several such jobs. Jobs with no class specification in their * $$ JOB card, or jobs for which no * $$ CTL statement is in effect, are put into class B.

SYSIN-Mode Example 2: A program may use SYSIN data and normal input data, both from diskettes.

The example assumes that SYSIN data is to be read from a diskette via DTFDU requests from the application program. This is indicated to VSE/POWER by a DSKTIN=cuu specification in the * $$ JOB statement. The

---

[5] DTFDU = define the file for diskette unit in an assembler program; a diskette I/O file declaration/definition for a compiler.

* $$ RDR statement to define the diskette I/O unit for normal input data may not refer to the same diskette unit you defined for SYSIN input in the * $$ JOB statement.

When the program is executed, VSE/POWER retrieves SYSIN data from the diskette, one record after the other, in response to the program's diskette read requests.

The command

```
PSTART RDR,00B,B,'FILE-ID1',8
```

starts a reader task to read from a diskette I/O unit. For the job, JECL and JCL statements might be stored on the diskette as follows:

```
* $$ JOB ...,DSKTIN=00B
      ... ... ...
// ASSGN SYSIPT,00B
      ... ... ...
* $$ RDR DEV=00D,FID='FILE-ID2'
      ... ... ...
```

The specification DSKTIN=00B in the * $$ JOB statement indicates that, during program execution, VSE/POWER is to:

1. Read records from the diskette unit at 00B.

2. Make these records available to the application program, one after the other.

The * $$ RDR statement tells VSE/POWER that data records are to be spooled from the diskette file FILE-ID2. The specified device address (00D in this example) must be defined to your system's supervisor.

The Diskette in Data Mode

For data files, a reader task can read diskette records of a length of up to 128 characters. These records are not examined for control statements and are written onto the spool file exactly as read. The records cannot be read by programs accessing a card reader; they can be read only by a user program that reads from the device defined in the * $$ RDR statement. This device must be a diskette I/O unit. DTFDU or DTFDI[6] can be used to access these files.

Two or more files per diskette volume may be specified by two or more * $$ RDR statements in the input stream.

---

[6] DTFDI = define the file device independent in an assembler program; a device-independent file declaration/definition for a compiler

The * $$ RDR statement causes VSE/POWER to insert data from a diskette file into the input being read from the card reader. You do not need to submit other JECL statements for a job containing a RDR statement. This statement is ignored in a writer-only partition.

Although a diskette file may be inserted into an input stream, it is not possible to insert data from another source, a card reader for example, into the middle of a diskette file. Likewise, it is not possible to insert a diskette data file into another.

Data-Mode Example 1: The job control statements are in the card reader, data is on a diskette. The operator enters a PSTART command for the card reader:

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

This command tells VSE/POWER to start a reader task for the reader at 00C and that the task is to read from a diskette at 00B. Both input devices belong to the reader task and cannot be accessed by any other partition or VSE/POWER task until the reader task has finished.

Assume that cards are in the card reader as follows:

```
// JOB EX1
// ASSGN SYS008,00B
// DLBL FILE,,,DU
// EXTENT SYS008
// EXTENT SYS008
// EXEC PROG
* $$ RDR DEV=00B,FID='FILE-ID',NOD=2
/*
/&
```

The specification SYS008 in the // EXTENT statements is not required if the logical unit was assembled into the DTFDU. One EXTENT statement is required for each diskette volume that is to be read.

The RDR statement causes card reading to be stopped and up to two diskettes of the file named FILE-ID to be read. Records on the diskette may be from 1 to 128 bytes long, and VSE/POWER does not examine them for control statements. When the end-of-file identifier is reached, card reading is resumed.

The user program may not read all of the records of file FILE-ID. To avoid that the remaining records are passed to job control as SYSIN data (once the * $$ RDR statement is reached), any request to the card input spool device causes VSE/POWER to skip records until the end of the file FILE-ID.

Data-Mode Example 2: Some job control statements are in the card reader, additional job control statements and data are on a diskette. The operator enters a PSTART command for the card reader:

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

The command causes the reader task to insert diskette data (from the device at 00B) into the input stream on the spool disk at the point where an * $$ RDR statement occurs. Assume that cards are in the card reader as follows:

```
// JOB EX1
    ... ... ...
/&
// JOB BLLNGT1
// ASSGN SYS008,04B
// DLBL  DISKET,,,DU
// EXTENT SYS008
    ... ... ...      Additional job control statements
    ... ... ...      that may be needed for the
    ... ... ...      execution of the program INVOICE.
// EXEC INVOICE
* $$ RDR DEV=04B,FID='TESTINP'
// JOB EX2
    ... ... ...
```

The * $$ RDR statement causes the reader task to stop card reading and to read from the diskette on 00B a file named TESTINP. This file could contain, for example, test records for the INVOICE program:

```
    ... ... ... ┐
    ... ... ... > Test records
    ... ... ... ┘
/*
/&
```

For more information on the use of the * $$ RDR statement, refer to "* $$ RDR: Adding 3540 Files to Input Jobs" on page 280.

TAPE PROCESSING WITH VSE/POWER

This section summarizes VSE/POWER's tape handling functions:

- Spooling output onto tape and printing this output from tape – See "Processing Spool Tapes."
- Spooling from a SYSIN tape – See "Processing for the SYSIN Tape Function" on page 116.
- Saving queue entries on tape and restoring them from tape – See "Processing for the Offload-Queue Function" on page 117.

Processing Spool Tapes

This is the function of spooling output to a tape and of reading list or punch output from such a tape.

Spooling Output to Tape

You can direct VSE/POWER to route printer or punch output to tape by specifying DISP=T in the * $$ LST statement.

If the mounted tape volume is at its load point, VSE/POWER checks the tape for the presence of standard labels. If standard labels exist, VSE/POWER prompts the operator whether it is to:

- Reject the volume if that volume contains a protected file, or
- Continue processing with the mounted tape.

If VSE/POWER is instructed to continue, it overwrites the labels on the tape and handles the tape as an unlabeled one.

If the output tape is not at its load point, it is assumed that the tape contains valid spool output. VSE/POWER writes a tape mark (overwriting the present tape mark where the tape is positioned) and begins storing the present spool data to tape.

If end-of-volume occurs (reflective marker was encountered), VSE/POWER normally forces segmentation of the output, and processing should continue with another tape. VSE/POWER requests the operator (by a message) to mount a new tape. If the new tape is a labeled one, VSE/POWER handles this tape in the same way as described above.

When spooling to tape for a job is complete, VSE/POWER releases the tape assignment with the tape positioned immediately behind the generated output file.

Input for List- or Punch-Output Processing

VSE/POWER checks that the mounted tape was created either by spooling output to tape (* $$ LST TADDR=) or by a POFFLOAD BACKUP or SAVE function. This involves a tape rewind operation.

The operator receives a message if VSE/POWER finds that a wrong tape was mounted.

When end-of-file occurs, the writer task stops. It is the operator's responsibility to ensure that subsequent tapes (if any) are processed.

Processing for the SYSIN Tape Function

To have VSE/POWER spool a job stream contained on tape you must have issued a PSTART RDR,tapeaddr command.

VSE/POWER rewinds the mounted tape volume to check for the presence of standard-labels. If standard labels exist, VSE/POWER prompts the operator for instructions either to stop the reader function or to continue.

Input can be in the form of fixed-block or unblocked records. For fixed-block input, the block length must be a multiple of 80 bytes up to a maximum of 4,080 bytes. For unblocked tape input, only fixed-length records of 80 or 81 bytes are accepted (81-byte records result from compiler and librarian programs). Unblocked input with a record length of a multiple of 80 leads to unpredictable results.

If any other record format is found, the reader task stops and the operator is informed via a message.

IBM standard-labeled tape files, multifile volumes, multivolume files, and single-volume unlabeled tape files are allowed.

End-of-File for Unlabeled Tapes

When end-of-file occurs, VSE/POWER checks the tape for an end-of-volume condition (two tape marks). In case of an end-of-volume condition, the reader task stops.

When end-of-file occurs, VSE/POWER checks the volume for additional files. If there are any, VSE/POWER processes them in the same way as described above.

Note: Unlabeled multivolume files are not supported.

End-of-File for Labeled Tapes

When end-of-file occurs (an EOF trailer label was found), VSE/POWER checks the tape for additional EOF markers (but not for the two successive tape marks behind the trailer label).

End-of-Volume for Multivolume Files

When end-of-volume occurs, VSE/POWER:

1. Unloads the affected tape.
2. Issues a tape mount request for the operator.
3. Waits for a PGO command, an instruction to continue processing with the next tape volume.

Processing for the Offload-Queue Function

To start an offload function (SAVE, BACKUP, LOAD or SELECT), you use the POFFLOAD command.

SAVE Function

For a POFFLOAD SAVE operation, the tape open and close handling by VSE/POWER is the same as described under "Spooling Output to Tape" on page 115, except when you specify NOREW. If NOREW is specified, VSE/POWER checks for the presence of standard labels only if the tape is positioned at its load point.

A POFFLOAD SAVE operation is performed in parallel with normal processing of queue entries.

For example, the command

```
POFFLOAD SAVE,LST,280,,C
```

causes list queue entries of class C to be copied onto the tape mounted on the drive at the address 280. For a list queue entry (of class C) to be copied, this entry's disposition must be K or D.

After having saved an entry, VSE/POWER deletes it from the list queue on disk if the entry's disposition was D; VSE/POWER retains the entry and sets its disposition to L if the original disposition was K.

If end-of-volume occurs, VSE/POWER retains the queue entry being processed but not successfully saved. VSE/POWER closes the tape and informs the operator by a message that a new tape must be mounted to process the remaining queue entries. When all queue entries of the specified classes have been written to tape, the task stops, and VSE/POWER issues task-complete messages.

VSE/POWER writes one tape mark at the end of every saved file; it writes two tape marks at the end of the queue entry saved last.

BACKUP Function

For a POFFLOAD BACKUP operation, the tape open and close handling by VSE/POWER is the same as described under "Spooling Output to Tape" on page 115, except when you specify NOREW. If NOREW is specified, VSE/POWER checks for the presence of standard labels only if the tape is positioned at its load point.

A POFFLOAD BACKUP operation does not permit concurrent queue access by other VSE/POWER tasks. You should therefore ensure that none of your system's users waits for urgent output when you schedule a BACKUP operation.

Assume you issue the command

```
POFFLOAD BACKUP,LST,280,,C
```

This causes all list queue entries of class C, regardless of their assigned disposition, to be copied onto the tape mounted on the drive at the address 280.

Queue entries saved by a BACKUP operation are retained in their queue unchanged.

LOAD Function

You use the POFFLOAD LOAD command to restore (load) queue entries written onto tape by using the POFFLOAD BACKUP or the POFFLOAD SAVE command. You can also use the POFFLOAD LOAD command to reload a spool tape (DISP=T) back into the VSE/POWER queues.

If you specify NOREW in the POFFLOAD LOAD command, ensure that the tape is correctly positioned before you issue the command.

The load operation ends if end-of-volume is recognized or when the last queue entry on the tape is loaded. If multiple volumes are used (not multivolume files), an extra POFFLOAD LOAD command has to be given for each volume.

VSE/POWER jobs that are restored to queues get a new job number (and a new creation date), except when you request that the original job numbers (and creation dates) be retained. You can request this by specifying NOJNO in your POFFLOAD LOAD or SELECT command. A job suffix (output-segment number) remains unchanged.

If the queue identifier on the tape does not match the identifier given in the POFFLOAD LOAD command, then VSE/POWER rejects the tape and stops the task. You are informed of this by a message.

In your POFFLOAD SAVE (or POFFLOAD BACKUP) command, you may have specified NOREW to save several versions of the VSE/POWER queues. To load only a certain queue from that tape, you must position the tape at the beginning of this queue.

SELECT Function

You use the POFFLOAD SELECT command to reload only certain queue entries from a POFFLOAD or spool tape into the VSE/POWER queues.

For example, the command

```
POFFLOAD SELECT,LST,280
```

causes the following message to be displayed:

```
1R41D SPECIFY TAPE SELECT CRITERIA OR PRESS ENTER TO QUIT
```

Reply with one or more selection operands as shown in the description of the POFFLOAD command.

Thus, if you reply with *PAYRL, for example, VSE/POWER reloads, back to the LST queue, all saved queue entries whose job names begin with the letters PAYRL. The tape to be used is mounted on the drive at address 280.

Stopping a POFFLOAD Operation

A POFFLOAD operation can be stopped immediately or after a queue entry has been completely saved or restored. You do this by submitting a PSTOP command.

Assume that a POFFLOAD SAVE operation is in process writing to a tape on the drive at the address 180. If the command

```
PSTOP 180
```

is issued, the save operation stops immediately. VSE/POWER unloads the tape and writes an information message to the console. The queue entry being processed when the PSTOP command was given remains in its queue unchanged.

Stopping a POFFLOAD SAVE or BACKUP operation results in an incorrect end-of-input indication on tape. When you try to print or punch from this tape, the print (punch) job for the interrupted queue entry ends abnormally.

Stopping a POFFLOAD LOAD or SELECT operation causes VSE/POWER to drop the currently processed queue entry without restoring it to disk and to issue an information message to the console.

TERMINATING VSE/POWER

VSE/POWER may be terminated normally when, for example, none of the programs to be run need VSE/POWER. It may end abnormally or it may have to be terminated in an emergency situation.

Normal Shut-Down

Before you terminate VSE/POWER, inform any operator of remote BSC work stations that the shutdown procedure is about to begin (for SNA work station users, a message is issued automatically). Then enter the command

```
PEND 00E
```

This causes the following:

1. All active tasks, including RJE and PNET tasks, continue processing until they reach the end of the VSE/POWER job they are processing.

2. If the PNET function is used, VSE/POWER sets all transmitters to indicate that they continue to transmit only until the end of the current job, and then stop.

3. VSE/POWER releases each of the partitions under its control as soon as program execution in the partition is finished.

4. VSE/POWER prints a status report on the printer at the address 00E. This status report consists of a statistics part, as shown in Figure 12 on page 121, and a printout of the contents of all VSE/POWER queues, which is identical to a PDISPLAY ALL,LST status display (as shown on page 172). If you do not specify the address of a printer, VSE/POWER does not print the status report.

```
VSE/POWER STATUS REPORT FOR xxxxxxx     05/06/85       09/44/10

QUEUE FILE          IJQFILE

TOTAL NUMBER OF TRACKS                                      12 TRACKS
TOTAL NUMBER OF QUEUE RECORDS                              750 RECORDS
NUMBER OF FREE QUEUE RECORDS                               625 RECORDS
MAX. NUMBER OF QUEUE RECORDS USED SINCE LAST COLDSTART     127 RECORDS
NUMBER OF QUEUE RECORDS LOST DUE TO I/O ERROR                0 RECORDS
QUEUE FILE STORAGE COPY VIO SPACE                          192 K-BYTES

DATA FILE           IJDFILE

TOTAL NUMBER OF TRACKS                                    2850 TRACKS
TOTAL NUMBER OF DBLK-GROUPS                               2565 GROUPS
NUMBER OF FREE DBLK-GROUPS                                1938 GROUPS
MAX. NUMBER OF DBLK-GROUPS USED SINCE LAST COLDSTART       627 GROUPS
NUMBER OF DBLK-GROUPS LOST DUE TO I/O ERROR                  0 GROUPS
DATA BLOCK GROUP SIZE                                       10 DBLKS
DATA BLOCK SIZE                                           1952 BYTES
SPOOL LIMIT PERCENTAGE                                      90 %

ACCOUNT FILE            IJAFILE

TOTAL NUMBER OF TRACKS                                      90 TRACKS
PERCENTAGE OF FILE THAT IS FILLED                           11 %

FIXABLE STORAGE USED BY VSE/POWER                           64 K-BYTES
VIRTUAL STORAGE OCCUPIED BY VSE/POWER PHASES               312 K-BYTES
NUMBER OF TIMES TASKS WERE WAITING FOR PFIXED STORAGE        0 TIMES
MAXIMUM NUMBER OF K-BYTES FIXED IN PRESENT SESSION          36 K-BYTES
MAXIMUM NUMBER OF TASKS ACTIVE AT ONE POINT IN TIME         19 TASKS
NUMBER OF TIMES TASKS WERE WAITING FOR VIRTUAL STORAGE       0 TIMES
MAXIMUM GETVIS STORAGE REQUESTED IN PRESENT SESSION         17 K-BYTES

TIME INTERVALS FOR SHARED SPOOLING (SYSID=2):
T1 =  5 SECONDS, T2 = 0 SECOND, T3 =  60 SECONDS, T4 = 180 SECONDS

NUMBER OF NOTIFY MESSAGES LOST:                          0 MESSAGE(S)
```

Figure 12 (Part 1 of 2). Example of the Statistics Part of a VSE/POWER Status Report After PEND

If VSE/POWER has been generated with job accounting support deactivated, the account file statistics are replaced by the message

```
NO ACCOUNT SUPPORT
```

The time intervals (at the bottom of the above example) are printed only if VSE/POWER operates in a shared spooling environment.

Statistics provided by VSE/POWER after a warm start give information about activities from the time VSE/POWER was started until the time these statistics are displayed. The MAX. NUMBER OF QUEUE RECORDS... line gives the number of active queue records - 127 in the example.

The above statistics are followed by a status display of all VSE/POWER queue files, as shown on page 172.

Figure 12 (Part 2 of 2). Example of the Statistics Part of a VSE/POWER Status Report After PEND

Abnormal End of VSE/POWER

If VSE/POWER ends abnormally, it informs you by a message and prompts you to specify whether or not you want a storage dump of the VSE/POWER partition. If you reply with YES, VSE/POWER prompts you whether or not a printout (SYSLST format) of the virtual-storage copy (VIO) of the queue file is required. If you reply with a valid printer or tape address, VSE/POWER prints, on the specified device, the VIO storage containing the virtual-storage copy of the queue file.

If a dump sublibrary has been defined for the VSE/POWER partition, VSE/POWER writes a dump of the partition into this sublibrary. If no such sublibrary exists or if the sublibrary is full, VSE/POWER prompts you for the address of the printer or tape on which the dump is to be written.

If the dump is to be written on a tape, it is your responsibility to write a final tape mark (by means of the // CLOSE job control statement) on the tape. Also, ensure that the tape does not include any secured data.

Shared Spooling Considerations

If a sharing system abnormally ends while it is processing a job, there are two possible effects for the other sharing systems:

- If the failing system had write access to the control information in the spool file at the time of termination, the sharing system receiving control next performs recovery for the abnormally terminated system.

- If the failing system did not have write access to the control information in the spool file, the other sharing systems are not aware that this sharing system is no longer active. Any job that was in process when the sharing system ended is still being flagged as active when a PDISPLAY command is issued. This means that the job is not available to any other sharing system for execution or printing.

  If the job is urgently needed, and the terminated sharing system cannot be restarted in time, the operator of any other active sharing system should enter the PRESET command and specify the SYSID of the terminated sharing system. This sets all active jobs for this sharing system to inactive. The operator is informed about the recovery action.

Emergency Termination

When it is necessary to terminate VSE/POWER immediately because of an emergency situation, submit the command

```
PEND FORCE
```

As a result of this command, input and output for the currently processed jobs is broken off. A warm start enables you to restart interrupted output from the beginning. An input job stream from a unit record device or from a remote work station is lost and must be resubmitted. The same is true for a job or for output submitted from a partition.

Before VSE/POWER releases its resources, it asks you whether a storage dump is desired (this is the same as described above, under "Abnormal End of VSE/POWER" on page 122.)

CHAPTER 4. VSE/POWER CENTRAL OPERATOR COMMANDS

This chapter gives a detailed description of the VSE/POWER central operator commands. The booklet VSE/POWER Reference Summary provides only a summary of these commands and their formats. You can use these commands at the central system to control the operation of VSE/POWER and the processing of jobs under VSE/POWER control.

VSE/POWER assigns a number to each job that it logs. This enables you to distinguish between jobs for which identical names were specified. You must use a job's name and, possibly, the assigned number whenever you issue a command to control the processing of this job.

Once a job has been logged by VSE/POWER, that job is referred to as a VSE/POWER job or as a queue entry.

The first LST (and the first PUN) output from a VSE/POWER job has the same number as the input job. However, any subsequent LST or PUN output is a separate queue entry and has a unique job number. This aids in manipulating the processing of VSE/POWER jobs.

At a remote work station, you use the VSE/POWER remote operator commands; they are not discussed in this manual. For a detailed description of their formats and functions, refer to the publication VSE/POWER Remote Job Entry User's Guide.

Throughout this chapter, the information given for the IBM 3800 Printing Subsystem applies also to the IBM 3200 Printing Subsystem, if not stated otherwise.

Overview of the Commands

The central operator commands allow you to control, besides VSE/POWER's central operation, also its remote job entry and networking functions. You enter the commands in the same way as attention routine commands. These commands can be submitted any time after VSE/POWER start-up is complete.

Figure 13 on page 127 gives an overview of the VSE/POWER central operator commands. As Figure 13 shows, the commands can be divided into logical groups as follows:

- Queue Management Commands. They control the various VSE/POWER input and output queues.
- Task Management Commands. They control the reader/writer tasks, the execution processors, and network operation.

- Control Commands. They request service functions such as routing commands or messages to another node or printing one or more layout pages for the purpose of forms adjustment.

A 1-character short form can be used for most of the command codes. Available short forms are shown in Figure 13.

Command Code	Short Form	Function
Queue Management Commands		
PALTER	A	Alter the processing attributes of a VSE/POWER job or a VSE/POWER controlled partition.
PDELETE	L	Delete queue entries or messages.
PDISPLAY	D	Display the status of jobs, messages, resources and the network.
PHOLD	H	Put a job of a queue in the hold or leave state.
POFFLOAD	O	Save or restore one or more entries of a queue.
PRELEASE	R	Release VSE/POWER jobs from the hold or leave state for further processing.
Task Management Commands		
PACT		Activate a transmitter or receiver task.
PCANCEL	C	Cancel a status display or a VSE/POWER job that is being executed.
PDRAIN	N	Discontinue job or output transmission by a transmitter task or discontinue receiving of jobs or output by a receiver task.
PEND		End VSE/POWER operation.
PFLUSH	F	Discontinue the work currently in progress for a task and allow the task to continue with subsequent work.
PGO	G	Reactivate a task or a partition.
PRESTART	T	Restart a writer task.
PSTART	S	Place a partition under the control of VSE/POWER, start a task, or initiate a link or a session between two nodes.
PSTOP	P	Release a partition from VSE/POWER control, stop a task, or end a link or session between two nodes.

Figure 13 (Part 1 of 2). VSE/POWER Central Operator Commands

Command Code	Short Form	Function
Control Commands		
PACCOUNT	J	Save account file records.
PBRDCST	B	Transmit a message.
PINQUIRE	I	Display the status of one or all of the following: - A BSC line - An SNA logical unit - A device under control of an application program in another partition - A node of the network
PLOAD		Load a network definition table.
PRESET		Reset active jobs in a shared-spooling environment.
PSETUP	U	Print the page layout of one or more pages.
PXMIT	X	Route commands to another node or to an application program that controls the writing of spooled output records to a device.

Figure 13 (Part 2 of 2). VSE/POWER Central Operator Commands

Format of the Commands

A VSE/POWER central operator command consists of:

1. A command verb (operation field) - You can specify the full command verb or just its short form, if there is one.

2. None, one, or more operands (operand field).

   For the symbols used in showing the format of operands, refer to "Conventions for Showing Command and Statement Formats" on page 15.

   Note: If a keyword operand occurs more than once, VSE/POWER uses the specification supplied with the operand repeated last.

PACCOUNT: Empty the VSE/POWER Account File

The command empties the VSE/POWER account file and, optionally, saves the processed account file records.

VSE/POWER issues a message whenever the account file is 80% full; it indicates, again by message, when the account file has been saved. You receive also a message if a VSE/POWER task requires space for writing an account record and no such space is available.

Use the command with care if the account file is being shared with other systems. Another sharing system that needs to access the account file goes into a wait until processing of the command is complete.

Format of the Command

Operation	Operands
{PACCOUNT|J}	{DEL| DISK,dfilename| PUN| tapeaddr,[density][,tfilename]}

Description of Operands

DEL
: Specify DEL if the records stored in the account file are to be deleted and the file is to be cleared.

DISK,dfilename
: Specify this operand if VSE/POWER is to write the contents of the account file onto disk. The file, which is used to save the currently stored account records, must reside on a disk device of the same type as is used for VSE/POWER's account file (IJAFILE).

  For dfilename, specify the seven-character file name which you used for defining the file to the system in the DLBL statement for the file. An ASSGN statement for the logical unit chosen in the EXTENT statement must also be given before VSE/POWER start-up.

PUN
: Specify PUN if the contents of the account file are to be spooled to the punch queue with:

  Priority 1
  PACCOUNT job name
  Output class P
  Disposition H

To retrieve the account file records, start a punch task for output class P.

tapeaddr,[density][,tfilename]

For tapeaddr, specify the address of the tape drive to which the file is to be written. Specify this address in the form cuu (channel and unit number).

For density, specify the mode setting that applies to the tape drive which is to be used. For a list of applicable mode codes, refer to VSE/Advanced Functions, System Control Statements.

For the IBM 3480, density is ignored.

If you use an IBM 8809, VSE/POWER does not accept a streaming mode specification (30 or 90).

For tfilename, specify the name of the file if the file is to be created on a tape with standard labels. Your specification for tfilename must match the seven-character file name in the TLBL statement.

If you omit tfilename, VSE/POWER treats the file as a standard unlabeled VSE tape file.

Examples of the PACCOUNT Command

PACCOUNT 180
Writes the account file to an unlabeled tape mounted on drive 180.

PACCOUNT 180,,ACCTFLE
Writes the account file onto tape as a standard-label file.

PACCOUNT DEL
Deletes all account file records.

PACCOUNT PUN
Spools account-file records to the punch queue with an output class of P and the job name PACCOUNT.

PACT: Activate a Transmitter/Receiver

This command applies only to networking. It activates or reactivates transmitter or receiver tasks. You can use this command any time between PSTART and PSTOP for your node, and before you issue a PEND command.

VSE/POWER supports up to eight concurrently active transmitters and receivers.

Format of the Command

Operation	Operands
PACT	PNET,nodeid,{RVn|TRn},{JOB|OUT}

Description of Operands

PNET Code this operand as shown.

nodeid For nodeid, specify the identifier of the node for which the named transmitter or receiver task is to be activated or reactivated.

RVn|TRn For n, specify the number of the receiver (RV) or transmitter (TR) task that is to be activated or reactivated. You can specify any integer from 1 to 7.

VSE/POWER automatically activates the TR1 tasks for job and output transmission when a connection is established in response to a PSTART PNET command. VSE/POWER activates, at the same time, the tasks RV1 through RV7.

JOB|OUT Specify JOB if the named transmitter or receiver task is to be used for jobs.

Specify OUT if the named transmitter or receiver task is to be used for list or punch output.

Example of the PACT Command

```
PACT PNET,NODEA,TR3,OUT
```

Activates the output-transmitter task 3 for node NODEA.

PALTER: Alter Job Attributes or Partition Classes

VSE/POWER accepts a PALTER command with up to nine operands.

The command alters either of the following:

- The attributes (such as the priority or the disposition) of a queued VSE/POWER job.

  With one PALTER command, you can request VSE/POWER to change one or more attributes of a job or of a certain group of jobs. A change of the class, destination, disposition, or priority of a job in an input queue places this job at the end of the specified class or priority group.

  The attributes of a job that is being executed cannot be altered. For a job whose output is being processed (this output is already in the list or punch queue), only the number-of-copies attribute can be changed, except if this output is being transmitted to a 3790 workstation (with PDIR=FMH2).

  A job's class, disposition, and priority can be different in the input and output queues. You specify job-related attributes in the * $$ JOB statement; you may override output-related attributes in either of the statements * $$ LST and * $$ PUN.

- The input classes of a partition.

  The assignment of classes to an active partition under control of VSE/POWER can be changed without stopping the partition and without releasing the assignments for the partition.

Format of the Command

There are two distinct command formats: for altering the attributes of a job or a group of jobs and for altering the class assignment to a partition. These formats are shown in the box on the next page.

Operation	Operands
	Search Operands
{PALTER|A}	queue[,*abc| ,ALL| ,class| ,jobname[,jobnumber[,jobsuffix]]] [,CCLASS=class] [,CDISP=disposition] [,CFNO=formnumber] [,CPRI=priority] [,CRDATE{=|>|<|¬=|>=|<=}date] [,CSYSID=sysid] [,CUSER=userid] [,FNODE=nodeid] [,FUSER=userid] [,TNODE=nodeid] [,TUSER=userid]
	Action Operands
	[,CLASS=class] [,CMPACT={*|name|NO}] [,COPY=number-of-copies] [,DEST={nodeid|LOCAL|SYSTEM}] [,DISP={disposition|*}] [,FNO={*|formnumber}] [,NODE={nodeid|LOCAL|SYSTEM}] [,PRI=priority] [,REMOTE=remid] [,SYSID={sysid|N}] [,USER={userid|LOCAL|SYSTEM}]
	VSE/POWER requires at least one action operand.
{PALTER|A}	partition[,class]

The search operands are discussed below; for a discussion of the action operands, refer to "Description of the Action Operands" on page 137; for a discussion of input-class related operands, see "Description of Partition Input Class Related Operands" on page 139.

Description of the Search Operands

queue The operand specifies the VSE/POWER queue for which the command is to be executed. Specify:

LST For the list queue.
PUN For the punch queue.
RDR For the reader queue.
XMT For the transmission queue.

*abc Use this form as additional search argument if you want VSE/POWER to change specific attributes of jobs whose names begin with the specified characters. For abc, you can specify up to eight characters that are common in the names of a group of queued VSE/POWER jobs.

ALL Specify ALL if specific attributes are to be altered for all VSE/POWER jobs in the named queue or just for a certain group of jobs in the queue as defined by one or more additional keyword-type search operands.

VSE/POWER uses ALL as default if you specify none of the group of selectable operands of which ALL is a part.

class For class, specify the applicable job class if certain job attributes are to be altered for all VSE/POWER jobs of a specific class.

jobname[,jobnumber[,jobsuffix]]
If your command applies to a specific job, specify for:

jobname
The name by which the job is known to VSE/POWER.

jobnumber
The number assigned to the job by VSE/POWER. This is of significance if two or more jobs with the same name are queued in the specified queue. Use the PDISPLAY command to get this number displayed by VSE/POWER.

jobsuffix
The VSE/POWER-assigned segment number if only the characteristics of a certain output segment of the named job are to be changed. You can use this operand only together with the VSE/POWER assigned job number. To get the correct jobsuffix, issue a PDISPLAY command for the job with FULL=YES specified.

Specifying PALTER with queue,jobname (optionally with jobnumber) causes VSE/POWER to do the requested change for the first occurrence of this name (and number).

Exceptions:

1. If the applicable queue entries are the result of count-driven segmentation, all related segment entries are also changed. If you specify also a job suffix, the alter request applies only to the specified segment of the named job.

2. If you specify a jobname with one of the remaining search operands (CCLASS to TUSER), VSE/POWER does the requested change for all occurrences of this name.

CCLASS=class
: The operand indicates that the requested change is to be done if the class currently assigned to the queue entry (or entries) matches the specified class.

CDISP=disposition
: The operand indicates that the requested change is to be done if the disposition currently assigned to the queue entry (or entries) matches the specified disposition.

CFNO=formnumber
: The operand indicates that the requested change is to be done if the form number defined for the output of the queue entry (or entries) matches the specified form number.

CPRI=priority
: The operand indicates that the requested change is to be done if the priority currently assigned to the queue entry (or entries) matches the specified priority.

CRDATE{=|>|<|¬=|>=|<=}date
: In this string:

  CRDATE — indicates that the requested change is to be done for those queue entries whose creation date is one of the following:

  - Equal to (=)
  - Greater than (>)
  - Less than (<)
  - Not equal to (¬=)
  - Greater than or equal to (>=)
  - Less than or equal to (<=)

  > the specified date

  date — specifies the date against which the queue entries' creation date is to be compared. It must have the format defined for the system (mm/dd/yy or dd/mm/yy). Do not specify mm greater than 12, dd greater than 31, and yy greater than the current year.

Note: In a shared spooling environment, all systems must have the same date format.

CSYSID=sysid
: The operand indicates that the requested change is to be done if the processing system defined for a queue entry (in a shared spooling environment) matches the specified processing system.

CUSER=userid
: The operand indicates that the requested change is to be done if the queue entry's "from user" or "to user" matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

FNODE=nodeid
: The operand indicates that the requested change is to be done if the queue entry's "from node" name matches the specified node name. However, VSE/POWER does not verify that the specified node name is defined in the network-definition table.

FUSER=userid
: The operand indicates that the requested change is to be done if the queue entry's "from user" identifier matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause a change of attributes of entries originated by local users.

TNODE=nodeid
: The operand indicates that the requested change is to be done if the queue entry's "to node" name matches the node name given in this operand. This operand is valid only if your specification for queue is XMT.

TUSER=userid
: The operand indicates that the requested change is to be done if the queue entry's "to user" identifier matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  The TUSER specification cannot be used for altering queue entries residing in the reader queue.

  Specifying TUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause a change of attributes of entries destined for local users.

Description of the Action Operands

CLASS=class
: The operand specifies the new class that is to be assigned to the named VSE/POWER job(s).

CMPACT={*|name|NO}
: The operand specifies whether data compaction is to be performed for the transmission of job output to an SNA work station. You can specify:

  - An asterisk (*) to indicate that the default compaction table (defined in the PRMT generation macro) is to be used.
  - For name, the four-character name of the compaction table that is to be used.
  - NO to indicate that no compaction is to be performed.

COPY=number-of-copies
: The operand specifies the number of copies to be produced after the command is issued, including the one in progress. Your specification, which may range from 1 to 255, applies only to output queue entries. The specification is ignored if it is given for an input queue entry.

  If the command refers to output on a 3800 printer, this operand changes only the transmission count. This count is displayed on the first line of a status report, while the total number of copies still to be produced, including the transmission in progress, is displayed on the second line.

  If you reduce the number of transmissions while copy grouping is done and increase this number again with a later PALTER command, then VSE/POWER restores the previously reduced group values; if the new number of transmissions is greater than the original copy-grouping value, VSE/POWER just adds 1 to the restored values.

DEST={nodeid|LOCAL|SYSTEM}
: This operand has the same effect as the NODE operand, which you should use instead. VSE/POWER accepts the DEST operand for the reason of compatibility.

DISP={disposition|*}
: The operand gives the new disposition of the affected VSE/POWER job(s). You can specify:

  D For dispatchable
  H For hold in queue
  K For keep after processing
  L For leave in queue

Use DISP=* when you want to change the temporary disposition X, A or Y of one or more queue entries back to the original disposition.

Note: A queue entry with a disposition of A, X or Y is not automatically processed by the VSE/POWER tasks. Also, the PRELEASE command does not release such a queue entry, nor does a PHOLD command change a temporary disposition. Therefore, you must change a disposition of A, X or Y to one of the valid dispositions before the queue entry can be processed.

To obtain a list of all queue entries with disposition Y, for example, issue a PDISPLAY ALL,CDISP=Y command.

For further information on dispositions refer to Appendix A on page 309.

FNO={*|formnumber}

Use the operand to change the four-character form-number specification for the affected queue entry. Specify:

- * if the affected output is to be processed with the location's standard form. For VSE/POWER, this form has no form number.
- For formnumber, the four-character number of the form that is to be used for processing the affected output.

NODE={nodeid|LOCAL|SYSTEM}

The operand specifies the new final destination. Specify:

- For nodeid, the new node identifier. The nodeid you specify must be defined in the active network definition table.
- LOCAL or SYSTEM to tell VSE/POWER that the current destination identifier must be replaced by the identifier of your own node. This causes the queue entry (or entries) to be removed from the XMT queue and added to the applicable local queue.

PRI=priority

The operand specifies the new priority (an integer from 0 to 9, where 9 is the highest priority) of the named VSE/POWER job(s).

REMOTE=remid

Your specification, the new remote identifier, applies only to output. The specification is ignored if given for an input queue entry.

For remid, specify the identifier of the work station to which the applicable output is to be routed. Specify this identifier in the form Rnnn or just nnn (where nnn = the remote identifier

as defined for VSE/POWER table generation). If you specify 000 (or just 0), VSE/POWER routes the output to a local output device in accordance with existing class and priority assignments.

SYSID={sysid|N}
The operand applies to a shared spooling environment. Specify:

- For sysid, a new system identifier if another system is to be used for processing the affected job(s) or output.
- N to have VSE/POWER make the affected job(s) or output available for processing on any of the sharing systems.

USER={userid|LOCAL|SYSTEM}
Use the operand to change the user identifier of the affected queue entry (or entries). Specify:

- For userid, the identifier to be used by VSE/POWER. This identifier must be known to your VSE system or to the target node, whichever applies. Specify a remote work station in the form Rnnn or nnn (where nnn = the remote identifier as defined for VSE/POWER table generation).
- LOCAL or SYSTEM to have VSE/POWER set the user identifier currently assigned to the affected entry (or entries) to blanks. This causes VSE/POWER to handle the job(s) in accordance with the existing class and priority assignments.

Note: Since in a VSE/SP environment no user except the central operator may execute jobs, VSE/POWER does not allow to assign a TO-userid to an entry in the RDR queue by the PALTER RDR,...,USER= command. This is only acceptable for job entries residing in the XMT queue, which might be transmitted to another (MVS or VM) sytem. Therefore, if an XMT queue entry with a TO-userid is requeued into the RDR queue, the existing TO-userid will not be displayed.

Description of Partition Input Class Related Operands

```
{PALTER|A}        partition[,class]
```

Use this command format to change the class assignment for a VSE/POWER controlled partition.

partition
For partition, specify the applicable partition's identifier (BG or Fn).

class For class, you can specify up to four input classes of jobs that are to be processed in the partition. As a class, specify any letter of the alphabet or the partition number. If you do not specify any class(es), only partition dependent jobs can run in the specified partition.

Examples of the PALTER Command

```
PALTER XMT,ALL,TNODE=NODEA,NODE=LOCAL
```
Searches the transmission queue for entries that are to be transmitted to NODEA; changes each of those entries to the local queues (job input to RDR, list output to LST, and punched output to PUN).

```
PALTER RDR,ALL,CPRI=3,PRI=5
```
Searches the reader queue for all priority 3 entries and changes all such entries to a priority of 5.

```
PALTER LST,CDISP=L,DISP=K,PRI=5
```
Searches the list queue for all disposition L entries; changes the disposition to K and sets the priority of these entries to 5.

```
PALTER LST,CRDATE<01/31/87,DISP=D
```
Searches the list queue for all entries that are older than January 31, 1987 and changes their disposition to D.

```
PALTER LST,PAYROLL,DISP=H,CLASS=B
```
Changes, for the list output of the job PAYROLL, the disposition to H and the class to B.

```
A PUN,*PAYRL,DISP=D
```
Sets to D the disposition of a VSE/POWER job in the punch queue if the name of the job begins with PAYRL.

```
PALTER LST,P,CLASS=Q
```
Changes all class P jobs in the LST queue to class Q.

```
A F1,ABC1
```
Changes partition F1 to accept input classes A, B, C, and 1.

PBRDCST: Send Messages

The command enables you to send (or broadcast) a message to:

- A specific user or to all interactive users in one of the following environments:
  - CMS under VM/SP if your virtual VSE system includes the networking support and can link to RSCS of VM/SP.
  - TSO under MVS/SP
  - VSE/ICCF

  The user(s) may be linked to your own node or to another node.

- A remote work station.

  A message routed to a remote work station is displayed as soon as this work station is ready to receive the message.

Format of the Command

Operation	Operands
{PBRDCST|B}	{nodeid[,userid]| *,userid| remid| ALLUSERS},'text'

Description of Operands

nodeid[,userid]|*,userid

For nodeid, specify the name of the destination node if your message is to be sent to another node in the network.

Specify an asterisk (*) instead of a node name if the message is to be sent to a user at your own node, that is, to a user of the same VSE system.

For userid, specify either of the following:

- The identifier of the user to whom the message is to be routed.
- ALLUSERS to have VSE/POWER place your message into the ALLUSERS message queue at the specified node, which must be a VSE system with VSE/POWER. A user at the destination node can request the message to be displayed if this node operates with VSE/POWER. The PDISPLAY MSG command is available for this purpose.

If you specify ALLUSERS together with the name of a node without VSE/POWER, this specification is taken as a user identification. Your message gets lost if ALLUSERS is not defined as the name of a user at the destination node.

If you do not specify a user-identifier, VSE/POWER directs your message to the operator of the specified node.

VSE/POWER rejects the command if there is no connection available for forwarding the message. If no such connection is available at another (forwarding) node, your message gets lost.

remid If your message is to be sent to an RJE work station connected to your own system, specify for remid this work station's identifier (in the form nnn or Rnnn). Specify 0 for remid to have VSE/POWER route the message to your system's operator (should the message be submitted from a partition, for example).

ALLUSERS
Specify this operand if your message is intended for all remote users connected to your system.

VSE/POWER queues ALLUSERS messages and assigns a sequence number to them. You can obtain the message (and also the assigned number) by issuing a PDISPLAY MSG command. Up to 16 ALLUSERS messages can be queued by VSE/POWER at a time.

'text' The operand represents the text of your message. Your text, words and numbers, must be enclosed within a pair of apostrophes. An apostrophe (') within the text must be entered as two apostrophes ('').

The maximum length of the message text, including the enclosing apostrophes, is as follows:

- For an ALLUSERS-type message: 46 characters.
- For a transmission within a network: 132 characters.
- For messages to a work station of your own system: 60 characters.

If the length of a message exceeds these values, VSE/POWER truncates the text at the end. You may issue two or more PBRDCST commands, if there is a need.

Examples of the PBRDCST Command

```
PBRDCST 150,'RJE WILL SHUT DOWN IN 30 MINUTES'
    Broadcasts this message to the user with the remote identifier 150.

PBRDCST ALLUSERS,'RJE STARTS TOMORROW AT 800 HRS.'
    Places this message into the ALLUSERS message queue.

B NODEB,'AT WHAT TIME WILL YOU SHUT DOWN.'
    Broadcasts this message to the central operator at NODEB.

B NODEC,USER2,'YOUR PAYROLL JOB ABENDED, WHAT SHOULD I DO'
    Broadcasts this message to the user USER2 at node NODEC.

B *,TU03,'ARE YOU LOGGED ON'
    Broadcasts this message to the VSE/ICCF user TU03.
```

PCANCEL: Terminate a Status Display or Cancel a Job

You can use the command to cancel one of the following:

- A status display started by a PDISPLAY command:
  - A queue display
  - A tape display
  - A display of VIO storage
- Further processing of a job or the output of a job.

Format of the Command

Operation	Operands
{PCANCEL|C}	[STATUS|jobname[,jobnumber]]

Description of Operands

STATUS Specify STATUS (or no operand at all) if you want VSE/POWER to discontinue the display of a status report in response to a PDISPLAY command.

However, a PCANCEL STATUS command sent to another node (via PXMIT) has no effect because processing of the transmitted PDISPLAY command is completed by the time the transmitted PCANCEL command can be processed.

jobname[,jobnumber]
Use this operand if a specific job is to be canceled. Specify for:

jobname
The name of the job that is to be canceled.

jobnumber
The VSE/POWER assigned job number if two or more jobs have the same name. You can obtain this job number by issuing the command PDISPLAY queue,jobname.

In an interactive environment, VSE/POWER accepts a command with this operand only if you are the originator of the job.

Examples of the PCANCEL Command

PCANCEL
    Terminates the display initiated by a PDISPLAY command.

C DIRDSPLY
    Cancels the execution of job DIRDSPLY.

PDELETE: Delete Jobs or Messages

The command deletes messages or removes one or more VSE/POWER jobs from the specified queue. Jobs can be removed from the VSE/POWER queues by any sharing system. This means that the operator at one sharing system must be careful when deleting jobs that could be required by another sharing system.

Jobs being executed in a partition or being processed by a writer task are not affected by this command.

VSE/POWER accepts a PDELETE command with up to nine operands.

Format of the Command

There are two distinct sets of operands: queue related and message related.

Operation	Operands
{PDELETE|L}	queue[,*abc| ,ALL| ,class| ,jobname[,jobnumber[,jobsuffix]]] [,CCLASS=class] [,CDISP=disposition] [,CFNO=formnumber] [,CPRI=priority] [,CRDATE{=|>|<|¬=|>=|<=}date] [,CSYSID=sysid] [,CUSER=userid] [,FNODE=nodeid] [,FUSER=userid] [,TNODE=nodeid] [,TUSER=userid]
	Besides the operand queue, VSE/POWER requires at least one additional operand.
{PDELETE|L}	MSG[,ALL|,number]

Following is a discussion of the queue-related operands; for a discussion of the message-related operands, refer to "Description of Message-Related Operands" on page 149.

Description of Queue-Related Operands

queue The operand specifies the VSE/POWER queue for which the command is to be executed. Specify:

LST For the list queue
PUN For the punch queue
RDR For the reader queue
XMT For the transmission queue

*abc Use this form as a search argument if you want VSE/POWER to delete those jobs whose names begin with the specified characters. For abc, you can specify up to eight characters that are common in the names of a group of queued VSE/POWER jobs.

ALL Specify ALL if all entries of the named queue are to be deleted. ALL is the default if you specify a queue with:

1. None of the group of selection operands of which ALL is a part, and
2. One or more keyword-type operands (in order to have VSE/POWER delete a certain group of entries in the queue).

class For class, specify the applicable job class if all VSE/POWER jobs of this class are to be deleted.

jobname[,jobnumber[,jobsuffix]]
If your command applies to a specific job, specify for:

jobname
The name by which the job is known to VSE/POWER.

jobnumber
The number assigned to the job by VSE/POWER. This is of significance if two or more jobs with the same name are queued in the specified queue. Use the PDISPLAY command to get this number displayed by VSE/POWER.

jobsuffix
The VSE/POWER-assigned segment number if only a specific output segment of the named job is to be deleted. You can use this operand only together with the VSE/POWER assigned job number. To get the correct jobsuffix, issue a PDISPLAY command for the job with FULL=YES specified.

Specifying PDELETE with queue,jobname (optionally with jobnumber) deletes the first queue entry found with this name (and number), except when the affected queue entries are the result of count-driven segmentation. In that case, all related segment entries are also deleted. If you specify also a job suffix, the delete request applies only to the specified segment of the named job.

CCLASS=class
: The operand indicates that the requested deletion is to be done if the class currently assigned to the queue entry (or entries) matches the specified class.

CDISP=disposition
: The operand indicates that the requested deletion is to be done if the disposition currently assigned to the queue entry (or entries) matches the specified disposition.

CFNO=formnumber
: The operand indicates that the requested deletion is to be done if the form number defined for the output of the queue entry (or entries) matches the specified form number.

CPRI=priority
: The operand indicates that the requested deletion is to be done if the priority currently assigned to the queue entry (or entries) matches the specified priority.

CRDATE{=|>|<|¬=|>=|<=}date
: In this string:

  CRDATE — indicates that the requested deletion is to be done for those queue entries whose creation date is one of the following:

Equal to (=)	> the specified date
Greater than (>)	
Less than (<)	
Not equal to (¬=)	
Greater than or equal to (>=)	
Less than or equal to (<=)	

  date — specifies the date against which the queue entries' creation dates are to be compared. It must have the format defined for the system (mm/dd/yy or dd/mm/yy). Do not specify mm greater than 12, dd greater than 31, and yy greater than the current year.

  Note: In a shared spooling environment, all systems must have the same date format.

CSYSID=sysid
: The operand indicates that the requested deletion is to be done if the processing system defined for a queue entry (in a shared spooling environment) matches the specified processing system.

CUSER=userid
: The operand requests an affected queue entry to be deleted if the entry's "from user" or "to user" identifier matches the identifier given in this operand. For a definition of userid,

refer to section "Conventions for Showing Command and Statement Formats" on page 15.

FNODE=nodeid
: The operand requests an affected queue entry to be deleted if the entry's "from node" name matches the specified node name.

FUSER=userid
: The operand requests an affected queue entry to be deleted if the entry's "from user" identifier matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause a deletion of entries originated by local users.

TNODE=nodeid
: The operand requests an affected queue entry to be deleted if the entry's "to node" name matches the node name given in this operand. This operand is valid only if your specification for queue is XMT.

TUSER=userid
: The operand requests an affected queue entry to be deleted if the entry's "to user" identifier matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  The TUSER specification cannot be used for deleting queue entries residing in the reader queue.

  Specifying TUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause a deletion of entries destined for local users.

Description of Message-Related Operands

MSG[,ALL|,number]
: Use this set of operands if you want to delete a specific ALLUSERS-type message or all such messages. You can use the PDISPLAY MSG command to have VSE/POWER display the queued messages and their numbers.

  For number, specify the number of the message you want to be deleted by VSE/POWER. If you do not specify a number, or if you explicitly specify ALL, VSE/POWER deletes all queued ALLUSERS-type messages.

Examples of the PDELETE Command

```
PDELETE RDR,ALL
```
Deletes the entire reader queue.

```
PDELETE LST,B
```
Deletes all class B entries in the list queue.

```
PDELETE PUN,CRDATE<02/15/87
```
Deletes all punch queue entries that are older than Feb. 15, 1987.

```
PDELETE MSG
```
Deletes all ALLUSERS-type messages.

```
PDELETE LST,*ASSM
```
Deletes all queue entries in the list queue whose job names begin with the letters ASSM.

```
PDELETE XMT,TNODE=NODEB
```
Deletes entries contained in the transmission queue and destined for node NODEB.

```
L LST,ALL,CFNO=SPC1,CDISP=L
```
Deletes all list queue entries that use the form number SPC1 and whose disposition is L.

PDISPLAY: Display VSE/POWER Status

You can use the command to have VSE/POWER display various kinds of job- and resource-status information. You can also display status information about queue entries on tape or, if VSE/POWER's networking function is used, network-related status information. Examples of and explanations to VSE/POWER produced status reports are given under "VSE/POWER Produced Status Displays" on page 163.

VSE/POWER accepts a PDISPLAY command with up to nine operands.

Format of the Command

This can be one of the following:

First command format:

Operation	Operands
{PDISPLAY|D}	queue[,*abc| ,ALL| ,class| ,FREE| ,HOLD| ,jobname[,jobnumber]| ,LOCAL| ,RJE[,remid]] [,CCLASS=class] [,CDISP=disposition] [,CFNO=formnumber] [,CPRI=priority] [,CRDATE{=|>|<|¬=|>=|<=}date] [,CSYSID=sysid] [,CUSER=userid] [,FNODE=nodeid] [,FULL=YES] [,FUSER=userid] [,TNODE=nodeid] [,TUSER=userid]

Second command format:

Operation	Operands
{PDISPLAY|D}	{*abc| ALL[,listaddr|LST]| FREE| HOLD| LOCAL| RJE[,remid]} [,CCLASS=class] [,CDISP=disposition] [,CFNO=formnumber] [,CPRI=priority] [,CRDATE{=|>|<|¬=|>=|<=}date] [,CSYSID=sysid] [,CUSER=userid] [,FNODE=nodeid] [,FULL=YES] [,FUSER=userid] [,TNODE=nodeid] [,TUSER=userid]

Third command format:

Operation	Operands
{PDISPLAY|D}	{M|MSG|Q|T|TRINFO}

Fourth command format:

Operation	Operands
{PDISPLAY|D}	A[,DEV| ,LOCAL| ,PART[,partition]| ,PNET[,nodeid]| ,RJE| ,SAS]

Fifth command format:

Operation	Operands
{PDISPLAY|D}	PNET[,ALL[,listaddr]|,nodeid|,LINKS]]

Sixth command format:

Operation	Operands
{PDISPLAY|D}	VIO{,listaddr|LST}

Seventh command format:

Operation	Operands
{PDISPLAY|D}	...,TAPE=cuu[,OUT={LST|listaddr|CON}] [,REW={YES|NO}]

The operands for the first and second formats are discussed below. For a discussion of the operands for the remaining formats, turn to one of the following sections:

"Description of Operands for the Third Format" on page 158.
"Description of Operands for the Fourth Format" on page 159.
"Description of Operands for the Fifth Format" on page 159.
"Description of Operands for the Sixth Format" on page 160.
"Description of Operands for the Seventh Format" on page 161.

Description of Operands for the First and Second Formats

queue The operand specifies the VSE/POWER queue for which the command is to be executed. Specify:

LST For the list queue
PUN For the punch queue
RDR For the reader queue
XMT For the transmission queue

If you specify one of the queues without an additional operand, VSE/POWER produces a status report on all jobs in that queue.

If you do not specify a queue, VSE/POWER selects and displays the status of the affected queue entries in all of its queues.

*abc For abc, you can specify the generic part of job names. If you want a status report on jobs whose names have a number of identical characters at the beginning, specify these identical characters for abc. The string you specify for abc may consist of up to eight characters.

ALL[,listaddr|LST]
Specify ALL with a queue to get a status report on all VSE/POWER jobs in the queue or on a certain group of jobs in the queue as defined by one or more additional keyword-type search operands. ALL (without a listaddr specification) is the default if you

specify a queue and none of the group of selectable operands of which ALL is a part.

Specifying the operand without a preceding queue specification causes VSE/POWER to produce a status report on all affected jobs in all of its queues.

Provide, for listaddr, the address of a printer if you want VSE/POWER to produce a printed status report. VSE/POWER then uses the printer for this purpose. Specify the address in the form cuu (channel and unit number). The printer must be available to the VSE/POWER partition, but may not be used for spooling printed output at the same time.

Specify ALL,LST if you want VSE/POWER to spool the queue display as LST queue entry rather than printing it. The LST queue entry has the following attributes:

- $LSTxxxx as job name, where xxxx are the last four digits of the VSE/POWER-assigned job number.
- Priority 9
- Disposition D
- Class A

Specifying ALL,listaddr (or ALL,LST) forces the FULL=YES display function.

class Specifying a class value causes VSE/POWER to produce a status report on all jobs of the specified class in the named queue.

jobname[,jobnumber]
The operand causes VSE/POWER to display the status of the named job. Specify for:

jobname
The name of the VSE/POWER job whose status is to be displayed.

jobnumber
The VSE/POWER assigned job number, if already known. If you do not specify a job number, VSE/POWER displays the status of all jobs queued by the specified job name.

FREE Specifying FREE causes VSE/POWER to produce a status report on all jobs in the specified queue (or in all queues) whose disposition is one of the following:

D = Delete after processing.
K = Keep after processing.
* = In execution

HOLD Specifying HOLD causes VSE/POWER to produce a status report on all jobs in the specified queue (or in all queues) whose disposition is one of the following:

A = Held for appending additional data.
H = Hold in the queue.
L = Leave in the queue.
X = Incomplete because of a system or program failure (if output entry).
If job, held because of previous abnormal termination of VSE/POWER.
Y = Queue entries held because output processing failed.

LOCAL Specifying LOCAL causes VSE/POWER to produce a status report on the following jobs in the specified queue (or in all queues):

For the RDR queue: All jobs.
For the LST/PUN queue: All jobs routed to the central location.
For the XMT queue: All jobs originated by the local system.

RJE[,remid]
Specifying RJE without a remote identifier for 'remid' causes VSE/POWER to produce a status report on all RJE-type jobs in the specified queue (or in all queues).

For remid, specify a remote identifier if you want a status report only on queue entries as follows:

- Entries that were submitted from a specific remote user.
- Entries whose output is to be routed to a specific remote user.

Specify the identifier in the form Rnnn or just nnn (where nnn = the identifier as defined for VSE/POWER table generation).

CCLASS=class
The operand requests a status report on an affected queue entry if the class currently assigned to the entry matches the specified class.

CDISP=disposition
The operand requests a status report on an affected queue entry if the disposition currently assigned to the entry matches the specified disposition.

For disposition, you may specify an asterisk (*) to have VSE/POWER display the status of the VSE/POWER jobs that are being processed.

CFNO=formnumber
The operand requests a status report on an affected queue entry if the form number defined for the output of the entry matches the specified form number.

CPRI=priority
The operand requests a status report on an affected queue entry if the priority currently assigned to the entry matches the specified priority.

CRDATE{=|>|<|¬=|>=|<=}date
In this string:

CRDATE requests a status report for those queue entries whose creation date is one of the following:

Equal to (=)	
Greater than (>)	
Less than (<)	> the specified date
Not equal to (¬=)	
Greater than or equal to (>=)	
Less than or equal to (<=)	

date specifies the date against which the queue entries' creation dates are to be compared. It must have the format defined for the system (mm/dd/yy or dd/mm/yy). Do not specify mm greater than 12, dd greater than 31, and yy greater than the current year.

Note: In a shared spooling environment, all systems must have the same date format.

CSYSID=sysid
The operand requests a status report on an affected queue entry if the processing system defined for a queue entry (in a shared spooling environment) matches the specified processing system.

CUSER=userid
The operand requests a status report on an affected queue entry if the entry's "from user" or "to user" identifier matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

FNODE=nodeid
The operand requests a status report on an affected queue entry if the entry's "from node" name matches the node name given in this operand.

FULL=YES
Use this operand if VSE/POWER is to provide a full (more detailed) status report about a job or a selected group of jobs.

Following is a list of information that VSE/POWER displays in addition if you specify FULL=YES:

Creation date

User information

Job-suffix (output-segment) number

Line count (for LST queue entries)

Original disposition if a job has a temporary disposition

The number of allocated DBLK groups for the queue entry (0 for tape display)

Output format. One of the following:

ASA = Records with American National Standard control characters
BMS = Basic mapping support (used by CICS/VS)
CPDS = Composed page data stream (also referred to as all-point addressable records)
ESC = Escape mode (the format is user defined)
SCS = Standard character string
3270 = 3270 record format (used by CICS/VS)

FUSER=userid

The operand requests a status report on an affected queue entry if the entry's "from user" identifier matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not result in a status report on entries originated by local users.

TNODE=nodeid

The operand requests VSE/POWER to display the status of an affected queue entry if the entry's "to node" name matches the node name given in this operand. This operand is valid only if your specification for queue is XMT.

TUSER=userid

The operand requests a status report on an affected queue entry if the entry's "to user" identifier matches the identifier given in this operand. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

The TUSER specification cannot be used for displaying the status of queue entries residing in the reader queue.

Specifying TUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not result in a status report on entries to be handled at and destined for the central location.

Description of Operands for the Third Format

```
| {PDISPLAY|D}     {M|MSG|Q|T|TRINFO}                    |
```

Use this set of operands to get a display of VSE/POWER related status information. Specify:

M To cause a redisplay of all VSE/POWER messages for which an operator response is still required.

MSG To cause a display of all ALLUSERS-type messages and to identify the users who submitted them. You can use the number displayed with each message to delete a specific message.

Q To cause a display of:

- The number of free queue records and the percentage of the queue file currently in use.
- The number of free DBLK groups and the percentage of the data file currently in use.
- The DBLK and DBLK group size.
- The percentage to which the account file is filled (if accounting is supported).

T To cause a display of:

- The current time and date.
- The number of storage pages reserved by PFIX macros.
- The current number of VSE/POWER tasks.
- The system-ID if a valid one was specified during table generation for VSE/POWER.
- The node identifier of the local node if networking is used.

TRINFO To cause a display of the following trace area information:

- Begin address.
- End address.
- Current position.

Description of Operands for the Fourth Format

```
{PDISPLAY|D}     A[,DEV|
                   ,LOCAL|
                   ,PART[,partition]|
                   ,PNET[,nodeid]|
                   ,RJE|
                   ,SAS]
```

Use this command format to get a display of the status of active tasks.

A Code this operand as shown.

Use this operand alone to get a list of all active tasks. You can restrict this type of display by specifying one of the optional operands described below.

DEV The operand limits the display to active device-service tasks.

LOCAL The operand limits the display to active reader and writer tasks.

PART[,partition]
The operand limits the display to the active tasks associated with active partitions if you specify PART alone. You may further limit the display by specifying a partition's identifier (in the form BG or Fn).

PNET[,nodeid]
The operand limits the display to all active network tasks if you specify PNET alone. You may further limit the display to a certain node by specifying, for nodeid, the name of that node.

RJE The operand limits the display to all active RJE tasks.

SAS The operand limits the display to all active spool-access support tasks.

Description of Operands for the Fifth Format

```
{PDISPLAY|D}     PNET[,ALL[,listaddr]|,nodeid|,LINKS]]
```

Use this command format to get a display of information from the currently used network definition table (NDT).

PNET Code this operand as shown.

If you specify PNET alone, VSE/POWER displays networking infor-

mation about your own node, the VSE system you are using. You may in addition, specify one of the (optional) operands described below.

ALL[,listaddr]
Specifying ALL causes VSE/POWER to display all NDT entries contained in the table.

Specify, for listaddr, the address of a printer if you want a printed status report. VSE/POWER then uses the printer for this purpose. Specify the address in the form cuu (channel and unit number). The printer must be available to the VSE/POWER partition, but may not be used for spooling printed output at the same time.

nodeid Specify, for nodeid, the name of the node whose NDT entry you want to be displayed by VSE/POWER.

LINKS Specify LINKS to have VSE/POWER display the names of the nodes with which your own node has a direct link or a session defined.

Description of Operands for the Sixth Format

```
{PDISPLAY|D}     VIO,{listaddr|LST}
```

Use this command format to get either a printout or to produce a LST queue entry of the VIO (virtual-storage) copy of the queue file.

VIO Code this operand as shown.

listaddr
Specify the address (channel and unit number) of the printer to be used.

In a shared spooling environment, use the D VIO,listaddr command with care, since another sharing system that needs queue file write access has to wait until processing of the command is completed.

LST Specify this operand to create a LST queue entry with:

- $VIOxxxx as job name, where xxxx are the last four digits of the VSE/POWER-assigned job number
- Priority 9
- Disposition D
- Class A

The layout of the printed queue file is similar to the dump format. Each queue record is identified by its relative queue record number, each queue record block by its relative queue record block number.

Description of Operands for the Seventh Format

```
{PDISPLAY|D}     ...,TAPE=cuu[,OUT={LST|listaddr|CON}]
                 [,REW={YES|NO}]
```

Use this command format to get a display of queue entries residing on either a POFFLOAD tape or a spool tape. To indicate which queue entry or entries you want to have displayed, specify (for ...) any of the operands listed under "Description of Operands for the First and Second Formats" on page 153.

TAPE=cuu
: Specify the address (channel and unit number) of the tape to be used.

OUT={LST|listaddr|CON}
: Specify where the information about the queue entries on the tape should be displayed.

  Specifying OUT= overrides a previous specification of ALL,LST or ALL,listaddr.

  LST
  : Specifies that a LST queue entry with the following characteristics is to be built:

    - $TAPnnnn as job name, where nnnn are the last four digits of the VSE/POWER-assigned job number
    - Priority 9
    - Disposition D
    - Class A

    The LST queue entry contains the following information for each queue entry on the tape:

    - Job name, number, and suffix, if applicable
    - Class
    - Queue identifier (part of header line)
    - Disposition
    - Priority
    - Creation date
    - Number of pages for a LST queue entry, cards for a PUN queue entry, or records for a RDR queue entry
    - Originator node and user-id
    - Destination node and user-id

  listaddr
  : Indicates that the information is to be printed on the specified printer. Specify the address in the form cuu (channel and unit number).

CON Specifies that the information about the queue entries is to be displayed at the system console. The information shown is the same as in an operator console display with message 1R46I.

REW={YES|NO}
YES indicates that the tape is to be rewound before, and to be unloaded after processing of a PDISPLAY command. NO indicates that the tape is not to be rewound before, and not to be unloaded after processing of a PDISPLAY command.

Examples of the PDISPLAY Command

PDISPLAY RDR,*ASSM
Displays the status of all VSE/POWER jobs in the reader queue whose names begin with ASSM.

PDISPLAY RDR,ASSEM
Displays the status of VSE/POWER job ASSEM in the reader queue.

PDISPLAY ALL,00E
Prints a status report on all VSE/POWER jobs in all queues. Specifying the address of a printer forces FULL=YES; the status report therefore includes "full" status information.

PDISPLAY RDR,CDISP=*
Displays the status of all VSE/POWER jobs that are being processed.

PDISPLAY ALL,TAPE=280,OUT=CON
Displays, on the system console, all queue entries that are saved on the tape with address 280.

D A
Displays all active tasks.

D M
Displays all messages for which an operator response is still pending.

D MSG
Displays all ALLUSERS messages.

D LST,CFNO=SPEC,CDISP=D
Displays all list queue entries that have the form number (form identifier) SPEC and the disposition D.

D PNET,LINKS
Displays the names of all nodes with which the own node has a direct link or session defined (the display includes the names of nodes for which the link or session is currently inactive).

VSE/POWER Produced Status Displays

VSE/POWER produces status displays in the form of messages. It writes these messages to one of the following:

- The device assigned to SYSLOG (by default), or
- A line printer, if listaddr is specified, or
- A list queue entry, which you can create by specifying LST.

A PDISPLAY request (PDISPLAY RDR,ALL for example) may cause many status lines to be written to the console in rapid succession. You, the operator, may have problems to read them all and to find the status information you are looking for. To avoid this problem, limit the number of displayed status lines by specifying also a class, with or without a priority. Example:

```
PDISPLAY RDR,C,CPRI=3.
```

In the status-display examples that follow, the reference numbers in front of the display lines refer to explanations below the examples. After having studied the examples, you should be able to read any status display produced by VSE/POWER in response to a PDISPLAY command.

The command is: PDISPLAY LST

Status-display example:

```
  (1)   1R46I    LIST QUEUE   P D C S  PAGES  CC FORM
        1R46I  PWR001   00162 3 D B 1      3  *8 FFFF  FROM=(SYSA)
               FLASH=FLSH,BURST,COPIES= 36
        1R46I  PWR002   00170 4 H C 2    518   1 ABCD  TO=(POST)
        1R46I  PAYROLL  00172 8 K P 2   1815  *2 PAYF  TO=(WICA)
               FLASH=PAYR,COPIES=  4
        1R46I  PWR003   00174 8 * Z 2    211           TO=(MAIN)
           ...  ...  ...
```

(1) 1R46I LIST QUEUE P D C S PAGES CC FORM
The header line for the display. On the line:

P: The priority column.

D: The disposition column. This can be one of the following:

- * - Being processed.
- A - Appendable. Spool data may be added to the queue entry via spool-access support.
- D - Dispatchable. Delete after processing.
- H - Hold in the queue until released.
- K - Keep after processing.
- L - Leave in the queue.
- X - Incomplete because of a system or program failure (if output entry).
  If job, held because of previous abnormal termination of VSE/POWER.
- Y - Queue entry held because output processing failed.

For further information on dispositions refer to
Appendix A on page 309.

C: The class column.

S: The system-identifier column (applies only if shared spooling is used). VSE/POWER displays the system on which the queue entry is to be or is being processed.

PAGES: The "number of pages" column. VSE/POWER displays the number of pages for list output; it displays the number of card-image records for punched output.

Status-display example - PDISPLAY LST - cont'd:

```
(1)   1R46I    LIST QUEUE   P D C S  PAGES  CC FORM
      1R46I  PWR001   00162 3 D B 1      3  *8 FFFF  FROM=(SYSA)
             FLASH=FLSH,BURST,COPIES= 36
      1R46I  PWR002   00170 4 H C 2    518   1 ABCD  TO=(POST)
      1R46I  PAYROLL  00172 8 K P 2   1815  *2 PAYF  TO=(WICA)
             FLASH=PAYR,COPIES=  4
      1R46I  PWR003   00174 8 * Z 2    211           TO=(MAIN)
       ...   ...  ...
```

(1) Continued:

For list output, VSE/POWER displays the number of pages left to be written to the selected output device. If this number is greater than 65,535, then VSE/POWER displays the actual number minus the applicable multiple of 65,535.

VSE/POWER increments its page count by 1 for every occurrence of a "skip to channel 1" command. If there is no such command, this count is zero. VSE/POWER increments its page count also for a page overflow. VSE/POWER considers each record as a page if the queue entry contains data of one of the following formats:

- Basic mapping format (BMS)
- Composed page data stream (CPDS)
- 3270 mapping

CC: The copy-count column. VSE/POWER displays the number of copies left to be produced, including the one that is being processed. This is also true if the command was issued for a queue entry which was canceled (by PFLUSH) or stopped (by PSTOP ...,RESTART).

If copy grouping (COPYG) is specified for output to an IBM 3800, then VSE/POWER indicates this by an * preceding the number of transmissions.

FORM: The form-number column.

Status-display example - PDISPLAY LST - cont'd:

```
          1R46I    LIST QUEUE   P D C S  PAGES  CC FORM
(2)       1R46I  PWR001   00162 3 D B 1      3  *8 FFFF  FROM=(SYSA)
(3)              FLASH=FLSH,BURST,COPIES= 36
          1R46I  PWR002   00170 4 H C 2    518   1 ABCD  TO=(POST)
          1R46I  PAYROLL  00172 8 K P 2   1815  *2 PAYF  TO=(WICA)
                 FLASH=PAYR,COPIES=  4
          1R46I  PWR003   00174 8 * Z 2    211           TO=(MAIN)
           ...   ...  ...
```

(2) 1R46I PWR001 00162 3 D B 1 3 *8 FFFF FROM=(SYSA)
The first display line for the first job in the queue. On the line:

PWR001: The name by which VSE/POWER knows this job.

00162: The VSE/POWER assigned job number. After a warm start (without any reformatting of the queue files), the assigned job number may be one higher than expected.

3 D B 1: Explained by P D C S - See the explanations to the preceding status-display line (1).

3: Number of pages.

*8: When copy grouping (COPYG) is specified, the number of transmissions of the output to the involved IBM 3800 printer.

FFFF: The identification of the output form.

FROM=(SYSA): The originator's user identifier.

(3) FLASH=FLSH,BURST,COPIES= 36
The second display line for the first job in the queue. VSE/POWER prints a second line (for a PDISPLAY LST request) only if additional LST operands are to be displayed. On this line:

FLASH=FLSH: The name, FLSH in the example, of the forms overlay.

BURST: The paper must be threaded through the burster-trimmer stacker before printing for the job can start.

COPIES= 36: The total of the specified copy groups (in COPYG=n,n... of * $$ LST, for example).

Status-display example - PDISPLAY LST - cont'd:

```
          1R46I    LIST QUEUE   P D C S  PAGES  CC FORM
          1R46I  PWR001   00162 3 D B 1      3  *8 FFFF  FROM=(SYSA)
                 FLASH=FLSH,BURST,COPIES= 36
    (4)   1R46I  PWR002   00170 4 H C 2    518   1 ABCD  TO=(POST)
    (5)   1R46I  PAYROLL  00172 8 K P 2   1815  *2 PAYF  TO=(WICA)
    (6)          FLASH=PAYR,COPIES=  4
    (7)   1R46I  PWR003   00174 8 * Z 2    211           TO=(MAIN)
            ...  ...  ...
```

(4) 1R46I PWR002 00170 4 H C 2 518 1 ABCD TO=(POST)
The first (and only) report line for the second job in the queue. The explanations given for the status-display line (2) apply accordingly. This output goes to a printer other than an IBM 3800.
The line includes the identifier (POST) of the user to which the output is to be routed.

(5) 1R46I PAYROLL 00172 8 K P 2 1815 *2 PAYF TO=(WICA)
The first display line for the third job in the queue. The explanations given for the status-display line (2) apply accordingly.

(6) FLASH=PAYR,COPIES= 4
The explanations given for the status-display line (3) apply accordingly. There is no need for the paper to be threaded through the printer's burster-trimmer stacker.

(7) 1R46I PWR003 00174 8 * Z 2 211 TO=(MAIN)
The first (and only) report line for the fourth job in the queue. The output is currently printed on a printer started for class Z and attached to the system with sysid 2.

Note: Only the system with the active list task (sysid=2) shows the actual number of pages left to be printed. The other sharing systems only show that the queue entry is currently printed and the total number of pages the queue entry comprises.

The command is: PDISPLAY A

Status display example:

```
  (1)  1R48I  LST,00E,T,2  POWERM12 00008 A
       1R48I  LST,00F,V,2,VM  VMOUTP12 00016 V
       1R48I   F2,FEC,A,       INACTIVE,
       1R48I  RDR,00C,B    GABLE,00017,C   86 RECORDS SPOOLED
       1R48I  RDR,01C,A,       INACTIVE,
       1R48I  DEV,PRT1,DDS=DDSIM,ABCD, PRI01,00150,A
       1R48I  SAS,00005, SAS=SYSCICS1,TOOL1, REQ=PUT, REPORT,04711,A
              100 RECORDS SPOOLED
       1R48I  C-RV,036,     AWAITING     NODE=BOEVS03
       1R48I  O-TR1,036,*,  WSKTNL02,00167,L  LEFT=004919 OF 010018
              NODE=BOEVS03
       1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B   0 LINES SPOOLED
       1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
           ...  ...  ...
```

(1) 1R48I LST,00E,T,2 POWERM12 00008 A
On the display line:

LST: The type of the involved task (list writer in the example). Besides LST, one of the following may show up:

```
  BG,Fx = Execution tasks
    DEV = Device-service
   C-RV = Console receiver
   C-TR = Console transmitter
    GSP = GETSPOOL service
  J-RVn = Job receiver
  J-TRn = Job transmitter
  L-OFF = POFFLOAD SAVE|BACKUP
  O-RVn = Output receiver
  O-TRn = Output transmitter
    PSP = PUTSPOOL service
    PUN = Punch writer
    RDR = Reader
  R-OFF = POFFLOAD LOAD|SELECT
    SAS = Spool-access support
```

where:
n = number of the transmitter or receiver task

00E: The address of the unit record device associated with this task.

T: The class (or classes - up to four) that can be processed by the task. The order of the classes is the order of processing. If VSE/POWER displays an *, then all classes are processed by the task.

Status display example - PDISPLAY A - cont'd:

```
 (1)  1R48I  LST,00E,T,2  POWERM12 00008 A
 (2)  1R48I  LST,00F,V,2,VM  VMOUTP12 00016 V
 (3)  1R48I   F2,FEC,A,       INACTIVE,
      1R48I  RDR,00C,B    GABLE,00017,C   86 RECORDS SPOOLED
      1R48I  RDR,01C,A,       INACTIVE,
      1R48I  DEV,PRT1,DDS=DDSIM,ABCD, PRI01,00150,A
      1R48I  SAS,00005, SAS=SYSCICS1,TOOL1, REQ=PUT, REPORT,04711,A
             100 RECORDS SPOOLED
      1R48I  C-RV,036,      AWAITING     NODE=BOEVS03
      1R48I  O-TR1,036,*,  WSKTNL02,00167,L  LEFT=004919 OF 010018
             NODE=BOEVS03
      1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B    0 LINES SPOOLED
      1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
           ...  ...  ...
```

(1) Continued:

2: The number of buffers being used. This can be one of the following:

1 = One buffer
2 = Two buffers
D = Two input and two output buffers

POWERM12: The name by which the involved job is known by VSE/POWER. The display is ******** if VSE/POWER does not yet know this name.

00008: The VSE/POWER-assigned job number.

A: The job class assigned to the job.

(2) 1R48I LST,00F,V,2,VM VMOUTP12 00016 V

The same as (1) above, except for:

VM: This indicates that the output of the job VMOUTP12 is directed to a printer under control of VM/SP.

(3) 1R48I F2,FEC,A, INACTIVE,

On the display line:

F2: The execution task servicing the indicated partition F2 in this example). Any of the possible partition indicators could show up.

FEC: The address of the unit record device associated with the task.

A: The class (or classes - up to four) that can be processed by the task (when it is active).

INACTIVE: The associated task is not active.

Status display example - PDISPLAY A - cont'd:

```
      1R48I  LST,00E,T,2  POWERM12 00008 A
      1R48I  LST,00F,V,2,VM  VMOUTP12 00016 V
      1R48I   F2,FEC,A,       INACTIVE,
 (4)  1R48I  RDR,00C,B    GABLE,00017,C   86 RECORDS SPOOLED
 (5)  1R48I  RDR,01C,A,       INACTIVE,
 (6)  1R48I  DEV,PRT1,DDS=DDSIM,ABCD, PRI01,00150,A
 (7)  1R48I  SAS,00005, SAS=SYSCICS1,TOOL1, REQ=PUT, REPORT,04711,A
             100 RECORDS SPOOLED
      1R48I  C-RV,036,      AWAITING     NODE=BOEVS03
      1R48I  O-TR1,036,*,  WSKTNL02,00167,L  LEFT=004919 OF 010018
             NODE=BOEVS03
      1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B    0 LINES SPOOLED
      1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
           ...  ...  ...
```

---

(4) 1R48I RDR,00C,B GABLE,00017,C 86 RECORDS SPOOLED
If an execution (or spool-access support) task is involved, VSE/POWER displays the number of records (lines or cards, whichever applies) so far spooled.

(5) 1R48I RDR,01C,A, INACTIVE,
The same as for line (3), except that a RDR task is involved.

(6) 1R48I DEV,PRT1,DDS=DDSIM,ABCD, PRI01,00150,A

If a device-service task (DEV) is involved, VSE/POWER displays:
- The printer device name (PRT1)
- The XPCC application ID of the owning device-driving system (DDSIM)
- The output classes that the printer is started for (A,B,C,D)
- Job name (PRI01), job number (150), and job class (A) of the accessed output entry

(7) 1R48I SAS,00005, SAS=SYSCICS1,TOOL1, REQ=PUT, REPORT,04711,A
100 RECORDS SPOOLED

If a spool-access support user task (SAS) is involved, VSE/POWER displays:
- The connection identifier assigned by VSE/POWER (00005)
- The XPCC application ID (SYSCICS1)
- The userid (TOOL1)
- The function being performed (PUT)
- Job name (REPORT), job number (04711), and job class (A), if present
- The number of records spooled so far (100)

Status display example - PDISPLAY A - cont'd:

```
          1R48I  LST,00E,T,2  POWERM12 00008 A
          1R48I  LST,00F,V,2,VM  VMOUTP12 00016 V
          1R48I   F2,FEC,A,       INACTIVE,
          1R48I  RDR,00C,B    GABLE,00017,C   86 RECORDS SPOOLED
          1R48I  RDR,01C,A,       INACTIVE,
          1R48I  DEV,PRT1,DDS=DDSIM,ABCD, PRIO1,00150,A
          1R48I  SAS,00005, SAS=SYSCICS1,TOOL1, REQ=PUT, REPORT,04711,A
                 100 RECORDS SPOOLED
   (8)    1R48I  C-RV,036,      AWAITING     NODE=BOEVS03
   (9)    1R48I  O-TR1,036,*,  WSKTNL02,00167,L  LEFT=004919 OF 010018
                 NODE=BOEVS03
  (10)    1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B       0 LINES SPOOLED
  (11)    1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
            ...  ...  ...
```

(8) 1R48I C-RV,036, AWAITING NODE=BOEVS03

If a networking task is involved, VSE/POWER displays:
- The type of the involved task (C-RV = console receiver)
- The address of the BSC line or CTCA address used for the network communication (036)
- The status of the task (AWAITING)
- The nodeid of the node with which communication is established (BOEVS03)

(9) 1R48I O-TR1,036,*, WSKTNL02,00167,L LEFT=004919 OF 010018
NODE=BOEVS03

If a networking task is involved, VSE/POWER displays:
- The type of the involved task (O-TR1 = output transmitter 1)
- The address of the BSC line or CTCA address used for the network communication (036)
- The * indicates that the transmitter task may transmit entries of all possible classes
- The VSE/POWER-assigned job name (WSKTNL02), job number (00167), and class (L) of the queue entry being transmitted
- The number of records that remain to be transmitted (4919) out of the total number of records to be transmitted (10018)
- The nodeid of the node with which communication is established (BOEVS03)

(10) 1R48I BG,FFF,TAPE=182,, PRINTLOG,00265,B 0 LINES SPOOLED
Display of a spool task for spooling job output to tape (DISP=T).

(11) 1R48I LST,02E,TAPE=183,, DFHFCTSP,00102,Q
Display of a list task (from spool tape to printer) started by PSTART LST,02E,X'183'.

The command is: PDISPLAY LST,ALL,FULL=YES

Status display example:

```
(1) 1R46I  LIST QUEUE     P D C S   PAGES  CC FORM
(2) 1R46I  PAYROLL  00172 8 K P       815   2 F001 FROM=(HBAK)
(3)        D=07/12/84,U='MONTHLY REPORT  ',L=004620,DBGP=322
    1R46I  COMPILE1 00175 5 D P       100   1      TO=(ICHA)
           D=07/12/84,U='SUPERV. ASSEMBLY',S=001,L=006056,DBGP=475
    1R46I  COMPILE1 00175 5 D P        83   1      TO=(ICHA)
           D=07/12/84,U='SUPERV. ASSEMBLY',S=002,L=002789,DBGP=273
    1R46I  DIRLST   04711 9 D X'       35   1      TO=(JDAR)
           D=07/12/84,L=000100,RF=BMS,DBGP=026
       ...  ...  ...
```

VSE/POWER provides all of the displayable information as shown above for all queue entries:
- During VSE/POWER start-up if SYSLST is assigned.
- After a PEND cuu command.

---

(1) 1R46I LIST QUEUE P D C S PAGES CC FORM
The status-display header line. For an explanation, see (1) of the first example in this section.

If a job's disposition is shown as A (appendable), X (incomplete), or Y (held because of output processing failure), VSE/POWER displays the original disposition. Example: ORGDP=D

(2) 1R46I PAYROLL 00172 8 K P 815 2 F001 FROM=(HBAK)
The first display line for the first job in the queue. The line includes the originator's user identifier (HBAK) in the example). If a job comes from another (not the own) node, VSE/POWER displays also the name of the originating node. In that case, the format of the display is:

FROM=nodeid(userid).

(3) D=07/12/84,U='MONTHLY REPORT ',L=004620,DBGP=322
A display line provided if FULL=YES is specified. On the line:

D=07/12/84: The job-creation date.

U='MONTHLY REPORT': The user information supplied by the originator.

L=004620: The number of lines that make up the job; 4620 in this example. (A write-without-line-spacing command counts as line.)

DBGP=322: The number of allocated DBLK groups on disk for the queue entry; 322 in this example (0 for tape display).

Status display example - PDISPLAY LST ALL,FULL=YES cont'd:

```
      1R46I  LIST QUEUE     P D C S   PAGES  CC FORM
      1R46I  PAYROLL  00172 8 K P       815   2 F001 FROM=(HBAK)
             D=07/12/84,U='MONTHLY REPORT  ',L=004620,DBGP=322
(4)   1R46I  COMPILE1 00175 5 D P       100   1      TO=(ICHA)
(5)          D=07/12/84,U='SUPERV. ASSEMBLY',S=001,L=006056,DBGP=475
(6)   1R46I  COMPILE1 00175 5 D P        83   1      TO=(ICHA)
(7)          D=07/12/84,U='SUPERV. ASSEMBLY',S=002,L=002789,DBGP=273
(8)   1R46I  DIRLST   04711 9 D X        35   1      TO=(JDAR)
(9)          D=07/12/84,L=000100,RF=BMS,DBGP=026
        ...  ...  ...
```

---

(4) 1R46I COMPILE1 00175 5 D P 100 1 TO=(ICHA)
The first display line for the second job in the queue. The line includes the identifier (ICHA in the example) of the user to which the output is to be routed.

If an output is to be routed to a user at another (not the own) node, VSE/POWER displays also the name of this other node. In that case, the format of the display is:

TO=nodeid(userid).

(5) D=07/12/83,U='SUPERV ASSEMBLY',S=001,LINES=006056,DBGP=475
This is similar to line (3) of the example, except that the line includes a job suffix (or output-segment number), S=001.

(6) 1R46I COMPILE1 00175 5 D P 83 1 TO=(ICHA)
This is similar to line (4) of the example.

(7) D=07/12/83,U='SUPERV ASSEMBLY',S=002,LINES=002789,DBGP=273
This is similar to lines (3) and (5) of the example.

(8) 1R46I DIRLST 04711 9 D X 35 1 TO=(JDAR)
This is similar to line (4) of the example.

(9) D=07/12/84,L=000100,RF=BMS,DBGP=026
No user information was supplied. On the line:

RF=BMS: The record format is other than unit record and, therefore, indicated by VSE/POWER. BMS = basic mapping support.

DBGP=026: The number of allocated DBLK groups on disk for the queue entry; 26 in this example.

The command is: PDISPLAY PNET,ALL

Status display example:

```
 (1) 1RB7I  ******  NDT NAME = NDTTPMM    ******
 (2) 1RB7I  NODE     ROUTE1   ROUTE2   AUTH  BSIZE APPLID  PASSWORD
     1RB7I  BOEVSE05 ----- LOCAL -----              TPNS
     1RB7I  BOEVS03  *SNA *   BOEVS03  NET     500 ZURS
     1RB7I  BOEVS05  *LINK*            JOB    1500
     1RB7I  HURSL01  BOEVS03           JOB     400
```

---

(1) 1RB7I ****** NDT NAME = NDTTPMM ******
A status-display header line. It gives the name, NDTTPMM, by which your active network-definition table is cataloged.

(2) 1RB7I NODE ROUTE1 ROUTE2 AUTH BSIZE APPLID PASSWORD
A status-display header line providing the headings for the display columns:

NODE — The name of the node to which the display line applies.

ROUTE1 — The name of a node with which a direct link exists. This link is used for transmission of data to the node named in the NODE column as long as the link is active.

ROUTE2 — The name of a node with which a direct link exists. This link is used for transmission of data to the node named in the NODE column if either:
- The link to the node named under ROUTE1 is not active, or
- No ROUTE1 node is defined because a direct link exists, but this direct link is not active.

AUTH — The level of command authorization (JOB, NOJOB, or NET) for the operator of the node named in the NODE column. For more information about command authorization, see the publication VSE/POWER Networking User's Guide.

BSIZE — The length (in bytes) of the transmission buffers. Note that this is the buffer size of the NDT, not necessarily the actually used buffer size.

APPLID On the display line for your own (local) node, the name by which your VSE/POWER is known to ACF/VTAM; else the name of the application with which your VSE/POWER communicates.

PASSWORD The node password as defined in the PNODE macro for the node named in the NODE column. VSE/POWER shows this password only if the PDISPLAY command was submitted at the system's console.

PDRAIN: Deactivate (Drain) a Transmitter or Receiver Task

This command applies only if the networking function is initialized; it deactivates a transmitter or receiver task.

A transmitter task is deactivated automatically if the receiver task (at another node) with which communication is to take place is not active.

A deactivated transmitter or receiver task remains deactivated until it is reactivated by a PACT command.

To stop further handling of a queue entry by a transmitter or receiver task, use the PFLUSH command.

Format of the Command

Operation	Operands
{PDRAIN|N}	PNET,nodeid,{TRn|RVn},{JOB|OUT}[,EOJ]

Description of Operands

PNET Code this operand as shown.

nodeid For nodeid, specify the name of the node with which the affected receiver or transmitter task is communicating.

TRn The operand specifies that the transmitter task n is to be deactivated.

For n, specify the number of the affected transmitter task if only one such task is to be deactivated; specify an asterisk (*) if all transmitter tasks of the specified type (JOB or OUT) are to be deactivated.

RVn The operand specifies that the receiver task n is to be deactivated.

For n, specify the number of the affected receiver task if only one such task is to be deactivated; specify an asterisk (*) if all receiver tasks of the specified type (JOB or OUT) are to be deactivated.

A PDRAIN command for a receiver task must be given if, for example, the spool files become full and you do not want additional data to be received from another node.

JOB Specify this operand if the affected transmitter or receiver task is used for transmitting or receiving jobs.

OUT Specify this operand if the affected transmitter or receiver task is used for transmitting or receiving output (list or punch).

EOJ Specify EOJ if the receiver or transmitter task should not stop until the complete queue entry has been received or transmitted.

If you do not specify EOJ, then:

- For a transmitter task, the transmission is stopped and the job or output being transmitted is re-queued for later transmission.
- For a receiver task, the job or output being received is purged at the own node and re-queued on the transmitting system with disposition H or L.

Example of the PDRAIN Command

```
PDRAIN PNET,NODEA,TR2,OUT,EOJ
```

The output transmitter task 2 continues transmitting data for the currently transmitted queue entry until end of job. VSE/POWER then deactivates the task.

PEND: End Operation Under Control of VSE/POWER

The command ends VSE/POWER operation; it causes all partitions under control of VSE/POWER to be released and the VSE/POWER partition to be restored to normal VSE operation.

All unit-record devices assigned in the VSE/POWER partition and in the partitions controlled by VSE/POWER are unassigned with one exception: if SYSIN was assigned to a disk device during warm start of VSE/POWER, the SYSIN device is not unassigned.

Unless FORCE (formerly KILL) is specified, all active VSE/POWER tasks are allowed to continue until they have come to the end of the currently processed queue entry. This means: until the end of execution if the queue entry is in the reader queue; until the end of output processing (printing or punching out) if the queue entry is in an output queue. RJE and network processing is completed in a similar way.

Format of the Command

Operation	Operands
PEND	[uraddr|FORCE]

Description of Operands

If you issue the command without an operand, VSE/POWER performs a normal shut-down operation without a status report.

uraddr The operand indicates that you want VSE/POWER to provide a status report giving (a) statistical information about the ended session and (b) the contents of all queues. For an example of the statistics part of a status report, refer to Figure 12 on page 121. Examples of queue displays are shown under "VSE/POWER Produced Status Displays" on page 163.

For uraddr, specify the address (in the form cuu) of the printer on which the report is to be printed. The printer remains assigned to the partition after VSE/POWER has terminated.

FORCE Specify FORCE if VSE/POWER operation is to end at once. VSE/POWER prompts the operator for an instruction whether or not a dump is to be provided.

All VSE/POWER-supported partitions are canceled, but currently active reader and writer tasks are terminated as for a PSTOP command.

Examples of the PEND Command

PEND
Normal end of VSE/POWER operation without a status report.

PEND 00E
Normal end of VSE/POWER operation with a status report.

PEND FORCE
Immediate end of VSE/POWER operation without a status report.

PFLUSH: Cancel a Queue Entry

The command causes the processing of a VSE/POWER queue entry by a VSE/POWER task to be canceled. This task then continues processing for the next available VSE/POWER queue entry.

When a job input is canceled, further processing for the affected job ends, and VSE/POWER does not build a queue entry for the job.

When job processing is canceled, further processing of the affected job stops. If this job's disposition is:

D VSE/POWER deletes the entry, except if HOLD is specified, in which case the entry is retained with a disposition of H.

K VSE/POWER retains the entry with its disposition changed to L.

When a job output is canceled, further output processing for this output stops. If this output's disposition is:

D VSE/POWER deletes the entry, except if HOLD is specified, in which case the entry is retained with a disposition of H.

K VSE/POWER retains the entry with its disposition changed to L.

Note: If a PSTOP command with EOJ was issued for the same writer task, PFLUSH causes this task to be stopped immediately. The job's output queue entry is retained.

You can start a canceled (but retained) output again by issuing either or both of the commands PALTER and PRELEASE. If the canceled output called for two or more output copies, VSE/POWER starts writing to the named output device from the beginning of the copy whose output was stopped.

Format of the Command

The command can be used with distinct sets of operands as shown below:

Operation	Operands
{PFLUSH|F}	{uraddr[,HOLD]|tapeaddr}
	partition[,HOLD]
	DEV,devname[,HOLD][,PARM='string']
	PNET,nodeid,{RVn|TRn},{JOB|OUT}[,HOLD]

Description of Operands

```
{PFLUSH|F}      {uraddr[,HOLD]|tapeaddr}
```

Use this set of operands if VSE/POWER started processing a job or an output which you want to be processed sometime later - for example when a certain resource becomes available.

uraddr For uraddr, specify the channel and unit address of the unit-record device associated with the reader or writer task that is to be terminated.

HOLD Specify HOLD if the command affects a writer task and you want VSE/POWER to retain the affected queue entry (see also the general discussion of the command preceding the format description).

tapeaddr
For tapeaddr, specify the channel and unit address of the tape drive that is being used as a SYSIN-reader or as a POFFLOAD device.

If you issue a PFLUSH to stop a SYSIN-reader or POFFLOAD LOAD or SELECT operation, then the currently processed queue entry is terminated. The associated task continues processing the next available queue entry.

If the PFLUSH command ends a POFFLOAD BACKUP or SAVE operation, the affected task stops at the end of the currently processed queue entry, and the job or output entry is returned to its original disposition (the same as for the PSTOP command).

```
{PFLUSH|F}      partition[,HOLD]
```

Use this set of operands to cancel further processing of a job in a VSE/POWER-controlled partition.

partition
For partition, specify the identifier (BG or Fn) of the partition that is to be canceled. This causes the VSE job running in the partition to be canceled without a dump, regardless of any dump options that may have been specified.

If an IBM subsystem (such as CICS/VS or VSE/ICCF) is running in this partition, VSE/POWER requests the operator to confirm the PFLUSH command.

HOLD Specify HOLD if VSE/POWER is to retain the affected queue entry in its queue (see also the general discussion of the command preceding the format description).

```
{PFLUSH|F}        DEV,devname[,HOLD][,PARM='string']
```

Use this set of operands to indicate to a subsystem or program in another partition to cancel a queue entry that is being processed for output on a device owned by that subsystem (program).

DEV Code this operand as shown.

devname For devname, specify the name of the output device to be used. This name must be the one by which the device-owning subsystem or application program knows the device.

HOLD Specify HOLD if VSE/POWER is to retain the affected queue entry in its queue (see also the general discussion of the command preceding the format description).

PARM='string'
For string in PARM='string', supply a device and subsystem specific parameter string, if this is required. VSE/POWER's requirements regarding the parameter string are:

- It may not be longer than 60 characters. This includes blanks and commas that the program may need as delimiters.
- An apostrophe (') within the string must be entered as two successive apostrophes ('').

```
{PFLUSH|F}        PNET,nodeid,{RVn|TRn},{JOB|OUT}[,HOLD]
```

Use this set of operands if a queue entry processed by a transmitter or receiver task is to be terminated. The task stops processing of the affected queue entry immediately and continues processing with the next available queue entry.

Note: Use the PDRAIN command if no further queue entries are to be transmitted or received.

PNET Code this operand as shown.

nodeid For nodeid, specify the name of the associated node:

- The node to which the queue entry you want to be terminated is being transmitted (your command refers to a transmitter task).

- The node from which the queue entry you want to be terminated is being received (your command refers to a receiver task).

RVn|TRn For n in RVn, specify the number of the receiver task that is to stop processing for the affected queue entry. VSE/POWER sends a 'receiver cancel' notification to the originating node (to initiate re-queueing of the queue entry at that node).

For n in TRn, specify the number of the transmitter task that is to stop processing for the affected queue entry.

JOB|OUT Specify JOB if the named transmitter or receiver task is used for transmitting or receiving jobs.

Specify OUT if the named transmitter or receiver task is used for transmitting or receiving output.

HOLD Specify HOLD with TRn (it is meaningless with RVn) if VSE/POWER is to retain the terminated queue entry in its queue (see also the general discussion of the command preceding the format description).

Examples of the PFLUSH Command

PFLUSH 00E
Terminates the processing of the list queue entry that is being printed. VSE/POWER deletes the entry or retains it with disposition L, depending on the entry's original disposition. VSE/POWER continues with the next available entry.

PFLUSH BG
Terminates the VSE job that is being processed in the background partition. VSE/POWER deletes the entry or retains it with disposition L, depending on the entry's original disposition. VSE/POWER continues with the next available queue entry.

PFLUSH BG,HOLD
Terminates the VSE job that is being processed in the background partition. VSE/POWER retains the entry with disposition H or L, whichever applies, and continues with the next available queue entry.

PFLUSH PNET,NODEB,TR1,JOB,HOLD
Terminates the transmission of a reader queue entry under control of the transmitter task TR1. VSE/POWER retains the entry with disposition H or L, whichever applies, and continues with the next available queue entry.

PGO: Reactivate a VSE/POWER Task

The command reactivates a VSE/POWER task that has waited for an operator response. You would normally use this command after having responded to an action-type message. You cannot use PGO to restart a partition or task that was stopped by a PSTOP or a PDRAIN command.

Notes:

1. Do not use PGO ...,CANCEL after the message 1Q28I; if the command is a PGO with a partition identifier, VSE/POWER terminates the queue entry for the program which runs in that partition.

2. A reader task for a card reader that gives a 'device end' interrupt (READY has been pressed) is reactivated automatically; you need not issue the PGO command for such a VSE/POWER task.

3. The command resets a previously set termination condition (such as EOJ, PEND, or PSTOP cuu).

Format of the Command

The command can be used with distinct sets of operands as shown below:

Operation	Operands
{PGO|G}	partition,uraddr[,CANCEL|,IGNORE|,UNLOAD]
	tapeaddr{,CANCEL|,IGNORE|,UNLOAD}
	uraddr
	DEV,devname[,PARM='string']

Description of Operands

{PGO|G}	partition,uraddr[,CANCEL|,IGNORE|,UNLOAD]

This set of operands applies to a partition whose output is being spooled to tape.

partition
: For partition, specify the identifier (BG or Fn) of the applicable partition.

uraddr For uraddr, specify the address of the unit-record device for which VSE/POWER spools output to tape.

CANCEL Specify this operand if spooling onto tape is to be discontinued and the job currently being processed in the specified partition is to be canceled.

IGNORE Specify this operand if VSE/POWER is to continue processing using the mounted tape. In this case, the tape labels are destroyed.

UNLOAD Specify this operand to have the spool tape unloaded. VSE/POWER requests the operator to mount a new, valid tape.

```
| {PGO|G}        tapeaddr{,CANCEL|,IGNORE|,UNLOAD}          |
```

This set of operands applies to the tape-SYSIN reader and the POFFLOAD functions of VSE/POWER:

tapeaddr
For tapeaddr, specify the channel and unit address of the tape drive being used.

CANCEL Specify CANCEL if the affected tape function is to be stopped immediately without any further action (see also Note 1 on page 184).

IGNORE Specify IGNORE if VSE/POWER is to continue processing using the mounted tape. In this case, the tape labels are destroyed.

UNLOAD Specify UNLOAD to have the affected tape unloaded. VSE/POWER requests the operator to mount a new, valid tape.

```
| {PGO|G}        uraddr                                     |
```

Use this command format to reactivate a writer task associated with a unit-record device.

For uraddr, specify the channel and unit number of the unit-record device whose associated writer task is to be reactivated.

```
| {PGO|G}        DEV,devname[,PARM='string']                    |
```

Use this set of operands to tell a subsystem or program in another partition to reactivate a device-service task for a device owned by that subsystem (program).

DEV Code this operand as shown.

devname For devname, specify the name of the associated output device. This name must be the one by which the device-owning subsystem or application program knows the device.

PARM='string'
For string in PARM='string', supply a device and subsystem specific parameter string, if this is required. VSE/POWER's requirements regarding the parameter string are:

- It may not be longer than 60 characters. This includes blanks and commas that the program may need as delimiters.
- An apostrophe (') within the string must be entered as two successive apostrophes ('').

Examples of the PGO Command

PGO 00E
Reactivates a list task (for example, after the required form has been set up).

PGO BG,00E,IGNORE
Reactivates spooling to tape of output from partition BG to the device at 00E. A command such as this could be used, for example, after a new tape has been mounted; tape labels, if present on the newly mounted volume are destroyed.

PGO BG,00E,CANCEL
Causes the job in the background partition to be canceled.

PGO 280,IGNORE
Reactivates the task. If the task writes to tape, tape labels (if present) are destroyed.

PGO 280,CANCEL
Stops the task immediately without any further action.

PGO DEV,APPLPR
Reactivates a device-service task for a device known by the owning application program by the name APPLPR.

PHOLD: Place a VSE/POWER Job into the Hold or Leave State

The command puts one or more VSE/POWER jobs of a queue into the hold or leave state. It causes a job with a disposition of:

D (dispatchable) - To be changed to disposition H (hold).
K (keep) - To be changed to disposition L (leave).

VSE/POWER ignores the command if it is given for a queue entry with a disposition other than D or K; it informs the operator about this by a message.

Note: The operator at the central location can request any VSE/POWER job in the queues to be held. Any other user (at a remote work station, for example) can request a job to be held only if it was originated by this user or, in case of output, is to be routed to this user.

VSE/POWER accepts a PHOLD command with up to nine operands.

Format of the Command

Operation	Operands
{PHOLD|H}	queue[,*abc| ,ALL| ,class] ,jobname[,jobnumber[,jobsuffix]]] [,CCLASS=class] [,CDISP=disposition] [,CFNO=formnumber] [,CPRI=priority] [,CRDATE{=|>|<|¬=|>=|<=}date] [,CSYSID=sysid] [,CUSER=userid] [,FNODE=nodeid] [,FUSER=userid] [,TNODE=nodeid] [,TUSER=userid]
	Besides the operand queue, VSE/POWER requires at least one additional operand.

Description of Operands

queue For queue, give the name of the queue for which the command is to be executed. Specify:

LST For list queue
PUN For punch queue
RDR For reader queue
XMT For transmission queue

*abc Use this form as additional search argument if you want VSE/POWER to hold those queue entries whose names begin with the specified characters. For abc, you can specify up to eight characters that are common in the names of a group of queued VSE/POWER jobs.

ALL Specify ALL if all entries (with disposition D or K) of the named queue are to be placed into the hold or leave state. ALL is the default if you specify a queue with one or more keyword-type operands in order to have only a certain group of entries in the queue placed into the hold or leave state.

class For class, specify the applicable job class if the queue entries with this class assignment are to be placed into the hold or leave state.

jobname[,jobnumber[,jobsuffix]]
If your command applies to a specific job, specify for:

jobname
The name by which the job is known to VSE/POWER.

jobnumber
The number assigned to the job by VSE/POWER. This is of significance if two or more jobs with the same name are queued in the specified queue. Use the PDISPLAY command to get this number displayed by VSE/POWER.

jobsuffix
The VSE/POWER-assigned segment number if only a certain output segment of the named job is to be placed into the hold or leave state. You can use this operand only together with the VSE/POWER assigned job number. To get the correct jobsuffix, issue a PDISPLAY command for the job with FULL=YES specified.

Specifying PHOLD with queue,jobname (optionally with jobnumber) causes VSE/POWER to hold the queue entry found first to have a matching name (and number). An exception are queue entries that are the result of count-driven segmentation. In that case, VSE/POWER holds all related segment entries. If you specify also a job suffix, the hold request applies only to the specified segment of the named job.

CCLASS=class
The operand requests an affected queue entry to be placed into the hold or leave state if the entry's current class matches the specified class.

CDISP=disposition
The operand requests an affected queue entry to be placed into the hold or leave state if the entry's current disposition matches the specified disposition.

CFNO=formnumber
The operand requests an affected queue entry to be placed into the hold or leave state if the form number currently defined for the entry's output matches the specified form number.

CPRI=priority
The operand requests an affected queue entry to be placed into the hold or leave state if the priority currently assigned to the entry matches the specified priority.

CRDATE{=|>|<|¬=|>=|<=}date
In this string:

CRDATE requests those queue entries to be placed into the hold or leave state whose creation date is one of the following:

Equal to (=)	┐
Greater than (>)	|
Less than (<)	> the specified date
Not equal to (¬=)	|
Greater than or equal to (>=)	|
Less than or equal to (<=)	┘

date specifies the date against which the queue entries' creation dates are to be compared. It must have the format defined for the system (mm/dd/yy or dd/mm/yy). Do not specify mm greater than 12, dd greater than 31, and yy greater than the current year.

Note: In a shared spooling environment, all systems must have the same date format.

CSYSID=sysid
The operand requests an affected queue entry to be placed into the hold or leave state if the system identifier currently assigned to the entry matches the specified identifier.

CUSER=userid
The operand requests an affected queue entry to be placed into the hold or leave state if the queue entry's "from user" or "to user" matches the specified user identifier. For a definition

of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

FNODE=nodeid
: The operand requests an affected queue entry to be placed into the hold or leave state if the entry's "from node" name matches the specified node name. However, VSE/POWER does not verify that the specified node name is defined in the network-definition table.

FUSER=userid
: The operand requests an affected queue entry to be placed into the hold or leave state if the entry's "from user" identifier matches the specified user identifier. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause the status of local-user originated queue entries to be changed.

TNODE=nodeid
: The operand requests an affected queue entry to be placed into the hold or leave state if the entry's "to node" name matches the specified node name. This operand is valid only if your specification for queue is XMT.

TUSER=userid
: The operand requests an affected queue entry to be placed into the hold or leave state if the entry's "to user" identifier matches the specified user identifier. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  The TUSER specification cannot be used for changing the status of reader-queue entries.

  Specifying TUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause a status change of entries destined for local users.

Examples of the PHOLD Command

PHOLD RDR,ALL
: Changes to H or L the disposition of those jobs in the reader queue which have a disposition of D or K, respectively.

H LST,C
: Changes to H or L the disposition of jobs in the list queue with job characteristics as follows:

A disposition of D or K, respectively.
Output class C.

H LST,CRDATE<05/15/87
Changes to H or L the disposition of those jobs in the list queue that were created before May 15, 1987.

H XMT,TNODE=NODEB
Changes to H or L the disposition of those jobs in the transmission queue which have a disposition of D or K, respectively, and are to be routed to the node NODEB.

PINQUIRE: Request Non-Local Status Information

The command can be used to request status information about one or all of the following:

A BSC line
An external output device
A node
An SNA logical unit

Note: If VSE/POWER runs on a virtual machine under VM/SP and this machine happens to enter CP mode, you cannot use the command's short form (I); this would cause the VM/SP control program (CP) to attempt IPL.

A discussion of VSE/POWER's display in response to the command is given under "Display of Status Information" on page 193.

Format of the Command

Operation	Operands
{PINQUIRE|I}	{ALL| lineaddr| luname| DEV=devname| NODE=nodeid}

Description of Operands

ALL
: Specify this operand if you want a display of all non-local status information which VSE/POWER can supply in response to this command.

DEV=devname
: For devname, specify the name of the external output device whose status you want to be displayed. This name must be the one by which the device-owning subsystem or application program knows the device.

lineaddr
: For lineaddr, supply the address of the BSC line for which status information is requested. Specify the address in the form cuu (channel and unit number).

luname
: For luname, give the name of the logical unit for which the status information is desired.

NODE=nodeid
: For nodeid, specify the name of the node for which status information is desired.

Display of Status Information

VSE/POWER displays status information in response to a PINQUIRE command in the form of messages. This section explains the display by examples.

The command is: PINQUIRE ALL

Status-display example:

```
1R56I  030  PROCESSING 22
1R56I  031  PROCESSING 150
1R56I  032  NOT INITIATED
 ...  ...  ...
1R56I  036  INACTIVE
1R56I  056  PROCESSING NODE NODEA
1R56I  RT74 LOGGED ON
1R78I  DEVICE PLOT1 ACTIVE
1R78I  DEVICE PLOT2 SETUP IN PROGRESS
```

There is one line of status information for each of the resources whose status is to be displayed. The line is either message 1R56I (for a display related to remote job entry or networking) or message 1R78I (for a display related to external device control.

Each display line includes a status descriptor. A list of possible status descriptors follows on the next page. The list gives a short explanation of these descriptors.

Status Descriptor	Explanation
Message 1R56I:	
INACTIVE	The indicated BSC line has been started by the central operator by way of a PSTART command, but no user is currently signed on.
LOGGED ON	The RJE,SNA session with the named logical unit is logged on, but no processor is active.
LOGGING ON	The RJE,SNA session with the named logical unit is in the process of being logged on.
NOT INITIATED	The indicated BSC line has not been started by the central operator.
NOT LOGGED ON	Applies to RJE,SNA. No RJE,SNA user is currently logged on to the named logical unit.
NOT SUPPORTED	The indicated BSC line was not defined during generation of the VSE/POWER control tables.
PROCESSING	For RJE: user has entered a valid SIGNON or LOGON command. This means - For RJE,BSC: The indicated line is active; VSE/POWER displays the applicable remote identifier. - For RJE,SNA: The session is logged on; a processor is either sending or receiving. For networking: a session is in process via the line and with the named node. If you inquire about a specific node, then the involved transmitter and receiver tasks have the status indicated by the status code as shown in the next example.
SESSION PENDING	The session to the named node is not yet established and is waiting for activity from this node.

Message 1R78I:

ACTIVE	Spooled output is being transferred for physical output on the named device.
CONNECTION PENDING	VSE/POWER received a PSTART command for the external device, but the subsystem (program) owning the device has not yet connected to VSE/POWER.
INACTIVE	The named device has been started successfully, but the related VSE/POWER task (DST) neither is processing a queue entry nor has made an attempt to select a queue entry for output processing.
SETUP IN PROGRESS	VSE/POWER passed a setup-device request to the device owning subsystem (program), but the subsystem has not yet returned a 'setup-processed' signal.
STARTING	VSE/POWER passed a start-device request to the device owning subsystem (program), but the subsystem has not yet returned a 'request-completion' response.
WAITING FOR OPERATOR REACTIVATION	VSE/POWER's device service task waits for a 'device' command (as defined by the device owning subsystem - usually a PGO or PSETUP command) to be issued by the external device operator.
WAITING FOR WORK	VSE/POWER's output queues contain no queue entry for output on the named device, which operates under subsystem control.

The command is: I NODE=NODEA

Status-display Example:

```
1R56I 056 PROCESSING NODE NODEA
1R56I JOB-TRANSMITTER 1=A 2=D 3=I 4=D 5=D 6=D 7=D
1R56I OUT-TRANSMITTER 1=A 2=D 3=I 4=D 5=D 6=D 7=D
1R56I JOB-RECEIVER... 1=I 2=I 3=I 4=I 5=I 6=I 7=I
1R56I OUT-RECEIVER... 1=A 2=I 3=I 4=I 5=I 6=I 7=I
```

VSE/POWER displays, as a header message, the one it displays for the node in response to a PINQUIRE ALL command. In addition, VSE/POWER provides status information about the transmitter and receiver tasks communicating with the node. In these additional messages, the number identifies the task (transmitter or receiver); the letter, a status code, gives the status as shown below:

A = Active The task is transmitting or receiving a queue entry.

D = Drained — The task is not available for any activity on the node. Issue a PACT command to make the task available again.

I = Inactive — The task is currently not in use. It is available whenever required for a transmission.

H = Halting — The task is being deactivated (drained). When the current transmission is finished, the task is no longer available. Issue a PACT command to reactivate the task.

The command is: I DEV=PLOT1

Status-display Example 1:

```
     1R78I  DEVICE PLOT1 ACTIVE
(1)  1R78I  CLASSES: ABCD - STATUS: HALTING
(2)  1R78I  DESTINATION: DEPT3244 DEPT3245
```

(1) 1R78I CLASSES: ABCD - STATUS: HALTING
The device PLOT1 is used for output of the named classes (A, B, C, and D) under control of a program in another partition. A status of HALTING means that VSE/POWER has passed a stop-device order to the owning subsystem, but the device has not yet been stopped.

(2) 1R78I DESTINATION: DEPT3244 DEPT3245
Gives the identifiers (DEPT3244 and DEPT3245) of the users to which the output is dedicated.

Status-display example 2:

```
(1)  1R78I  DEVICE PLOT1 WAITING FOR WORK
(2)  1R78I  CLASSES: ABCD - QUEUE:  L
     1R78I  DESTINATION: DEPT3244 DEPT3245
```

(1) 1R78I DEVICE PLOT1 WAITING FOR WORK
The device PLOT1 waits for an output entry to be put into the list queue for either of the users DEPT3244 or DEPT3245.

(2) 1R78I CLASSES: ABCD - QUEUE: L
The device PLOT1 is used for output of the named classes (A, B, C, and D) under control of a program in another partition. L means LST queue (P: PUN queue).

PLOAD: Load a Network Definition Table

This command applies only to the networking function. The command dynamically loads a new network definition table (NDT) while VSE/POWER is still using an already loaded (old) NDT. Any task currently using the old table continues to do so. A task that becomes active after PLOAD is issued uses the new table.

The following node-related values of the old NDT remain active for a connected node until this node is restarted:

Buffer size
Number of buffers
Number of transmitter and receiver tasks.

Notes:

1. The node name in the new table's entry with LOCAL=YES must match the node name in the corresponding entry of the old table.

2. If you issue too many PLOAD commands for large tables, you may run out of GETVIS space and have to start-up VSE/POWER anew.

Format of the Command

Operation	Operands
PLOAD	PNET,phasename

Description of Operands

PNET Code this operand as shown.

phasename
For phasename, specify the name of the network definition table that is to be loaded. This is the name that was used for phasename in the applicable PNODE generation macro. The phase must reside in one of the sublibraries accessible from the VSE/POWER partition.

Example of the PLOAD Command

```
PLOAD PNET,NET2
    Loads the network definition table NET2; checks the table and, if
    OK, uses it for all further processing.
```

POFFLOAD: Save Queue Entries On Tape / Restore Queue Entries From Tape

The command enables you to save or restore queue entries of the various queues, using a tape as intermediate storage.

You can request VSE/POWER to save:

- The queue entries of all queues, or
- All queue entries of a selected queue, or
- The queue entries of a selected class of a queue, except for the queue entries in the transmission queue.

You can request VSE/POWER to restore:

- Individual queue entries from the tape, or
- All queue entries from the tape.

A tape created with the POFFLOAD function can be transferred from one VSE/POWER controlled system to another. Output queue entries saved on tape may be printed or punched out directly from that tape.

To obtain a record of queue entries saved on tape, issue the PDISPLAY TAPE=cuu... command after you have produced a POFFLOAD tape. It causes VSE/POWER to display information about each queue entry on the tape.

Before you start a save operation, ensure that your tape reel's capacity is large enough to hold at least one of the queue entries that are to be saved. For BACKUP, the tape capacity must be large enough to hold the largest queue entry.

Format of the Command

The command can be used with distinct sets of operands:

Operation	Operands
{POFFLOAD|O}	BACKUP,queue,tapeaddr, [density],[A|class|*][,NOREW]
	BACKUP,{XMT|ALL},tapeaddr,[density][,NOREW]
	SAVE,queue,tapeaddr, [density],[A|class|*][,NOREW]
	SAVE,{XMT|ALL},tapeaddr,[density][,NOREW]
	{LOAD|SELECT},{queue|ALL},tapeaddr, [newclass],[NOREW],[oldnode][,NOJNO]

Description of Operands

BACKUP You can use the command with this operand to "save" all entries of a queue, including those which are being processed, by creating a backup tape.

While processing a POFFLOAD BACKUP command, VSE/POWER:

1. Suspends the processing of any queue entries.
2. Locks the entire queue file for exclusive use.
3. Does not allow any of the following commands to be entered: PALTER, PDELETE, PHOLD, PRELEASE, or PRESET.

This means that no other VSE/POWER task can access the queue file. In a shared spooling environment, the command causes also all other sharing systems to wait. Queue entries so saved remain in their queue unchanged.

Only one POFFLOAD BACKUP command can be processed by VSE/POWER at a time.

If your output tape becomes full while the command is being processed, then VSE/POWER requests the operator to mount a new tape volume. Again, the queue file remains locked during this time, which means that no other VSE/POWER task can access any queues. In view of these restrictions, IBM recommends that you do a backup only when no other VSE/POWER task is active on your own or sharing VSE system.

SAVE Use the command with SAVE if a part or all of a VSE/POWER queue's contents are to be saved and VSE/POWER should continue to operate. However, a POFFLOAD SAVE command saves only queue entries with a disposition of D or K.

When the save operation for a queue entry is complete, VSE/POWER:

- Deletes the entry if it had a disposition of D.
- Changes the entry's disposition to L if it had a disposition of K.

LOAD A POFFLOAD LOAD command causes queue entries previously saved on tape (by the POFFLOAD BACKUP or SAVE command) to be reloaded into the VSE/POWER queues. You can also use the POFFLOAD LOAD command to load a VSE/POWER spool tape (DISP=T) back into the VSE/POWER queues. VSE/POWER changes the disposition of these queue entries from T to D. The DBLK size from the input spool tape may be any value up to the maximum, which is 12,288 bytes.

If the queue entries to be reloaded are contained on two or more tape volumes, you must provide one POFFLOAD LOAD command for each of the volumes.

VSE/POWER assigns a new job number and a new creation date to each of the reloaded queue entries. Specify NOJNO if this is not desirable.

SELECT A POFFLOAD SELECT command causes VSE/POWER to reload individual queue entries from a POFFLOAD tape or a spool tape into the VSE/POWER queues. The system operator is prompted via message

```
1R41D SPECIFY TAPE SELECT CRITERIA OR PRESS ENTER TO QUIT
```

to specify the selection criteria. Only queue entries with matching criteria are reloaded. The operator replies with one (or a meaningful combination) of the following operands:

```
{*abc|                          ┐
 ALL|                           > Optional if at least one of the
 class|                         | following operands is specified.
 jobname[,jobnumber]}           ┘
[,CCLASS=class]
[,CDISP=disposition]
[,CFNO=formnumber]
[,CPRI=priority]
[,CRDATE{=|>|<|┐=|>=|<=}date]
[,CSYSID=sysid]
[,CUSER=userid]
[,FNODE=nodeid]
[,FUSER=userid]
[,TNODE=nodeid]
[,TUSER=userid]
```

The selection operands are similar to those of the queue manipulation commands. For a description of these operands, see either the PALTER or PDISPLAY command.

queue The operand specifies the queue whose entries are to be saved or loaded. For queue, you can specify one of the following:

If your first operand is BACKUP or SAVE:

LST For list queue
PUN For punch queue
RDR For reader queue

If your first operand is LOAD or SELECT: the same as above and also

XMT For transmission queue

XMT|ALL Specify:

XMT If, for POFFLOAD with BACKUP or SAVE, only queue entries in the transmission queue are to be written onto tape.

ALL If all entries of all VSE/POWER queues are to be written onto tape (in accordance with specified restrictions) or to be loaded from tape.

tapeaddr
For tapeaddr, specify the device address of the tape drive that is to be used. You specify this address in the form cuu (channel and unit number).

density For density, specify the density (or tape mode) applicable to the tape drive you are using. If you omit the operand, VSE/POWER assumes the mode defined for the tape drive during system start-up.

For the correct value which you should specify refer to the description of the ASSGN statement in the publication VSE/Advanced Functions, System Control Statements. For the IBM 3480 tapes, a density specification is ignored.

A|class|*
For class, specify the class or classes of the queue entries that have to be saved. You can specify up to four classes.

Specify an asterisk (*) if you want the queue entries of all classes to be saved (in accordance with existing restrictions).

VSE/POWER assumes class A if you omit this operand for a BACKUP or SAVE request with LST, PUN, or RDR.

newclass
For newclass, specify the class with which the saved queue entries are to be loaded. The specification causes VSE/POWER to overwrite the class recorded for the queue entries on tape. If you do not specify the operand, VSE/POWER loads the queue entries with the recorded class assignment.

NOJNO Use this operand if you do not want VSE/POWER to assign new job numbers to the queue entries that are being loaded. If you omit the operand, VSE/POWER assigns a new number and a new creation date to each of the queue entries that is being loaded from tape.

NOREW Specify this operand if the tape is not to be rewound before, and not to be unloaded after processing of a POFFLOAD command. If you omit this operand, the tape is rewound before processing and unloaded after processing of a POFFLOAD command. Together with BACKUP or SAVE, this operand allows multiple queue off-loads to be saved on one tape.

oldnode This operand applies if you performed the BACKUP or SAVE operation on one system and the LOAD operation on another, or if the system's node name was changed in between.

For oldnode, specify the node name for which VSE/POWER is to replace the "from node" by the node name defined for your system in the currently used network definition table.

If oldnode is specified for LOAD or SELECT, the "to node" of each entry transferred from tape is replaced by the local node name.

Note: The VSE/POWER Networking function does not allow you to change the local node name (1) while VSE/POWER is running or (2) at bring-up time, when a warm start is performed. To change the local node name, you must perform a cold start. All queue entries of the old spool file must first be saved via the POFFLOAD BACKUP function and later restored via the POFFLOAD LOAD function. VSE/POWER automatically changes the old node name to the new name.

Examples of the POFFLOAD Command

POFFLOAD SAVE,LST,280,,AM
Saves list queue entries of classes A and M on the tape mounted on the drive at the address 280. VSE/POWER uses the default density; it continues processing queue entries in parallel.

O SAVE,LST,280,,AM
Saves list queue entries of classes A and M on the tape mounted on the drive at the address 280. VSE/POWER uses the default density.

O BACKUP,ALL,280
Produces a backup of all queues. The tape is mounted on the drive at the address 280. VSE/POWER uses the default density; it does not process any queue entries until this backup operation is finished.

O BACKUP,RDR,280,,*
Produces a backup of all entries of the reader queue, regardless of their class assignments.

POFFLOAD LOAD,LST,280
Loads the saved queue entries back to the list queue. The tape to be used is mounted on the drive at address 280.

POFFLOAD LOAD,LST,280,,NOREW,,NOJNO
Loads the saved queue entries back to the list queue. The tape is not to be rewound and no new job numbers (and creation dates) are to be assigned to the queue entries that are being loaded.

POFFLOAD SELECT,LST,280,,NOREW
Loads only those queue entries back to the list queue that you specify in response to message 1R41D.

PRELEASE: Remove VSE/POWER Jobs from Hold/Leave State

The command takes one or more VSE/POWER jobs out of the hold (disposition H) or leave (disposition L) state and makes them available for processing. VSE/POWER accepts the command with up to nine operands.

Using the command changes the sequence of the jobs in the specified queue; they are processed in accordance with the classes and priorities defined for them. If an affected job is in the reader (RDR) queue and was executed previously, VSE/POWER gives the job a new job number.

After the command has been processed, the disposition of an affected job is changed as follows:

An original disposition of L is changed to K.
An original disposition of H is changed to D.

Note: Queue entries with a temporary disposition of A, X, or Y are not changed.

A job placed into the reader queue from a work-station user in the hold or leave state can be released only by that user. The same is true for output of a VSE/POWER job in the hold or leave state and directed to a work station user.

Format of the Command

Operation	Operands
{PRELEASE|R}	queue[,*abc| ,ALL| ,class| ,jobname[,jobnumber][,jobsuffix]] [,CCLASS=class] [,CDISP=disposition] [,CFNO=formnumber] [,CPRI=priority] [,CRDATE{=|>|<|¬=|>=|<=}date] [,CSYSID=sysid] [,CUSER=userid] [,FNODE=nodeid] [,FUSER=userid] [,TNODE=nodeid] [,TUSER=userid]
	Besides the operand queue, VSE/POWER requires at least one additional operand.

Description of Operands

queue For queue, specify the name of queue for which the command is to be executed:

LST	For list queue
PUN	For punch queue
RDR	For reader queue
XMT	For transmission queue

*abc Use this form if VSE/POWER is to release only those jobs in the named queue whose names begin with the specified characters. For abc, you can specify up to eight characters that are common in the names of a group of queued VSE/POWER jobs.

ALL Specify ALL if all entries of the named queue are to be released, except those with a disposition of L. Specify ALL with one of the other keyword-type search operands (or omit this specification) if a certain group of entries in the queue is to be released.

class For class, specify the applicable job class if all queue entries of this class in the specified queue are to be released, except those with a disposition of L.

jobname[,jobnumber[,jobsuffix]]
If your command applies to a specific job, specify for:

jobname
The name by which the job is known to VSE/POWER.

jobnumber
The number assigned to the job by VSE/POWER. This is of significance if two or more jobs with the same name are queued in the specified queue. Use the PDISPLAY command to get this number displayed by VSE/POWER.

jobsuffix
The VSE/POWER-assigned segment number if only a certain output segment of the named job is to be released. You can use this operand only together with the VSE/POWER assigned job number. To get the correct jobsuffix, issue a PDISPLAY command for the job with FULL=YES specified.

PRELEASE with queue,jobname (optionally with jobnumber) causes VSE/POWER to release the queue entry found first to have a matching name (and number). An exception are queue entries that are the result of count-driven segmentation. In that case, all related segment entries are also released. If you specify a job suffix, the release request applies only to the specified segment of the named job.

CCLASS=class
The operand requests an affected queue entry to be released if the class currently assigned to the entry matches the specified class.

CDISP=disposition
The operand requests an affected queue entry to be released if the disposition currently assigned to the entry matches the specified disposition.

CFNO=formnumber
The operand requests an affected queue entry to be released if the form number defined for the output of the entry matches the specified form number.

CPRI=priority
The operand requests an affected queue entry to be released if the priority currently assigned to the entry matches the specified priority.

CRDATE{=|>|<|¬=|>=|<=}date
In this string:

CRDATE indicates that those queue entries are to be released whose creation date is one of the following:

Equal to (=)	┐
Greater than (>)	|
Less than (<)	> the specified date
Not equal to (¬=)	|
Greater than or equal to (>=)	|
Less than or equal to (<=)	┘

date specifies the date against which the queue entries' creation dates are to be compared. It must have the format defined for your system (mm/dd/yy or dd/mm/yy). Do not specify mm greater than 12, dd greater than 31, and yy greater than the current year.

Note: In a shared spooling environment, all systems must have the same date format.

CSYSID=sysid
The operand applies only to a shared spooling environment. It requests an affected queue entry to be released if the processing system defined for the entry matches the specified processing system.

CUSER=userid
The operand requests an affected queue entry to be released if the queue entry's "from user" or "to user" matches the specified user identifier. For a definition of userid, refer to section

"Conventions for Showing Command and Statement Formats" on page 15.

FNODE=nodeid
: The operand requests an affected queue entry to be released if the entry's "from node" name matches the specified node name. However, VSE/POWER does not verify that the specified node name is defined in the network-definition table.

FUSER=userid
: The operand requests an affected queue entry to be released if the entry's "from user" identifier matches the specified user identifier. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause the release of entries originated by local users.

TNODE=nodeid
: The operand requests an affected queue entry to be released if the entry's "to node" name matches the node name specified for nodeid in this operand. This operand is valid only if your specification for queue is XMT.

TUSER=userid
: The operand requests an affected queue entry to be released if the entry's "to user" identifier matches the specified user identifier. For a definition of userid, refer to section "Conventions for Showing Command and Statement Formats" on page 15.

  The TUSER specification cannot be used for releasing queue entries residing in the reader queue.

  Specifying TUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user identifier of LOCAL; it does not cause a release of entries destined for local users.

Examples of the PRELEASE Command

PRELEASE LST,PAYROLL
: Releases VSE/POWER job PAYROLL in the list queue.

PRELEASE RDR,ALL
: Releases all VSE/POWER jobs held in the reader queue with a disposition of H.

PRELEASE LST,*PAY
: Releases all VSE/POWER jobs in the list queue whose names begin with PAY.

```
PRELEASE RDR,ALL,CRDATE<=05/15/87
    Releases all entries in the reader queue whose creation date is less
    than or equal to May 15, 1987.

R LST,CFNO=SPEC
    Releases all entries in the LST queue which require forms SPEC.
```

PRESET: Reset the Job Status

The command is used in a shared spooling environment. It enables the operator of a system with an active VSE/POWER to reset the status of jobs that were in execution on another, abnormally terminated system.

Normally, the command is not required. VSE/POWER attempts automatic recovery whenever the abnormally terminated system had control of the spool files, except when this system was the last one having control over these files. In that case, VSE/POWER attempts this recovery during next VSE/POWER start-up.

The command must be used only if:

- Another VSE/POWER (or VSE system) has abnormally terminated while executing jobs from the VSE/POWER queues.
- The abnormally terminated system cannot be reinitialized within an acceptable time.
- Jobs which were executed (including print output) are required urgently.
- No other sharing VSE/POWER system has received message 1QB7I since the abnormal termination of the failing system.

Format of the Command

Operation	Operands
PRESET	sysid[,sysid,...]

Description of Operands

sysid The operand specifies the identifier of the VSE/POWER controlled system that terminated abnormally.

For sysid, specify the identifier, the one-digit number which VSE/POWER displays with message 1Q12I during start-up. You can specify up to nine identifiers.

VSE/POWER verifies that none of the specified identifiers is that of your own system. It requests the operator to confirm the identifiers that are specified in the PRESET command.

A specification error in the command can lead to unexpected results. If, for example, you enter the identifier of a running system not your own, then VSE/POWER cannot determine that the identifier is wrong and sets all jobs for this system into the inactive state. This could cause data integrity problems later on.

Example of the PRESET Command

```
PRESET 2,4
    Initiates queue-file recovery for the systems with the identifiers 2
    and 4.
```

PRESTART: Restart Printed or Punched Output

The command restarts printed or punched output for a VSE/POWER job from the beginning or from a specified page or card of this output. The command cannot be used for restarting an output after a PSTOP command has been issued for the associated writer task; PSTART must be given to reactivate this task.

Restarting printed output is not possible for page numbers higher than 65,535. For page/card count, see also page 164.

If you enter a PRESTART command while VSE/POWER is printing separator pages, these pages are printed a second time.

Format of the Command

The command can be used with distinct sets of operands as shown below:

Operation	Operands
{PRESTART|T}	uraddr,[1|count][,index]
	DEV,devname[,1|,count][,PARM='string']

Description of Operands

DEV Code this operand as shown. It indicates to a subsystem or program that output on a device under control of that subsystem (program) is to be restarted.

devname For devname, specify the name of the output device to be used. This name must be the one by which the device-owning subsystem or program knows the device.

uraddr For uraddr, specify the channel and unit address of the unit record device whose associated list or punch task is to be restarted.

1|count For count with uraddr, specify a signed or an unsigned value from 0 to 65,535 as the restart point (the default is 1). This value is the number of pages to be counted in case of a list task; it is the number of card images to be counted in case of a punch task.

A plus sign tells VSE/POWER to count in the forward direction; a minus sign indicates that counting is to be done backward - in either case from the point where processing was interrupted. An unsigned value requests VSE/POWER to count from the beginning of the output.

Assume, for example, that page 30 of a print output is being processed when you enter the PRESTART command. VSE/POWER stops printing for page 30 and continues as follows:

For a specification of +15 - On Page 45
For a specification of -15 - On Page 15

If you specify too great a backward count or omit the operand, VSE/POWER restarts the output from the first page or card. If you specify too great a forward count, VSE/POWER ignores the command with message 1Q42I.

For printed output, VSE/POWER counts actual pages. If the output was spooled by VSE/POWER, such a page is either of the following:

The occurrence of a "skip to channel-1" command.
A page with as many lines as are defined in the printer's FCB.

If the output was spooled on a system without VSE/POWER, then VSE/POWER counts only the occurrence of a "skip to channel-1" command as a page.

For count with DEV,devname, specify a meaningful signed or unsigned restart value in accordance with the output control requirements of the owning subsystem (or program). VSE/POWER accepts values from 0 to 999999.

index For index, which applies only to an IBM 3800 printer, specify the copy-group index that is to be used. If you omit the operand, output processing continues with the copy-group index currently used. If you specify 0, VSE/POWER uses a copy-group index value of 1.

PARM='string'
For string in PARM='string', supply a device and subsystem specific parameter string, if this is required. VSE/POWER's requirements regarding the parameter string are:

- It may not be longer than 60 characters. This includes blanks and commas that the program may need as delimiters.
- An apostrophe (') within the string must be entered as two successive apostrophes ('').

Example of the PRESTART Command

```
PRESTART 00E,-10
    Causes VSE/POWER to reprint the preceding ten pages and to continue.
```

PSETUP: Check the Forms Alignment of Printed Output

The command causes VSE/POWER to print one or more pages of list output with all printable characters of a page replaced by the character X. When afterwards the PGO command is issued to resume printing, VSE/POWER prints those pages again with their actual contents. The command allows you to check and adjust forms alignment.

Necessary adjustments are to be made manually while VSE/POWER prints the setup pages.

Use the PSETUP command only after having received message 1Q40A or 1QA5A.

If a PSETUP command is given for a queue entry that was previously stopped by PSTOP cuu,RESTART, printing starts with the first page.

Format of the Command

The command can be used with distinct sets of operands as shown below:

Operation	Operands
{PSETUP|U}	uraddr[,1|,nn]
	DEV,devname[,1|,nn][,PARM='string']

Description of Operands

DEV Code this operand as shown. It indicates to a subsystem or program in another partition that setup pages are to be printed on a printer under control of that subsystem (program).

devname For devname, specify the name of the printer to be used. The name you specify is the one by which the device-owning subsystem or program knows the printer.

uraddr For uraddr, specify the channel and unit address of the printer on which setup pages are to be printed.

1|nn For nn with uraddr, specify the number of setup pages that you want VSE/POWER to print. You can request VSE/POWER to print up to 99 setup pages. If you omit the operand, VSE/POWER prints one setup page. A page in this context is either a full page or the occurrence of a "skip to channel-1" command.

For nn with DEV,devname, your specification is the same as described above for nn with uraddr. VSE/POWER passes the value to the application program which controls the named device.

PARM='string'
For string in PARM='string', supply a device and subsystem specific parameter string, if this is required. VSE/POWER's requirements regarding the parameter string are:

- It may not be longer than 60 characters. This includes blanks and commas that the program may need as delimiters.
- An apostrophe (') within the string must be entered as two successive apostrophes ('').

Example of a PSETUP Command

```
PSETUP 00E,2
    Print two setup pages:.

      Page 1

        xxxx            xxxxxx      xxxxxx
        xx xxxxxx       xxxxx   x   xxxxxxxx
        xx xxxxxxxx     xxxxx   x     xxxxxx


      Page 2

        xxxx            xxxxxx      xxxxxx
        xx xxxxx        xxxxx   x     xxxxxx
        xx xxxx         xxxxx   x     xxxxxx
```

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

```
      Page 1

        NAME            NUMBER      SALARY
        L. KEEFFE       69060   $   1,080.05
        N. REYNOLDS     41240   $     798.83


      Page 2

        NAME            NUMBER      SALARY
        D. HALEY        33110   $     497.11
        A. WATT         76779   $     699.42
```

PSTART: Start a Task or a Partition

The command can be used to start a VSE/POWER task or a partition under control of VSE/POWER.

Format of the Command

The command has distinct sets of operands as shown below:

Operation	Operands	
{PSTART|S}	{LST|PUN},uraddr, [A|class],[buf],[VM],[SEP|NOSEP] [,SKIP|,NOSKIP]	
	{LST|PUN},uraddr,tapeaddr,,,[SEP|NOSEP] [,SKIP|,NOSKIP]	
	RDR,uraddr,[A|class][,buf]	
	RDR,tapeaddr[,A|,class]	
	RDR,uraddr, [A|class],duaddr[,buf]	
	RDR,duaddr, [A|class],'file-id',[1|vols],[S][,V]	
	DEV,devname, subsystem[,A|,class][,PARM='string']	
	{BG|Fn},[class],[A|outclass][,MT]	
	RJE,lineaddr,[password][,TRACE]	(1)
	RJE,SNA	(2)
	PNET,nodeid, [node-password],lineaddr, [line-password],[TRACE][,NR]	(1)
	PNET,nodeid, [node-password],,,[TRACE][,NR]	(2)
	{CNSLTR|DUMPTR}	

(1) Used for BSC
(2) Used for SNA

Description of Operands

```
{PSTART|S}      {LST|PUN},uraddr,
                [A|class],[buf],[VM],[SEP|NOSEP]
                [,SKIP|,NOSKIP]
```

Use this command format:

- If VSE/POWER is to process output that has been spooled to disk.
- If you do or do not want the writer task to produce separator pages or cards on all output, regardless of the JSEP value associated with the job.
- If your VSE system operates as a virtual machine under VM/SP and the processing of spooled output is to be started. You may use this operand set in order to start the processing of spooled output under control of a virtual VSE system. In that case, however, VSE/POWER can issue an automatic CP CLOSE for the printer or punch only if your VSE system operates with a MODE=VM supervisor.

LST Code this if spooled list output is to be processed.

PUN Code this if spooled punch output is to be processed.

uraddr For uraddr, specify the channel and unit address of the output device which VSE/POWER is to use.

Under VM/SP, uraddr is the address of the virtual output device that is to be used.

A|class Specify, for class, the desired class or classes if the default class cannot be used:

VSE/POWER accepts up to four letters of the alphabet, each letter representing a class of output that VSE/POWER is to write to the named device.

buf The operand is valid only for a list task (PSTART LST,...).

For buf, specify the number of buffers that the task is to use. You can specify one of the following:

1 = One buffer.
2 = Two buffers.
D = Two input and two output buffers.

If you omit the operand, VSE/POWER starts the task with one buffer available.

Note: Storage for the buffers is used by a task only as long as the task is active.

VM Specify VM to tell VSE/POWER that the LST or PUN device is a virtual device owned by VM/SP. This causes VSE/POWER to generate VM/SP-CP commands as follows:

```
SP cuu TO {userid|SYSTEM} CLASS c COPY nnn FORM ffff
CL cuu NAME jobname jobnumber
```

SEP Code this operand if you want separator pages or cards for each job output, regardless of any JSEP specification for a job (in the * $$ LST or * $$ PUN statement). VSE/POWER takes the number of separator pages or cards from the JSEP operand of the POWER generation macro. However, if JSEP=0 was specified in the POWER generation macro, VSE/POWER assumes one separator page or card.

NOSEP Code this operand if you do not want any separator pages or cards at all, regardless of any JSEP specification for a job.

If you do not specify SEP|NOSEP, the VSE/POWER writer task takes the JSEP specification from the associated * $$ LST or * $$ PUN statement.

SKIP Code this operand if you want VSE/POWER to insert a skip-to-channel 1 preceding the job output. VSE/POWER inserts a skip-to-channel 1 if

- A write command occurs and no skip-to-channel 1 was found earlier, or
- Only control commands were found in the output.

NOSKIP Code this operand if you do not want VSE/POWER to insert a skip-to-channel 1.

This operand overrides a SKIP=YES specification in the autostart SET statement, if present.

If you omit the SKIP|NOSKIP operand, VSE/POWER takes the SKIP=YES option from the autostart SET statement (if present) as default.

```
{PSTART|S}      {LST|PUN},uraddr,tapeaddr,,,[SEP|NOSEP]
                [,SKIP|,NOSKIP]
```

Use this set of operands if you want VSE/POWER to print or punch out output that was spooled to tape.

LST Code this if spooled list output is to be processed.

PUN Code this if spooled punch output is to be processed.

uraddr For uraddr, specify the channel and unit address of the output device which VSE/POWER is to use.

tapeaddr
For tapeaddr, specify the channel and unit address of the tape drive on which the spool tape is mounted. This address must be specified in the form X'cuu'.

SEP Code this operand if you want separator pages or cards for each job output, regardless of any JSEP specification for a job (in the * $$ LST or * $$ PUN statement). VSE/POWER takes the number of separator pages or cards from the JSEP operand of the POWER generation macro. However, if JSEP=0 was specified in the POWER generation macro, VSE/POWER assumes one separator page or card.

NOSEP Code this operand if you do not want any separator pages or cards at all, regardless of any JSEP specification for a job.

If you do not specify SEP|NOSEP, the VSE/POWER writer task takes the JSEP specification from the associated * $$ LST or * $$ PUN statement.

SKIP Code this operand if you want VSE/POWER to insert a skip-to-channel 1 preceding the job output. VSE/POWER inserts a skip-to-channel 1 if

- A write command occurs and no skip-to-channel 1 was found earlier, or
- Only control commands were found in the output.

NOSKIP Code this operand if you do not want VSE/POWER to insert a skip-to-channel 1.

This operand overrides a SKIP=YES specification in the autostart SET statement.

If you omit the SKIP|NOSKIP operand, VSE/POWER takes the SKIP=YES option from the autostart SET statement (if present) as default.

```
{PSTART|S}      RDR,uraddr,[A|class][,buf]
```

Use this set of operands if VSE/POWER is to spool input to disk.

RDR Code this operand as shown.

uraddr For uraddr, specify the channel and unit address of the input device which VSE/POWER is to use.

A|class VSE/POWER accepts either of the following:

- One of the letters A through Z.
- As a partition dependent specification one of the numbers 0 through 9 (where 0 = partition BG, 1 = partition F1, 2 = partition F2, and so on).

VSE/POWER assigns the specified class to all jobs submitted without a class specification in their * $$ JOB cards, except when an * $$ CTL statement is in effect.

buf For buf, specify the number of buffers that the task is to use. You can specify one of the following:

1 = One buffer.
2 = Two buffers.

If you omit the operand, VSE/POWER starts the task with one buffer available.

Note: Storage for the buffers is used by a task only as long as the task is active.

```
{PSTART|S}      RDR,tapeaddr[,A|,class]
```

Use this set of operands if VSE/POWER is to spool input from tape.

RDR Code this operand as shown.

tapeaddr
For tapeaddr, specify the channel and unit address of the tape drive on which the input tape volume is mounted.

A|class For class, the rules as given above for the RDR operand set apply.

```
{PSTART|S}      RDR,uraddr,
                [A|class],duaddr[,buf]
```

Use this set of operands to start a reader task that is to switch spooling of input from a card reader to a diskette I/O unit.

Except for the operand duaddr, the operands of this set are the same as the corresponding operands of the RDR set described above.

duaddr For duaddr, specify the channel and unit address of the diskette I/O unit which VSE/POWER is to use.

```
{PSTART|S}      RDR,duaddr,
                [A|class],'file-id',[1|vols],[S][,V]
```

Use this set of operands to start a reader task for spooling input directly from a diskette I/O unit.

RDR Code this operand as shown.

duaddr For duaddr, specify the channel and unit address of the diskette I/O unit which VSE/POWER is to use.

A|class The operand is the same as described above for the RDR operand set.

file-id For file-id, specify the name of the diskette file to be read. This name must be identical with the file name in the HDR1 label on the diskette.

1|vols For vols, specify how many diskettes may be read. The reader task stops reading input either after it has read the specified number of diskettes or at the end of a diskette whose HDR1 label indicates that this is the last diskette of the file. You can specify any number from 01 to 99. VSE/POWER reads only one diskette if you omit the operand.

S Specify S if diskette volumes are to be checked for correct sequence. The sequence number of the first volume must be 1, that of the second volume must be 2, and so on.

V Specify V if VSE/POWER is to accept as diskette input only those files that have been verified. This causes VSE/POWER to reject any diskette input file that has not been verified. If you omit the specification, VSE/POWER accepts diskette input files verified and unverified.

```
{PSTART|S}      DEV,devname,
                subsystem[,A|,class][,PARM='string']
```

Use this set of operands to indicate to a subsystem or program in another partition that a device under control of that subsystem (program) is to be started for processing spooled output.

DEV Code this operand as shown.

devname For devname, specify the name of the device to be used. This name must be the one by which the device-owning subsystem or application program knows the device.

The specified name must be unique in the system; it must consist of up to eight alphameric characters. Do not use R000 through R250 as devname, since these are the reserved RJE userids.

subsystem
For subsystem, specify the name which the subsystem used to establish a communication path to VSE/POWER.

A|class For class, specify the class or classes as you would for any other VSE/POWER writer task.

PARM='string'
For string in PARM='string', supply a device and subsystem specific parameter string, if this is required. VSE/POWER's requirements regarding the parameter string are:

- It may not be longer than 60 characters. This includes blanks and commas that the program may need as delimiters.
- An apostrophe (') within the string must be entered as two successive apostrophes ('').

```
{PSTART|S}      {BG|Fn},[class],[A|outclass][,MT]
```

Use this set of operands if a partition is to be placed under control of VSE/POWER. If the partition is already controlled by VSE/POWER, the command acts the same way as an attention-routine START command. You receive a message if the partition is still active or not supported, or if it has a priority higher than or equal to that of the VSE/POWER partition.

BG|Fn Specify BG if the background partition is to be placed under control of VSE/POWER.

For n in Fn, specify the number of the applicable foreground partition.

class For class, specify the class(es) of the jobs that may be processed in the partition. You can specify up to four classes, each one as a letter of the alphabet.

One of the specified classes may be partition related. This is specified as one of the numbers 0 through 9 (where 0 = partition BG, 1 = partition F1, 2 = partition F2, and so on); Only one of the specified classes can be partition related.

The order in which you specify the classes will be the order in which VSE/POWER processes jobs in the partition. If you omit the operand, VSE/POWER selects for processing only jobs with a matching partition-related class assignment (only class 3 jobs in F3, for example).

A|outclass
: For outclass, specify the default class (one of the letters of the alphabet) which VSE/POWER is to use for output spooled from programs executed in the partition. VSE/POWER assigns class A if you omit the operand.

MT
: Specify MT if the involved partition is to be used for a never ending (or long running) multitask job. This causes VSE/POWER to do the following:

  - If there are no more input records for the partition, only the associated reader task is placed into the wait state. Other tasks continue and accept output from the partition.
  - Makes output from the partition available only if a request is made via an LFCB macro (output to printers with a forms control buffer) or a SEGMENT macro (output to other printers or punched output). However, the initial job input for this partition must include an LST and a PUN statement for each corresponding spool device that may be used.

  A PSTOP command for this partition cancels the multitasking specification.

```
{PSTART|S}       RJE,lineaddr,[password][,TRACE]
```

Use this set of operands if VSE/POWER is to activate the RJE,BSC function.

RJE,lineaddr
: Specify RJE,lineaddr if an RJE,BSC task is to be started.

  For lineaddr, specify the channel and unit address of the telecommunication line that is to be used by the task.

password
: For password, specify the password that is to be used by remote operators who submit jobs via the affected line. They must use this password (a string of up to eight characters) in their SIGNON commands.

TRACE
: Specify TRACE to have all I/O operations for the specified telecommunication line recorded in the available trace area.

```
{PSTART|S}        RJE,SNA
```

Use this set of operands if VSE/POWER is to activate the RJE,SNA function.

```
{PSTART|S}        PNET,nodeid,
                  [node-password],lineaddr,
                  [line-password],[TRACE][,NR]
```

Use this set of operands if VSE/POWER is to activate its PNET function using the available BSC or CTCA support.

PNET Code this operand as shown.

nodeid For nodeid, specify the name of the node to which a connection is to be made. This name must be known to VSE/POWER.

node-password
For node-password, specify the password to be supplied for communication with the other node. If you do not specify the password, VSE/POWER uses the password that was defined in the PNODE generation macro for your node.

lineaddr
For lineaddr, specify the channel and unit address of the telecommunication line or CTCA adapter that VSE/POWER is to use for communicating with the specified node.

line-password
For line-password, specify the password that is to be used for sign-on via the affected communication line. If you omit this specification, VSE/POWER uses the password defined in the BSC PLINE generation macro. However, there is no default line-password if PNET is activated using the CTCA support.

TRACE Specify TRACE if you want the line activity or any line problems to be traced. This causes all I/O operations for the specified line to be recorded in the available trace area.

NR Specify NR if automatic restart of the networking function is not to be attempted by VSE/POWER.

```
{PSTART|S}      PNET,nodeid,
                [node-password],,,[TRACE][,NR]
```

Use this set of operands if VSE/POWER is to activate its PNET function using the available SNA support.

The explanations given above for the corresponding operands in the BSC-support command format apply also to the SNA support, with one exception: TRACE.

TRACE Specify TRACE if you want all information on SEND and RECEIVE operations to be recorded in the available trace area.

```
{PSTART|S}      {CNSLTR|DUMPTR}
```

This command format is available for controlling VSE/POWER's line-trace function dynamically.

CNSLTR Specify this operand to initiate the tracing of the PNET BSC-I/O activity and writing recorded events to the system console (which amounts to a recording of the events in the system's hard-copy file).

DUMPTR Specify this operand to initiate writing the contents of the line-trace area into the dump sublibrary for the VSE/POWER partition.

Examples of the PSTART Command

PSTART RDR,00C,2
Starts a reader task for the device at 00C. Assigns class 2 to all jobs that have no class assignment.

PSTART RJE,030,NEWYORK
Starts an RJE task on line 030. NEWYORK is the password.

S RJE,047,,TRACE
Starts an RJE task on line 047, without password (indicated by ',,' following 047), and with tracing for the line.

PSTART LST,00B,AB
Starts a list task on X'00B' to print output from VSE/POWER jobs with classes A and B.

S PUN,00D,X'285'
Starts a punch task for the punch device at 00D to punch output that was spooled onto the tape mounted on the drive at 285.

PSTART F3,AB3
Places partition F3 under control of VSE/POWER; selects (for processing jobs with input classes A, B, and 3).

A class specification of AB4 in the above command would have been flagged as invalid because the partition-dependent class 4 does not match the partition identifier specified in the command (F3).

PSTART RDR,00C,,2
Starts a reader task with two buffers for the device at 00C. Assigns class A to all jobs read from the device without class assignments.

PSTART F2,Z,B,MT
Places partition F2, a multitask partition, under control of VSE/POWER. Selects (for execution) input class Z only. The default output class for this partition is B.

PSTART RDR,281,4
Starts a tape reader task to read jobs from the tape mounted on the drive at the address 281. Assigns class 4 to all jobs read from the tape without class assignment.

PSTART PNET,NODEB,,032
Starts a BSC connection to NODEB on line 032.

S PNET,NODEC,SNANODE
Starts an SDLC session with NODEC, using the node password SNANODE.

PSTOP: Stop a Task or Release a Partition

The command can be used to stop:

- A VSE/POWER task (such as reader, print-writer, spool-access support).
- Output to a device owned by a subsystem.
- A connection to another node.

It can be used to release a partition from VSE/POWER control.

Notes:

1. To restart a task after it has been stopped by PSTOP, use the PSTART command.
2. The operand KILL available in the past has been replaced by the operand FORCE.

Format of the Command

The command can be used with distinct sets of operands as shown below:

Operation	Operands
{PSTOP|P}	uraddr[,EOJ|,RESTART]
	{BG|Fn}
	tapeaddr[,EOJ]
	DEV,devname[,EOJ|,RESTART|,FORCE][,PARM='string']
	lineaddr[,EOJ|,FORCE]
	PNET,nodeid[,EOJ]
	RJE,SNA[,luname][,EOJ]
	SAS{,ALL|,connect-identifier}
	{CNSLTR|DUMPTR}

Description of Operands

```
| {PSTOP|P}        uraddr[,EOJ|,RESTART]                    |
```

Use this set of operands to stop a reader or writer task associated with a unit-record device.

uraddr For uraddr, specify the address of the unit-record device whose associated task is to be stopped. By specifying this operand alone, you cause VSE/POWER to do the following:

- For a reader task

  Stop further input processing for the affected VSE/POWER job. No entry for this job is placed into the input queue.

- For a writer task

  Stop further output processing. The affected queue entry is retained in the output queue. When the task is started again, processing of this output starts from the beginning. However, if a 3800 printer is used, processing restarts with the copy group index being processed when the PSTOP command was issued.

EOJ Specify EOJ to have VSE/POWER do the following:

- For a reader task

  Continue processing until all input for the affected VSE/POWER job has been spooled.

- For a writer task

  Continue until all currently processed printed or punched output for the affected VSE/POWER output queue entry is complete.

RESTART This operand applies only to output processing. Specify the operand if, on restart of the task, processing is to begin:

- At the next record after the last one processed before the PSTOP if a punch task was stopped.

- With the next page following the last one processed before the PSTOP if a list task was stopped. This applies also if the number-of-copies value for the output is greater than 1.

```
{PSTOP|P}        {BG|Fn}
```

Specify the identifier of the partition that is to be released from VSE/POWER control. The command is not effective until the VSE/POWER job currently processed in the partition is completed.

After a partition has been released by VSE/POWER, you can use this partition as a normal VSE batch partition. To restart the partition under control of VSE/POWER, use the PSTART command.

BG|Fn Specify BG if the background partition is to be released from control of VSE/POWER.

For n in Fn, specify the number of the applicable foreground partition.

```
{PSTOP|P}        tapeaddr[,EOJ]
```

Use this operand with PSTOP to stop a tape function.

tapeaddr
For tapeaddr, specify the channel and unit address of the drive that is being used by the affected function.

However, a task controlling input from or output to a tape cannot be stopped if this task is waiting for an operator's response. To stop the task, use a PGO tapeaddr,CANCEL command instead.

EOJ Specify EOJ if the task should stop on completion of the currently processed queue entry.

```
{PSTOP|P}      DEV,devname[,EOJ|,RESTART|,FORCE][,PARM='string']
```

Use this set of operands to tell a subsystem or program in another partition to stop further output for a device owned by that subsystem (program).

DEV Code this operand as shown.

devname For devname, specify the name of the device that is being used. This name must be the one by which the device-owning subsystem knows the device.

EOJ Specify EOJ to have the subsystem stop requesting from VSE/POWER more output to the involved device at the end of the currently processed output queue entry.

RESTART Specify RESTART if, on restart of the task, processing is to begin at the next record (or page) after the last one processed before the PSTOP. This applies also if the number-of-copies value for the output is greater than 1.

FORCE Specify FORCE if an immediate termination of the communication path to the subsystem is to be forced. The queue entry for the stopped output is retained in its queue unchanged. Use FORCE only when the subsystem does not react properly.

PARM='string'
For string in PARM='string', supply a device and subsystem specific parameter string, if this is required. VSE/POWER's requirements regarding the parameter string are:

- It may not be longer than 60 characters. This includes blanks and commas that the program may need as delimiters.
- An apostrophe (') within the string must be entered as two successive apostrophes ('').

```
{PSTOP|P}        lineaddr[,EOJ|,FORCE]
```

You can use this set of operands to stop an RJE,BSC line. How to stop a line is described under "Stopping an RJE,BSC Line" on page 107.

lineaddr
For lineaddr, specify the address of the line that is used by the RJE task which is to be stopped. Specify this address in the form cuu (channel and unit address).

Specifying this operand alone, which is called an immediate stop, causes VSE/POWER actions as follows:

1. If an RJE reader task is affected

   Stop further input processing for the affected job; no entry for this job is placed into the input queue.

   If an RJE writer task is affected

   Stop further processing of job output; however, this output is retained in the output queue. When the task is started again, processing of the output starts with the first record.

2. Stop the line in accordance with the line protocol.

3. Make an attempt to send a line-stopped message (1R02I) to the connected work station.

EOJ Specify EOJ if the task should stop on completion of the currently processed queue entry.

FORCE Specify FORCE if the named line is to be stopped at once in disregard of the line protocol. Use of the command with this operand is indicated, for example, if an immediate-stop request for a line should fail. The effect of the command on the job or output being transmitted is the same as for an immediate stop.

```
| {PSTOP|P}          PNET,nodeid[,EOJ]                          |
```

Use this set of operands if the networking activities for an individual node are to be stopped. There is no need for a PSTOP PNET,... command at normal end of operation; the PEND command initiates the necessary shut-down functions.

PNET Code this operand as shown.

nodeid For nodeid, specify the name of the node for which networking activities are to be stopped.

EOJ Specify EOJ if the PNET task is to continue processing until all activity on the line or for the session has completed to end of job. This means that all transmitter tasks continue with their current transmissions and all receiver tasks continue receiving transmissions until EOJ is reached. It is at this point that the affected transmitter and receiver tasks stop. VSE/POWER ends the communication with the node when the last transmitter or receiver task has stopped.

If you do not specify EOJ, VSE/POWER stops the active transmitter and receiver tasks immediately. This causes partially received input from the named node to be purged. Any job or output that is being transmitted when you issue the command remains in the transmission (XMT) queue.

To re-activate transmission of data, issue the PSTART PNET,nodeid,... command. This causes VSE/POWER to transmit job or output data from the beginning of the job or output.

```
| {PSTOP|P}          RJE,SNA[,luname][,EOJ]                     |
```

Use this set of operands to stop certain or all RJE,SNA activities.

RJE,SNA Code this as shown.

By specifying RJE,SNA alone (without the delimiting comma after SNA), you cause VSE/POWER to deactivate the ACF/VTAM interface immediately.

luname Specify this operand (logical unit name) to have VSE/POWER stop a certain session. For luname, supply the logical unit name associated with this session.

The operand allows you to log off individual sessions of multiple logical-unit work stations without logging off the entire work station.

EOJ The operand ensures orderly termination. By specifying EOJ, you cause VSE/POWER to continue processing the current queue entry (or the affected RJE session) until end-of-job occurs.

If an RJE task waits for new forms to be mounted at an RJE work station, VSE/POWER stops the line immediately.

```
{PSTOP|P}        SAS{,ALL|,connect-identifier}
```

Use this set of operands to stop one or all of the currently active spool-access support tasks.

SAS Code this operand as shown.

ALL Specify ALL if all spool-access support tasks are to be stopped.

connect-identifier
For connect-identifier, specify the VSE/POWER assigned connection identifier of the spool-access support task that is to be stopped. VSE/POWER displays this identifier, a number, in response to a

```
PDISPLAY A,SAS
```

command; you find this number on the display line immediately behind the task name.

```
{PSTOP|P}        {CNSLTR|DUMPTR}
```

This command format is available for controlling VSE/POWER's line-trace function dynamically.

CNSLTR Specify CNSLTR to stop the tracing and recording (to the console) of PNET BSC-I/O activity.

DUMPTR Specify DUMPTR to stop writing the contents of the line-trace area into the dump sublibrary for the VSE/POWER partition.

Examples of the PSTOP Command

PSTOP 00E,EOJ
Stops the list task operating on 00E after it has completed printing the list output of the currently processed VSE/POWER list-queue entry. When a 3800 printer was used, printing stops with the copy group index being processed when the PSTOP command was issued.

P 030
Stops immediately the RJE task operating on line 030. If this task is reading, the input records already spooled for the current RJE job are lost. If the task is writing, output will start from the first record of that output when the task is restarted.

PSTOP 00D,RESTART
Stops the punch task operating on 00D. When the task is started again, processing of the stopped queue entry continues at the point of interruption.

PSTOP F3
Releases partition F3 from control of VSE/POWER after the VSE/POWER job currently processed in that partition is finished.

P RJE,SNA,EOJ
Stops all RJE,SNA sessions as soon as processing of all jobs in all sessions is finished.

PSTOP RJE,SNA,SES1
Stops immediately the SNA session with the name SES1.

PSTOP 281
Stops immediately the task which uses the specified tape drive (at the address 281) for input or output.

P PNET,NODEB,EOJ
Stops transmission from and to NODEB as soon as all data for the currently processed queue entry (or job) has been transmitted.

PXMIT: Route a Command

This command routes another command either to another node within the network or to a subsystem (or program) that writes spooled output to a device under its control.

By routing a VSE/POWER command to another node, you can initiate processing or have VSE/POWER display information about data streams residing at that node. By routing a command to a subsystem, you can exercise additional control over the handling of output data by this subsystem.

You may submit a PXMIT command at the system console, at a locally attached remote operator station, or via a spool-access (cross-partition) communication path.

Format of the Command

Operation	Operands
{PXMIT|X}	{nodeid|DEV,devname},command

Description of Operands

nodeid This operand is valid only if the PNET function is active.

For nodeid, specify the name of the node that is to receive the command. This name must be known to VSE/POWER of your system.

VSE/POWER rejects the command if no connection to or session with the named node exists.

DEV,devname
Use these two operands if a command is to be routed to a subsystem (program) that writes spooled output to a device under the control of that subsystem.

For devname, specify the name of the device to which the subsystem writes the output affected by the routed command. The name you specify is the one by which the device-owning subsystem knows the output device.

command Supply the command (including its operands, if any) that is to be routed. VSE/POWER routes the supplied command unchanged. The command to be routed and the PXMIT (or X) command (including the routing indication) cannot be longer than 72 characters.

Supply the command to be routed with the operation in short form (P for PSTOP, for example); this leaves more space for operands which you want to include.

To another node within the network, you can route only those commands for which your node has authorization. You receive a message from VSE/POWER if you supply an unauthorized command. For information on command authorization and the commands that can be issued, refer to the publication VSE/POWER Networking User's Guide.

To a subsystem handling spooled output, you can pass any command that would be accepted and processed by this subsystem. VSE/POWER checks the passed command only for a command verb (or operation code) of no more than 16 characters followed by at least one blank. The passed command may contain as many optional operands as may fit into the total length allowed for a PXMIT command (72 bytes); the format of these operands is subsystem defined.

Examples of the PXMIT Command

```
PXMIT NODEA,D XMT,TNODE=NODEB
```
Displays all entries from the transmission queue of node NODEA which are destined for NODEB.

```
X NODEA,PSTART RJE,030
```
Starts an RJE task on line 030 on node NODEA.

```
X DEV,PLOT1,SET LOGDEST=3244
```
Passes the command SET LOGDEST=3244 to the subsystem that controls the device named PLOT1.

CHAPTER 5. VSE/POWER JOB ENTRY CONTROL LANGUAGE STATEMENTS

To define a job to VSE/POWER and have that job logged in the system, the VSE/POWER job entry control language (JECL) or the VSE job control language (JCL) may be used. Examples are shown in Figure 14.

If JCL is used, VSE/POWER logs a job by the name specified in the // JOB statement; this statement, together with the /& statement, delimits the job. VSE/POWER generates a /& statement if no such statement immediately precedes a // JOB statement.

If JECL is used, the * $$ JOB and * $$ EOJ statements delimit a job. VSE/POWER uses the name specified in the * $$ JOB statement to log this job.

A job defined to VSE/POWER by using JECL may comprise several VSE jobs. One VSE job may be defined as two or more VSE/POWER jobs. In this case, either of the following is required:

- The partition in which the VSE job executes must be started with the MT option of the PSTART command, or
- When the VSE/POWER job boundary is reached, job control must be in the partition in which the VSE job executes.

Because JECL statements begin with an asterisk-blank character combination, VSE job control treats these statements as comments when VSE/POWER is not in operation. For a writer-only partition, this means that LST and PUN statements are logged twice. If this is not desired, have a // OPTION NOLOG statement precede the JECL statements.

VSE/POWER assigns a number to each job that it logs. This enables VSE/POWER to distinguish between jobs for which identical names were specified. To control the processing of a job, your operator must use this job's name and, possibly, the assigned number in the applicable command.

Once a job has been logged by VSE/POWER, that job is referred to as a VSE/POWER job or as a queue entry.

The first list (LST) and the first punch (PUN) output from a VSE/POWER job has the same number as the input job. However, any subsequent LST or PUN output is a separate queue entry and has a unique job number. This aids in manipulating the processing of VSE/POWER jobs.

All count-driven output segments of a job have the same number as the first segment. Yet, for accounting purposes, all output contains the job number of the reader queue entry when a record is placed into the account file.

```
  VSE Job Stream                    Comments

// JOB ONE                        VSE job without JCL changes.
 ... ... ...
// EXEC JOBSTEPA
/*
/&

// JOB TWO                        No * $$ JOB or * $$ EOJ required for
 ... ... ...                      LST or PUN statements.
// EXEC JOBSTEPB
 ... ... ...
* $$ PUN CLASS=X
 ... ... ...
// EXEC JOBSTEPC
/*
/&

* $$ JOB THIRD                   Optional VSE/POWER JECL.
// JOB THREE                     Required if statements from SYSIN
 ... ... ...                     include a // JOB statement.
// EXEC JOBSTEPD
/*
/&
* $$ EOJ

* $$ JOB FOURTH                  No * $$ EOJ required if
// JOB FOUR                      VSE/POWER job is followed
 ... ... ...                     by * $$ JOB statement.
// EXEC JOBSTEPE
 ... ... ...
/*
// EXEC JOBSTEPF
/*
/&

* $$ CTL CLASS=B                 Default class changed to B.

* $$ JOB FIFTH
// JOB FIVE                      No * $$ EOJ required.
 ... ... ...
// EXEC JOBSTEPG
/*
/&
```

Figure 14 (Part 1 of 2). Examples of the Use of VSE/POWER JECL

```
  VSE Job Stream                Comments

* $$ JOB SIXTH                 Multiple VSE jobs
// JOB SIX                     in one VSE/POWER job.
 ... ... ...                   (For this, both statements
// EXEC JOBSTEPH               * $$ JOB and * $$ EOJ are
/*                             required.)
/&

// JOB SEVEN
 ... ... ...
// EXEC JOBSTEPI
/*
/&
* $$ EOJ

* $$ CTL CLASS=A               Default class reset to A.

// JOB EIGHT
 ... ... ...
// EXEC JOBSTEPL               VSE/POWER generates the missing
/*                             /& statement.

// JOB NINE                    Multiple LST outputs per job.
* $$ LST FNO=8X1               (The second report is segmented.)
 ... ... ...
// EXEC JOBSTEPM
* $$ LST JSEP=2,RBS=100
 ... ... ...
// EXEC JOBSTEPN
/*
/&
```

Figure 14 (Part 2 of 2). Examples of the Use of VSE/POWER JECL

Correction of JECL Errors

Any * $$ JOB, * $$ LST, and * $$ PUN statement which VSE/POWER finds to be in error can be corrected when the associated job is being executed. The central operator receives a message indicating the error. The operator may then do one of the following:

- Correct the statement.

  The operator enters the correct JECL statement. The full corrected statement must fit on one entry line. VSE/POWER processes this statement and continues executing the job.

- Cancel the job.

  The operator enters FLUSH in response to the message. VSE/POWER starts processing the next job in the associated queue.

- Ignore the error.

  The operator enters a null line by pressing END/ENTER. VSE/POWER then selects default values for the JECL statement operands in error and continues processing.

Format of JECL Statements

For the symbols used to show the formats of the JECL statements, refer to the section "Conventions for Showing Command and Statement Formats" on page 15.

The various fields of a JECL statement are briefly discussed below:

Columns	Contents and Description
1-4	* $$ - The characters indicate that this is a VSE/POWER JECL statement.
5-71	Operation code - The code, a mnemonic of the desired operation, can follow directly the second dollar sign in column 4, or this mnemonic can be separated from the dollar sign by one or more blanks. At least one blank must separate the operation code from the first (or only) operand. Operand(s) - The operation code may be followed by one or more operands, separated by commas. A blank ends the operand field. No operand may contain embedded blanks, except if the specification is enclosed in a pair of apostrophes. Example: USER='FIRST WEEK'. If a keyword operand contains two or more specifications, then these specifications must be separated by commas and enclosed in parentheses. For keyword operands, do not code a comma in front of the first operand. For example, if you omit the BURST operand in the * $$ LST statement, do not include a comma before the CHARS operand. If an operand occurs more than once, VSE/POWER uses the specification supplied with the operand repeated last. If the operands to be specified exceed column 71, the field may be continued. In this case, the following rules apply:

1. The field can be broken after a comma that separates two operands, and it can be continued on the next card (or line) starting at one of columns 6 through 16.
2. The statement that is being continued must have a non-blank character in column 72.
3. Columns 1 to 4 of each continuation statement must contain the * $$ identifier.
4. An operand field that is continued must contain at least one operand or a comma.

Comments - A comment must be separated from the last operand by at least one blank; it may include embedded blanks. A comment may be broken off anywhere and continued on the next card or line. For the continuation of a comment, the same rules apply as for the continuation of operands, except that a continued comment may begin in column 5 already.

If no operand precedes a comment, this omission is to be indicated by a comma preceded and followed by at least one blank.

72 Continuation column - Must contain a non-blank character if the next card or line is to contain one or more operands of the statement or part of a comment. If the statement does not exceed 71 characters, column 72 must be blank.

73-80 Sequence field - The field may contain up to eight characters of optional information used for control-statement identification.

Note: VSE/POWER converts the VSE // JOB and // EXEC statements to uppercase characters before spooling.

Overview of JECL Statements

Figure 15 on page 242 lists JECL statements in alphabetic order. It gives a short explanation of the purpose of each of them. This overview is followed by a detailed description of the statements and their operands, again in alphabetical order.

JECL Statement	Function
* $$ CTL	Assigns a new default input class to VSE/POWER jobs.
* $$ DATA	Inserts data from the reader queue into a library member after this member was retrieved for inclusion into a VSE/POWER job.
* $$ EOJ	Indicates the end of a VSE/POWER job.
* $$ JOB	Indicates the beginning of a VSE/POWER job and specifies the routing of jobs, output, and notify messages.
* $$ LST	Provides handling information for printed output; routes list output to a node.
* $$ PUN	Provides handling information for punched output; routes punch output to a node.
* $$ RDR	Inserts a diskette file into the input stream.
* $$ SLI	Inserts data from an accessible library.
* $$ /*	Indicates the end of a VSE job step (used with the SLI statement only) - see (1) below.
* $$ /&	Indicates the end of a VSE job (used with the SLI statement only) - see (1) below.
/*$SLI	Indicates end of input data for an SLI member - see (2) below.

(1) Described under "* $$ SLI: Adding a Library Member to a Job Stream."
(2) Described under "* $$ DATA: Inserting Data into an SLI Library Member."

Figure 15. Overview of JECL Statements

The statements

```
* $$ CTL
* $$ JOB
* $$ EOJ
```

have to be presented when the system's job control program is in control. All other JECL statements may be included anywhere in the input stream.

More than one * $$ LST or * $$ PUN statement is allowed per VSE/POWER job.

The statements * $$ JOB and * $$ EOJ must be used in a writer-only partition (a partition that uses VSE/POWER only for spooling output). Other JECL statements are used in a writer-only partition with restrictions as noted below.

- The following statements are treated as comments:

  ```
  * $$ RDR
  * $$ CTL
  * $$ SLI
  * $$ DATA
  ```

- * $$ LST and * $$ PUN statements are recognized either at job control time or when user-generated and written to SYSLOG.

- * $$ JOB operands that are not meaningful are ignored.

- Output produced outside the bounds of a VSE/POWER-defined job is ignored. This implies that complete VSE jobs (from // JOB to /&) must be contained in a VSE/POWER job.

In a writer-only partition, JECL statements contained in a user data stream are considered to be data. This allows SLI library members containing JECL statements to be cataloged.

Note: Throughout this chapter, information given for the IBM 3800 Printing Subsystem apply also to the IBM 3200 Printing Subsystem, if not specifically stated otherwise.

* $$ CTL: Assign a Default Input Class

The statement allows you to assign a new default input class to VSE/POWER jobs which are not controlled by JECL or for which no class was specified in the * $$ JOB statement.

An * $$ CTL statement may be placed at any VSE/POWER job boundary. The statement overrides all input default class specifications presently effective. It remains in effect until the next * $$ CTL statement is encountered or the reader task is stopped with a PSTOP command.

The statement is ignored in a writer-only partition.

Format of the Statement

Operation	Operands
* $$ CTL	[CLASS={A|class}]

Description of Operands

CLASS={A|class}

For class, specify the input class to which all subsequent jobs are assigned if they do not contain a class specification in their * $$ JOB statement. The CLASS operand can be used in order to group jobs that require the same I/O configuration, partition, or other resource for execution.

Specify, for class, any alphabetic character from A to Z or a one-digit number from 0 through n. For more details about the assignment of input classes, see the description of the operand CLASS=class under "* $$ JOB: Mark the Start of a VSE/POWER Job" on page 249.

The class you specify in this statement overrides the class specified in the PSTART command or the VSE/POWER default (class A), whichever applies.

* $$ DATA: Insert Data into an SLI Library Member

The statement allows you to insert data into a library member which VSE/POWER retrieves in accordance with a request by an * $$ SLI statement. A member so retrieved is referred to as an SLI member.

Format of the Statement

Operation	Operands
* $$ DATA	name

Description of the Operand

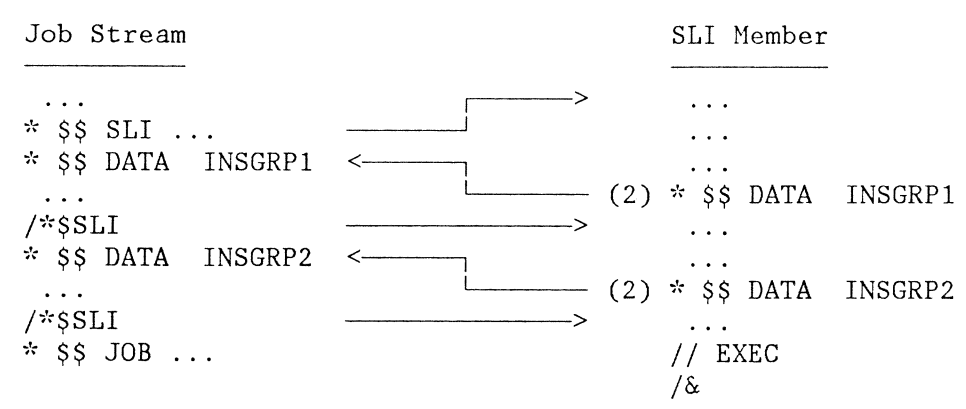
name — For name, specify the name of the corresponding * $$ DATA statement in the retrieved library member where data is to be inserted. The name can be from one to eight alphameric characters long; the first character of this name must be alphabetic.

Use of the Statement

VSE/POWER requires matching * $$ DATA statements in both the job stream and in the SLI member. In the SLI member, an * $$ DATA statement marks the insertion point for data from the job stream. In the job stream, the matching * $$ DATA statement marks the beginning of the data that is to be inserted.

The statement is ignored in a writer-only partition.

Figure 16 on page 246 shows how the required * $$ DATA statements relate to an SLI statement.



(1) * $$ SLI ...
Causes VSE/POWER to switch reading input from the job stream to the specified (sub)library.

(2) * $$ DATA INSGRPn
When VSE/POWER finds this statement in the SLI member, it switches reading input back to the job stream. It expects to find a matching * $$ DATA statement at the point where it originally switched reading from the job stream to the (sub)library.

(3) /*$SLI
Marks the end of the data that is to be inserted. The statement is not included in the SLI member (as opposed to a /* or a /& statement).

Figure 16. Use of * $$ DATA Statements

Data to be inserted into an SLI member cannot include any of the following:

Another * $$ DATA statement
An * $$ SLI statement

Data to be inserted must be terminated by either:

- The VSE job-step or job delimiter (/* or /&, respectively), or
- The /*$SLI statement if the end of insertion data is not also an end of job step or an end of job condition.

Update statements ($SLI in columns 73-76) and * $$ DATA statements may be used together. However, they must appear in the job stream in the same order as the corresponding images appear in the retrieved library member. For more information about update statements, refer to "* $$ SLI: Add a Library Member to a Job Stream" on page 284.

* $$ EOJ: Mark the End of a VSE/POWER Job

The statement marks the end of a VSE/POWER job. It is normally placed at the end of a VSE job, but may be placed anywhere in a VSE job stream.

The statement is required in a writer-only partition.

Format of the Statement

Operation	Operands
* $$ EOJ	

The statement has no operands.

* $$ JOB: Mark the Start of a VSE/POWER Job

The statement may be placed anywhere in a VSE job stream to indicate the beginning of a VSE/POWER job. It specifies job-handling information to VSE/POWER.

If the statement is omitted, VSE/POWER jobs are delimited by the // JOB and /& job control statements; the VSE/POWER job name is then the same as the VSE job name.

If a VSE/POWER job consists of two or more VSE jobs, these jobs should be delimited by a /& statement at the end. Reason: if a VSE job fails, the job stream is ignored up to the next VSE-job delimiter (// JOB or /&) or VSE/POWER-job delimiter (* $$ JOB or * $$ EOJ statement).

The * $$ JOB statement is required in a writer-only partition, in which case no continuation statement is allowed. It must follow immediately the * $$ EOJ statement of the preceding VSE/POWER job.

An * $$ JOB statement in error can be corrected when the job is being executed. For more information, refer to "Correction of JECL Errors" on page 239.

Format of the Statement

Operation	Operands
* $$ JOB	[JNM={AUTONAME|jobname}]
	[,BLDG=number]
	[,CLASS={A|class}]
	[,DEPT=number]
	[,DISP={D|disposition}]
	[,DSKTIN=cuu]
	[,LDEST={nodeid|(nodeid,userid)}]
	[,NTFY={YES|(nodeid,userid)}]
	[,PDEST={nodeid|(nodeid,userid)}]
	[,PRI=priority]
	[,PROGR=name]
	[,PWD=password]
	[,ROOM=number]
	[,SYSID={n|N}]
	[,USER=user-information]
	[,XDEST={nodeid|(nodeid,userid)}]

Description of Operands

JNM={AUTONAME|jobname}

For jobname, specify the name by which the job and its associated queue entries are to be known to VSE/POWER. If a job contains several VSE jobs, they are all logged as one VSE/POWER job by the job name which you specify.

You can specify a name consisting of two to eight alphameric characters in any combination. For a definition of alphameric characters refer to page 17.

Do not use any of the following as a job name:

ALL	HOLD	RJE
DEV	LOCAL	STATUS
FREE	PNET	VIO

A job name such as this may conflict with specifications in some VSE/POWER operator commands.

BLDG=number

Specifies the programmer's building number to be placed on the job separators (see also the PROGR operand). You can specify a number of up to eight alphameric characters.

CLASS={A|class}

For class, specify the class that is to be assigned to the job in the input queue. You can specify this class as a letter of the alphabet or as one of the numerals 0 through 9.

You can use this specification for grouping jobs that require the same I/O configuration, partition, or other resource for execution. VSE/POWER ignores the operand in a writer-only partition.

Classes 0 through 9 are the partition-dependent classes. Jobs with class 0 are selected for execution only in the background partition, jobs with class 1 in partition F1, and so on.

If you do not specify a class, VSE/POWER uses the class you specified by way of an * $$ CTL statement. If no such statement was given, VSE/POWER uses the class specified in the applicable PSTART command. If no class is specified at all, VSE/POWER assigns class A.

DEPT=number

Specifies the programmer's department number to be placed on the job separators (see also the PROGR operand). You can specify a number of up to eight alphameric characters.

DISP={D|disposition}

For disposition, specify how the VSE/POWER job is to be handled in the reader queue. Your specification may be one of the following:

D Delete after processing. The job is automatically scheduled according to its class and priority. When the job is complete, VSE/POWER deletes the queue entry from the reader queue.

H Hold. The job remains in the reader queue; it is not dispatched for processing until the operator either:

- Changes the job's disposition to K or D by way of a PALTER command, or
- Issues a PRELEASE command for the job.

K Keep after processing. The job is automatically scheduled according to its class and priority. When the job is complete, VSE/POWER retains the job in the reader queue with a disposition of L.

L Leave in queue. The job remains in the reader queue. It is not processed until the operator either:

- Changes the job's disposition to K or D by way of a PALTER command, or
- Issues a PRELEASE command for the job.

The specification of a disposition is ignored in a writer-only partition. For further information on dispositions, see also Appendix A on page 309.

DSKTIN=cuu

The operand tells VSE/POWER that SYSIN data is to be read by your program via the available diskette access support (DTFDU for an assembler language program). If you specify this operand, VSE/POWER can pass to your program SYSIN records of a length of up to 128 bytes.

For cuu, specify the channel and unit address of the diskette I/O unit that is to be used.

If both SYSIN data and normal (other) diskette input data are to be read by your program, you have to use different diskette I/O units. You specify the channel and unit address for non-SYSIN input data in the operand DEV=diskette-cuu of the * $$ RDR statement.

The operand is ignored for a writer-only partition.

LDEST={nodeid|(nodeid,userid)}
The operand specifies where to route the list output of this VSE/POWER job. This specification may be overridden by a specification in the DEST operand of a subsequent * $$ LST statement. If you do not specify this operand, VSE/POWER routes the output to the point of origin.

For nodeid, specify either of the following:

- The name of the node of destination as defined to VSE/POWER by way of the PNODE generation macro.
- An asterisk (*) if the output is to remain at the execution node; example: LDEST=(*,userid).

For userid, specify the identifier of the user who is to get the output. For a definition of userid, refer to "Conventions for Showing Command and Statement Formats" on page 15.

NTFY={YES|(nodeid,userid)}
This operand specifies to where VSE/POWER is to send notify messages. VSE/POWER issues such messages if one of the following occurs:

- A locally or remotely executing job is finished.
- An output has been received for processing.
- A job or an output has been transmitted by the network to the next system in the network.

Specify:

NTFY=YES
If you want the originator to receive notify messages. If your job is read in by a local unit record device, VSE/POWER sends the notify messages to the local console.

NTFY=YES can be specified for VSE/ICCF users.

(nodeid,userid)
If notify messages for the job are to be sent to a specific user.

For nodeid, specify the name of the node to which the user specified for 'userid' is linked. If a user at your own (VSE) node is to receive notify messages, you may omit the node name and indicate this omission by a comma; for example: (,userid). If a user at the executing node is to receive notify messages, you may replace the executing node name by an asterisk (*); for example: (*,userid).

A nodeid specification overwrites the originating node name.

For userid, specify the identifier of the user who is to receive the messages. For a definition of userid, refer to "Conventions for Showing Command and Statement Formats" on page 15.

Specify 0 for userid if the messages are to be routed to the central operator.

If your VSE system operates under VM/SP, you may want messages to be routed to CMS users not under control of your VSE. This is possible only if:

1. Your VSE system includes the networking support, and
2. VSE/POWER can link to RSCS of VM/SP.

PDEST={nodeid|(nodeid,userid)}

The operand specifies to where VSE/POWER is to route punch output of your VSE/POWER job. This specification may be overridden by a specification in the DEST operand of a subsequent * $$ PUN statement. If you do not specify this operand, VSE/POWER routes the output to the point of origin.

For nodeid, specify the name of the node of destination as defined to VSE/POWER by way of the PNODE generation macro. If the output is to remain at the execution node, you may specify an asterisk (*). Example: PDEST=(*,userid).

For userid, specify the identifier of the user who is to get the output. A definition of userid, is given under "Conventions for Showing Command and Statement Formats" on page 15.

PRI=priority

The operand specifies the priority that is to be assigned to the job in the input and output queues. The specification is ignored in a writer-only partition; it may be overridden in an * $$ LST or * $$ PUN statement.

For priority, specify a one-digit number from 0 to 9, where 9 is the highest priority. If you omit the operand, VSE/POWER assigns the default priority as defined by the PRI operand of the POWER generation macro.

Jobs are dispatched on a first-in-first-out basis within priority and in the sequence of classes as specified in the PSTART command.

PROGR=name

Specifies the name of the person (programmer) responsible for the job. This name is printed on the separator pages. You can specify a name of up to 20 alphameric characters. If the name contains special characters, such as a hyphen or a blank, you must enclose the name within a pair of apostrophes. An apostrophe within the name must be entered as two apostrophes.

PWD=password
: Specify this operand if you wish to have the job protected by a password. The password you define must be specified if anyone other than the central operator wants to manipulate the job while it is in the local queue (for example, have the job processed in another partition).

  If you do not define a password, VSE/POWER sets the password field to binary zeros, and any access other than by the local operator is denied.

  Note: The default password setting for access requests from a program is different from this setting for a locally submitted job. To allow a program to access a locally submitted job (or its output), specify a password. In addition, ensure that the accessing program presents this password in its service request (a GET service request if the spool-access support is used, a GETSPOOL macro request if the SPOOL-macro support is used).

  A password must be defined as an alphameric string of up to eight characters; it can be specified also in an * $$ LST or * $$ PUN statement.

ROOM=number
: Specifies the programmer's room number to be placed on the job separators (see also the PROGR operand). You can specify a number of up to eight alphameric characters.

SYSID={n|N}
: This operand applies to shared-spooling. The specification in this operand can be overridden by an * $$ LST or * $$ PUN statement.

  Omit the operand if your job is to be processed on any of the sharing systems.

  Specify:

  SYSID=n
  : If your job is to be processed on a certain one of your sharing systems.

    For n, give the number with which the system's VSE/POWER was initialized (by SYSID=n in the POWER generation macro).

  SYSID=N
  : If the job may be processed on any of the sharing systems.

USER=user-information
: Up to 16 bytes of user information may be specified. If JLOG=YES was specified during VSE/POWER generation, the specified user information is printed on SYSLOG when execution of the

job begins. If applicable, VSE/POWER includes the specified user information in account records and prints this information onto separator pages.

For user-information, specify an alphameric string. Embedded blanks within this string are allowed only if the string is enclosed by a pair of apostrophes.

XDEST={nodeid|(nodeid,userid)}
The operand specifies to where VSE/POWER is to route the job for processing.

For nodeid, specify the name of the applicable node as defined to VSE/POWER.

For userid, specify the identifier of the destination user, a VSE system with VSE/POWER and operating under VM/SP, for example. However, any other control information that you specify in your * $$ JOB statement is not passed on to VSE/POWER on the system identified by your userid specification.

The operand is ignored for a writer-only partition.

* $$ LST: Define List-Output Attributes

The statement defines the attributes of list output as spooled by VSE/POWER. It may specify the routing of this output.

The specified attributes remain in effect until one of the following occurs:

A subsequent * $$ LST statement for the same spool device.
A SEGMENT macro for the same printer.
A VSE/POWER end-of-job (* $$ EOJ) statement.

Any number of * $$ LST statements may be included in a VSE/POWER job. Inclusion of * $$ LST statements does not require * $$ JOB and * $$ EOJ statements to be used.

For each * $$ LST statement within a VSE/POWER job, VSE/POWER creates a separate list-queue entry and assigns a job number to this entry. This is referred to as data-driven output segmentation. The first list entry of the job gets the same job number as the job's RDR-queue entry. The second and subsequent list entries have unique job numbers.

If your * $$ LST statement includes IBM 3800 related operands and the output is directed to a printer of a different type, then VSE/POWER ignores these operands.

If you do not include an * $$ LST statement, VSE/POWER uses:

- For a printer other than IBM 3800, the default values set during VSE/POWER generation or defined earlier by way of an * $$ JOB statement or a PSTART command, whichever applies.

- For an IBM 3800, the default printer-setup values defined for this printer by the VSE SETDF command (for more information about the 3800 and the use of these values, see the DOS/VS IBM 3800 Printing Subsystem Programmer's Guide).

An * $$ LST statement in error can be corrected when the job is being executed. For more information, refer to "Correction of JECL Errors" on page 239.

Format of the Statement

Operation	Operands
* $$ LST	[BURST={Y|N}]
	[,CHARS=(tablename[,tablename...])]
	[,CLASS=class]
	[,CMPACT={name|NO}]
	[,COPY={1|number-of-copies}]
	[,COPYG=(groupvalue[,groupvalue...])]
	[,DEST={nodeid|(nodeid,userid[,subsystem])}]
	[,DFLT={Y|N}]
	[,DISP={D|disposition}]
	[,FCB=phasename]
	[,FLASH=([overlayname][,count])]
	[,FNO=formnumber]
	[,JSEP={n|(n,option)|(,option)}]
	[,LST=lstaddr]
	[,LTAB=linetab]
	[,MODIFY=(copymodname[,tablename])]
	[,PRI=priority]
	[,PWD=password]
	[,RBC=norbc]
	[,RBM=(norbm1,norbm2)]
	[,RBS=norbs]
	[,REMOTE=remid]
	[,SYSID={n|N}]
	[,TADDR={tapeaddr|(tapeaddr,mode)}]
	[,UCS=(phasename[,option])]
	[,USER=user-information]

You can define your own JECL output operands in the autostart DEFINE statement and specify them in the * $$ LST (or * $$ PUN) statement. For more information, see "DEFINE: Specify User-Defined Output Operands" on page 294.

Description of Operands

BURST={Y|N}

The operand applies to IBM 3800 only. It specifies whether the output to be printed is to go to the (optional) burster-trimmer stacker. Specify:

Y If the output is to be burst into separate sheets.

N If the output is to be continuously fan-fold.

If you omit the operand, VSE/POWER uses the default paper-threading request (which may have been set by way of a SETDF

command). If you specify also DFLT=N, VSE/POWER uses BURST=N as the default.

If you direct your output to the burster-trimmer stacker, stacked sheets of each output are automatically separated from those of the preceding output. Copies and copy groups are also separated.

CHARS=(tablename[,tablename...])
The operand applies to IBM 3800 only. It names the character-arrangement table(s) to be used for the printing of the output file.

For tablename, specify the one- to four-character phase name of the table (without the system-assigned prefix XTB1). You can specify up to four table names.

For more information on character-arrangement tables and their use, see the DOS/VS IBM 3800 Printing Subsystem Programmer's Guide.

Note: If you code this operand but not the UCS operand and the output is directed to an impact printer with the UCS feature, then VSE/POWER uses the first (or only) character-arrangement table name as the UCB-image name.

CLASS=class
The operand defines the class that is to be assigned to the printed output after it has been spooled by VSE/POWER. It may be specified together with FNO=formnumber to group the same type of printed output.

The operand is ignored if your output is to be spooled to tape.

For class, you can specify any letter of the alphabet.

If you omit the operand, VSE/POWER assigns the class that was specified in the PSTART command for the partition used to execute your program.

CMPACT={name|NO}
For name, specify the name of the compaction table that is to be used. The name is the one you used in the name field of the PCPTAB macro when the table was generated.

Specify NO if no compaction is to be used for this job.

If you omit the operand, VSE/POWER uses the default compaction table specified in the PRMT macro, if this is applicable.

For more information on data compaction, refer to Appendix A.

COPY={1|number-of-copies}
: The operand specifies the number of copies to be printed from the list queue entry. The operand is ignored if the output is spooled to tape. To get two or more copies of output spooled to tape, have the tape reprinted the desired number of times.

  For number-of-copies, you can specify any number from 0 to 255. If you specify 0, VSE/POWER prints one copy.

COPYG=(groupvalue[,groupvalue...])
: The operand applies to IBM 3800 only. It specifies how the printed copies of the output file are to be grouped. No copy grouping occurs if the operand is omitted; The operand is ignored if your program's output is spooled to tape.

  For groupvalue, specify the number of copies of each individual page that is to be printed in a group before the next page is going to be printed. VSE/POWER transfers the entire output file for each specified group value.

  You can specify up to eight group values. Neither a single group value nor the sum of the specified values can exceed 255.

DEST={nodeid|(nodeid,userid[,subsystem])}
: Any specification in this operand overrides the specification in the LDEST operand of the * $$ JOB statement.

  If you specify REMOTE=remid after this operand, your specification in the DEST operand is overwritten; VSE/POWER then sets the node name to that of your own node and the specified user identifier to the identifier specified for remid in the REMOTE operand.

  Specify for:

  nodeid
  : One of the following:

    - The name of the node to which the list output of this VSE/POWER job is to be routed. By specifying only a node name, you indicate that this output is to be processed on a printer at the central location of that node.

    - An asterisk (*) if the destination of the output is also the job execution node.

    - A null string if the output is destined for a user at the local node; to do this, simply specify DEST=(,userid).

userid
One of the following:

- The applicable user identifier. For a definition of userid, refer to "Conventions for Showing Command and Statement Formats" on page 15. If you specify a user identifier, VSE/POWER cannot print the associated output on a local printer.

- ANY. This indicates that VSE/POWER may make the output available to any user that requests this output.

  Specification of ANY does not allow you to manipulate the queue entry unless your authorization allows it.

subsystem
Applies if, for nodeid, you specified the name of an MVS node.

For subsystem, give the name of the external writer that is to process the spooled output.

DFLT={Y|N}
The operand applies to IBM 3800 only. It specifies whether the IBM 3800 Printer is to be set up with the defaults that were defined by the operator in a SETDF command. Specify:

Y If the defaults set up by an earlier SETDF command are to be used. VSE/POWER then uses these defaults if the corresponding operand is not specified in your * $$ LST statement. Defaults can be set by way of the SETDF command for the following operands: BURST, CHARS, FCB, FLASH, FNO, and MODIFY.

N If the defaults set by an earlier SETDF command are not to be used. In this case, VSE/POWER uses, if necessary, the IBM 3800's hardware defaults instead.

DISP={D|disposition}
For disposition, specify how VSE/POWER is to handle your list output. Your specification may be one of the following:

D Delete after processing. Your output is written to a printer according to its class and priority. When the printing of the output is complete, VSE/POWER deletes this output from the list queue.

H Hold. Your output remains in the list queue; it is not written to a printer by VSE/POWER until the operator changes the disposition to D or K by a PALTER command or issues a PRELEASE command for the output.

K Keep after processing. Your output is automatically written to a printer according to its class and priority. When the printing of the output is complete, this output is retained in the list queue with a disposition of L.

L Leave in queue. Your output remains in the list queue; it is not written to a printer until the operator changes the disposition to D or K by a PALTER command or issues the PRELEASE command for the output.

N Output without spooling. This causes output spooling to be suppressed. The output is written to an output device under control of the program which produces the output. In this case, VSE/POWER ignores all other operands of your * $$ LST statement, except the LST operand.

If, however, the spooled device is not available during program execution, VSE/POWER forces a disposition of D for the output and informs you by a message.

T Spool the output to tape. To start printing the output, the central operator must issue a PSTART command for this output.

If you specify DISP=T, the following operands have no effect:

CLASS	DEST	RBS
CMPACT	PRI	REMOTE
COPY	PWD	SYSID
COPYG	RBC	

For further information on dispositions, see also Appendix A on page 309.

FCB=phasename

For phasename, specify the name of the FCB-image phase which VSE/POWER is to use for printing the related job output. VSE/POWER ignores an LTAB=linetab specification if you specify this operand. The named phase must be cataloged in a sublibrary defined as accessible from the VSE/POWER partition.

The first four characters of the specified name must be $$$$ if your location makes use of a device-independent FCB-image. For more information about a device-independent FCB-image, see "Device-Independent FCB Image Names" on page 42. These four characters must be FCB1 if your output is directed to an IBM 3800.

When spooling the related output, VSE/POWER uses the named FCB image for the control of line-skip and page-eject signals to the job's application program(s). Your program should start its list output with a skip-to-channel 1. If it does not, VSE/POWER

inserts a skip-to-channel 1 at print time, provided SKIP=YES was specified in the autostart SET statement for start-up, or the writer task was started with the option SKIP.

For printing the related output on a line printer with an FCB, VSE/POWER loads the named phase into the printer's FCB.

For printing the output on a line printer without an FCB, your operator must mount a carriage control tape that matches the FCB image used by VSE/POWER for spooling this output. Your operator may have to exchange the tape again on completion of the output.

If you omit this operand for output directed to a printer with an FCB, then VSE/POWER:

1. Controls the spooling of this output in accordance with the LTAB specification in the POWER generation macro.
2. Loads the printer-specific default FCB image into the printer's FCB when this output is being printed.

To ensure correct output, verify that your specification in the LTAB operand of the POWER generation macro matches the default FCB image.

Notes:

1. VSE/POWER does not support Select Vertical Format (SVF). When list output is to be produced at an SNA work station, the operator at this station has to set up the vertical tab tables. However, the FCB name is sent to the work station if PDIR is supported and if the FCB name has been specified in one of the following:

   - The FCB operand of the JECL * $$ LST statement. This statement may be submitted in the input stream or it may be passed in the JECL operand of a SEGMENT macro.
   - The LFCB macro.

2. The FCB cannot be resident in the SVA. VSE/POWER maintains a table of FCBs and converted LTABs in storage.

3. If an FCB is re-cataloged while VSE/POWER is active, a mismatch of spool controls and output controls may occur. VSE/POWER most likely uses the replaced FCB image during program execution and the new FCB image during printing.

FLASH=([overlayname][,count])
: This operand applies to IBM 3800 only. If you omit the operand and specify DFLT=Y, VSE/POWER uses the default forms-overlay name set for the printer by a SETDF command. If no default forms-overlay name has been set or if DFLT=N was specified, no flashing occurs for your output. Specify for:

overlayname
The one- to four-character name of the forms-overlay frame to be used by the printer.

If the specified name matches the overlay name used for the preceding output, then VSE/POWER processes your output at once. If the two names do not match, VSE/POWER issues a message requesting the operator to insert the named forms-overlay frame into the IBM 3800. There is no verification by VSE/POWER that the correct frame has been inserted.

count
This count, a number from 0 to 255, specifies the number of copies to be flashed with the overlay, beginning with the first copy.

If you specify an overlay name without a count, or if the count value is greater than the number of copies specified (with the COPY or COPYG operand), all copies are flashed. If you code a comma after the name but do not specify a count, then no copies are flashed.

If you specify a count without an overlay name, then the forms-overlay frame loaded at the time of printing is used.

If you specify a count of 0, then the operator is prompted to load the requested forms-overlay frame, but the overlay is not flashed.

FNO=formnumber
The operand specifies the type of forms to be used for the output. If you omit the operand, VSE/POWER assumes that there is no form number, representing your location's standard form.

For formnumber, specify a string of up to four alphameric characters.

If the specified form number matches the form number used for the preceding output, then VSE/POWER processes the output at once. If the two form numbers do not match, VSE/POWER issues a forms-mount message. This applies also to a change from a form-number specification to no such specification and vice versa.

VSE/POWER issues a mount message also if standard print forms are used for the first queue entry processed after a PSTART. Exception: the list-writer task was started for the queue entry with VM specified in the PSTART command.

A forms-mount message is issued for FCB printers every time an FCB is loaded. Forms alignment is then possible.

Do not specify this operand if the output is directed to an IBM 3741.

JSEP={n|(n,option)|(,option)}
The operand specifies the number of desired job separator pages and, optionally, whether separator pages are to be inserted also between each copy of an output file. If you omit this operand, VSE/POWER uses your definition in the JSEP and COPYSEP operands of the POWER generation macro.

Your operator can override this operand by specifying SEP or NOSEP in the PSTART command for the list task.

For a description of the information that is printed on the separator pages, refer to the discussion of the JSEP operand under "POWER Generation Macro" on page 47.

Specify for:

n The number (any from 0 to 9) of job separator pages to be printed.

VSE/POWER prints one more separator page than is specified because printing may have occurred on the paper perforation. The length of a separator page is determined by the FCB image if you specify an FCB name. This length is determined by the LTAB specification if you do not specify an FCB name.

option
Either of the following:

Y To request separator pages to be inserted also between copies of an output file.

N To have no separator pages inserted between copies of an output file.

The specified option takes effect only if a non-zero value is specified for n in JSEP=n either during table generation or in this operand. If you do not specify an option, VSE/POWER uses the COPYSEP value that you supplied for table generation in the POWER generation macro.

LST=listaddr
Use this operand if a program produces two or more different reports which, in addition, may require different forms. You should specify this operand in the * $$ LST statement for each of the reports.

For listaddr, specify the logical or physical address of the printer to be used for printing your output. The printer whose address you specify must be assigned. Specify the address in either of the following forms:

SYSxxx
This form requires a corresponding logical unit assignment when the statement is being processed. For xxx in SYSxxx, you can specify LST or any valid programmer logical unit.

cuu This is the channel and unit number of an actual or a dummy printer.

If you omit this operand, the default is the first printer address specified at partition start-up. If, for example

PRINTERS=00E,01E,02E

was specified during start-up, then printer 00E is used as the default. To have the operands of the * $$ LST statement take effect for printer 01E or 02E, LST=01E or LST=02E must be specified.

LTAB=d0d1d2d3d4d5d6d7d8d9d10d11d12
The operand defines line-skip positions for output that is to be printed on a line printer with a carriage control tape. For output to a printer with an FCB, use the operand FCB=phasename instead. VSE/POWER ignores this operand if you specify also FCB=phasename.

VSE/POWER uses the specified values to control the spooling of your output; it uses your specification in the LTAB=linetab operand of the POWER generation macro if you omit the operand.

In the operand, you define the positions for the first line on the page and for skips to channels 1 through 12. In your specification:

d0 = The number of lines between channel 12 of a page and the first line of the next page.
d1 = The number of lines between the first line on a page and the first skip-stop (channel-1) position.
d2 = The number of lines from channel 1 to channel 2.
d3 = The number of lines from channel 1 to channel 3.
and so on
d12 = The number of lines from channel 1 to channel 12.

If channel 1 is the same as the first line of the printed page, code d1 as 00. Similarly, code as 00 any channel that is not represented. Channel 1, however, must always be present. If you specified D12 as 00, VSE/POWER uses the system-defined default page length.

For VSE/POWER, the sum of d0 and d12 is the maximum number of lines that can fit on a page.

VSE/POWER issues a message if, while spooling the output, a skip to an undefined channel is encountered.

For printing the job's output, you should ensure the following:

- For output to a printer with an FCB

  That the printer's default (VSE system start-up) FCB contents matches your specification in this operand (or the default values used by VSE/POWER for spooling the applicable output.

- For output to a printer with a carriage control tape

  That your operator has mounted a carriage control tape that matches your specification in this operand. Consider using the FNO=formnumber operand together with this operand. It causes the list-writer task to pause and thus gives the operator a chance to mount the correct carriage control tape.

An LFCB or SETPRT macro issued in your program or an FCB operand change the LTAB format for the currently processed VSE/POWER job.

MODIFY=(copymodname[,tablename])
: This operand applies to IBM 3800 only. The operand specifies which predefined data, if any, is to be printed on all pages of your output. Specify for

  copymodname
  : The name of the copy-modification phase. This is the suffix (of up to four characters) that you appended to the system-assigned prefix MOD1 when you cataloged the phase.

  tablename
  : The name (of up to four characters) of the character-arrangement table to be used when printing the copy-modification text. This table need not be one of those you specified with the CHARS operand if your location's IBM 3800 has enough character-generation storage.

    If you do not specify a character-arrangement table, VSE/POWER uses the first table whose name is specified or defaulted with the CHARS operand.

PRI=priority
: The operand specifies the priority of the list output as a number from 0 to 9 (where 9 is the highest priority). If you omit the operand, VSE/POWER assigns the job's priority also to the output.

PWD=password
: Specify this operand if you wish to have your output protected by a password. The password you define must be specified if anyone wants to access this output from another partition.

If you do not define a password, VSE/POWER uses the job-level password protection for protecting the output.

For more information about specifying a password, see the description of the PWD operand under "* $$ JOB: Mark the Start of a VSE/POWER Job" on page 249.

RBC=norbc

The operand specifies the number of pages to be processed before VSE/POWER takes a checkpoint. When VSE/POWER or the system fails, the partial output on the LST queue is not discarded; instead, the output (up to the last recorded checkpoint) is added to the appropriate class chain with disposition 'X'. The operator can query such queue entries by issuing the PDISPLAY LST,CDISP=X command and then take appropriate action.

For norbc, you can specify a number from 1 to 999999. If you specify 0, no checkpoint is taken. The output (incomplete queue entry) will then be discarded when VSE/POWER starts again.

The operand is ignored when DISP=T or N is specified.

RBM=(norbm1,norbm2)

Specify for:

norbm1

The number of lines (list records) to be intercepted before the operator gets a "limit exceeded" message. Specify a number from 1 to 999999.

If you do not specify this number, VSE/POWER uses the value that was defined as the first one in the STDLINE operand of the POWER generation macro.

norbm2

The additional number of list records to be intercepted each time before VSE/POWER reissues the above mentioned message. Again, you can specify a number from 1 to 999999.

If you do not specify this value (in this case you can omit the preceding comma and the parentheses), VSE/POWER uses the value defined as the second one in the STDLINE operand of the POWER generation macro.

RBS=norbs

The operand specifies the number of list pages processed before your output is segmented (this is referred to as count-driven output segmentation). The operand is ignored if your output is to be spooled to tape.

For norbs, specify a number from 1 to 999999. If you specify 0, no segmentation of the output takes place. The value you

specify overrides the value that was specified in the RBS operand of the POWER generation macro.

The operator gets a message whenever a count-driven segmentation occurs.

If you code this operand together with a COPY specification of 2 or more, the pages of the printed output may not be in the sequence as expected. If, four example, you request four copies of a 1000 page output and a segment count of 500, then your output will be arranged as follows:

1. Four copies of pages 1 through 500, followed by
2. Four copies of pages 501 through 1000.

REMOTE=remid
: For remid, specify the remote identifier, a three-digit number, to which the output is to be routed.

  Specify 0 for remid if the associated output is to be routed to the central location.

  If you omit this operand, VSE/POWER routes the output to the central location if the job was submitted locally; it routes the output in accordance with the LSTROUT specification in the PRMT generation macro if the job was submitted remotely.

  In a shared spooling environment, VSE/POWER ignores the SYSID operand if you specify a value other than 0 in this operand.

  If you specify the DEST operand after the REMOTE operand, then the DEST-operand specification overwrites the REMOTE-operand specification.

SYSID={n|N}
: This operand applies to shared spooling. Specify

  SYSID=n
  : If your output is to be available on a certain one of your sharing systems. For n, give the number with which the system's VSE/POWER was initialized (by SYSID=n in the POWER generation macro).

  SYSID=N
  : If the output is to be available on any of the sharing systems.

  If this operand is omitted, VSE/POWER uses your SYSID specification, if any, in the * $$ JOB statement.

TADDR={tapeaddr|(tapeaddr,mode)}
: The operand indicates that the associated output is to be spooled to tape. Specify for:

tapeaddr
The channel and unit address of the tape drive to be used.

mode
The applicable recording density of the tape drive. A list of valid mode specifications is given in the publication VSE/Advanced Functions, System Control Statements.

The streaming-mode specifications for the IBM 8809 (30 and 90) are not supported by VSE/POWER.

The operand is ignored for the IBM 3480.

Specifying this operand forces a disposition of T.

If DISP=T is specified and this operand is omitted, VSE/POWER prompts the operator (by message 1Q55D) for a tape-unit address when your job is being executed.

Two or more output files can be written to a single tape.

Multivolume files are supported. The output to a multivolume file is split at logical boundaries, which is either a skip to channel 1 or a full page.

UCS=(phasename[,option])

The operand requests that a UCB (universal character set buffer) image be loaded into the UCB of the printer that is to be used.

For phasename, specify the name of the UCB-image phase. This phase must be cataloged in one of the sublibraries accessible by VSE/POWER.

For option, specify:

F To indicate that the UCB is to be loaded with the folding operation code; this causes lower case letters to be printed in upper case.

C To prevent data checks from being generated because of print-line mismatches with the UCB.

You can specify both, if this is applicable, by coding FC or CF.

VSE/POWER loads the specified UCB-image phase into the printer's UCB when a list-writer task starts processing the associated output. This UCB image remains in the printer's UCB until VSE/POWER processes another output to this printer and this output was spooled with a different UCB load request.

If you code UCS and omit CHARS in an * $$ LST statement for output to an IBM 3800, VSE/POWER uses the UCS name as the name of the first character-arrangement table.

If you code UCS in an * $$ LST statement for output to an IBM 4248 running in native mode, VSE/POWER uses the UCS name as print band identifier and makes a print band verification by comparing the specified print band identifier with that of the mounted band.

USER=user-information

For user-information, you can specify up to 16 alphameric characters. If this character string includes any blanks, the entire string must be enclosed in a pair of apostrophes.

The specified character string replaces whatever character string was specified in the * $$ JOB statement (if any) for the job. The new character string appears on job separator pages and in the list-account record for the job.

* $$ PUN: Define Punch-Output Attributes

The statement defines the attributes of punch output as spooled by VSE/POWER. These attributes are valid until one of the following occurs:

A subsequent * $$ PUN statement for the same spool device.
A SEGMENT macro for the same punch device
An end-of-job (* $$ EOJ or /&) statement.

Any number of * $$ PUN statements may be included in a VSE/POWER job. Inclusion of * $$ PUN statements does not require * $$ JOB and * $$ EOJ statements to be used.

For each * $$ PUN statement within a VSE/POWER job, VSE/POWER creates a separate punch-queue entry and assigns a job number. This is referred to as data-driven output segmentation. The first punch-queue entry of this job has the same job number as the job's RDR-queue entry; the second and subsequent punch-queue entries have different, unique job numbers.

If an * $$ PUN statement is not included in a VSE/POWER job, the default values defined for VSE/POWER table generation are assumed.

An * $$ PUN statement in error can be corrected when the job is being executed. For more information, refer to "Correction of JECL Errors" on page 239.

Format of the Statement

Operation	Operands
* $$ PUN	[CLASS=class]
	[,COPY={1|number-of-copies}]
	[,DEST={nodeid|(nodeid,userid[,subsystem])}]
	[,DISP={D|disposition}]
	[,FNO=formnumber]
	[,JSEP={n|(n,option)|(,option)}]
	[,PRI=priority]
	[,PUN=punchaddr]
	[,PWD=password]
	[,RBC=norbc]
	[,RBM=(norbm1,norbm2)]
	[,RBS=norbs]
	[,REMOTE=remid]
	[,SYSID={n|N}]
	[,TADDR={tapeaddr|(tapeaddr,mode)}]
	[,USER=user-information]

You can define additional JECL output operands of your own in the autostart DEFINE statement and specify them in the * $$ PUN (or * $$ LST) statement. For more information, see "DEFINE: Specify User-Defined Output Operands" on page 294.

Description of Operands

CLASS=class

The operand defines the class that is to be assigned to the punch output after this output has been spooled by VSE/POWER. It may be specified together with FNO=formnumber to group the same type of punch output. The operand is ignored if your output is to be spooled to tape.

For class, you can specify any letter of the alphabet; you can specify a partition-related number (any from 0 to 9) if your CLASS operand is preceded by a DISP=I specification.

If you do not specify a class, VSE/POWER assigns the class which was specified in the PSTART command for the partition used to execute your program.

COPY={1|number-of-copies}

The operand specifies the number of copies to be punched out. This operand is ignored if the output is to be spooled to tape; to get two or more copies of output that was spooled to tape, have the tape punched out the desired number of times.

For number-of-copies, you can specify any number from 0 to 255. If you specify 0, VSE/POWER produces one copy.

DEST={nodeid|(nodeid,userid[,subsystem])}

Any specification in this operand overrides the specification in the PDEST operand of the * $$ JOB statement.

If you specify REMOTE=remid after this operand, your specification in the DEST operand is overwritten; VSE/POWER then sets the node name to that of your own node and the specified user identifier to the identifier specified in REMOTE=remid.

Specify for:

nodeid

One of the following:

- The name of the node to which the punch output of this VSE/POWER job is to be routed. By specifying only a node name, you indicate that this output is to be processed on a card punch at the central location of that node.

- An asterisk (*) if the destination of the output is also the job execution node.

- A null string if the output is destined for a user at the local node; to do this, simply specify DEST=(,userid).

userid
One of the following:

- The applicable user identifier. For a definition of userid, refer to "Conventions for Showing Command and Statement Formats" on page 15.

  If you specify a user identifier, VSE/POWER cannot punch out the associated output on a local punch device.

- ANY. This indicates that VSE/POWER may make the output available to any user that requests this output.

- PWR$JOB together with the identifier of a remote node using VSE/POWER. This causes VSE/POWER at the destination node to place the output into its reader queue (you can cause the same action by specifying DISP=I together with DEST=nodeid).

subsystem
Applies if, for nodeid, you specified the name of an MVS node.

For subsystem, give the name of the external writer that is to process the spooled output.

DISP={D|disposition}
For disposition, specify how your punch output is to be handled in the punch-queue. Your specification may be one of the following:

D Delete after processing. Your output is written to a card punch according to its class and priority. When the punch-out operation for this output is complete, VSE/POWER deletes the output from the punch queue.

H Hold. Your output remains in the punch queue. The output is not written to a card punch until the operator changes the disposition to D or K by a PALTER command or issues a PRELEASE command for the output.

I Return the output to the input queue. In some cases it may be useful to return the punch output of a VSE/POWER job directly to the input queue. For example, the assembler normally stores its output (including JCL) on tape for subsequent input to a catalog run. Under VSE/POWER, this

output is returned to the input queue if you have a disposition of I assigned to the output.

The newly built input-queue entry has the same job name, remote-identifier, routing information, and priority as the entry that produces the output. The new entry's class is that specified in the * $$ PUN statement; its disposition becomes D.

Use disposition I only for jobs that produce punch output in executable format. The option suppresses count-driven output segmentation and checkpointing; it causes a multiple-copies specification to be ignored.

If your * $$ PUN statement includes also the DEST operand, then VSE/POWER places the output into the XMT queue with a reader queue indication. Your output will therefore be put into the reader queue at the final destination.

K Keep after processing. Your output is automatically written to a card punch according to its class and priority. When the punch-out operation for your output is complete, then VSE/POWER retains this output in the punch queue with a disposition of L.

L Leave in queue. Your output remains in the punch queue; it is not written to a card punch until the operator changes the disposition to D or K by a PALTER command or issues the PRELEASE command for the output.

N Output without spooling. This causes output spooling to be suppressed. The output is written to an output device under control of the program which produces the output. In this case, VSE/POWER ignores all other operands of your * $$ PUN statement, except the PUN operand.

If the spooled device is not available during program execution, then VSE/POWER forces a disposition of D for the output and informs you by a message.

T Spool the output to tape. To start punching out the output, the central operator must issue a PSTART command for this output.

If you specify DISP=T, the following operands have no effect:

CLASS	PRI	RBS
COPY	PWD	REMOTE
DEST	RBC	SYSID

For further information on dispositions, see also Appendix A on page 309.

FNO=formnumber

The operand specifies the type of card forms to be used for the output. If you omit the operand, VSE/POWER assumes that there is no form number, representing your location's standard card form. Do not specify this operand if the output is directed to an IBM 3741.

For formnumber, specify a string of up to four alphameric characters.

If the specified form number matches the form number used for the preceding output, then VSE/POWER processes the output at once. If the two form numbers do not match, VSE/POWER issues a forms-change message. This applies also to a change from a form-number specification to no such specification and vice versa.

VSE/POWER issues a mount message also if standard cards are used for the first queue entry processed after a PSTART. Exception: the punch-writer task was started for the queue entry with VM specified in the PSTART command.

JSEP={n|(n,option)|(,option)}

The operand specifies the number of job desired separator cards and, optionally, whether separator cards are to be inserted between each copy of the output file.

The operand causes stacker selection to be ignored. VSE/POWER uses, instead, the default stacker for the given device.

If you omit this operand, VSE/POWER uses your definition in the JSEP and COPYSEP operands of the POWER generation macro.

Your operator can override this operand by specifying SEP or NOSEP in the PSTART command for the list task.

For a description of the information that is punched into the separator cards, refer to the discussion of the JSEP operand under "POWER Generation Macro" on page 47.

Specify for:

n The number (any from 0 to 9) of job separator cards to be punched.

option
Either of the following:

Y To request separator cards to be inserted also between copies of the output.

N To have no separator cards inserted between copies of an output file.

The specified option takes effect only if a non-zero value was given for n in JSEP=n either during table generation or in this operand. If you do not specify an option, VSE/POWER uses the COPYSEP value that you supplied for table generation in the POWER generation macro.

PRI=priority
The operand specifies the priority of the list output as a number from 0 to 9 (where 9 is the highest priority). If you omit the operand, VSE/POWER assigns the job's priority also to this output.

PUN=punchaddr
For punchaddr, specify the logical or physical address of the card punch to be used for punching out your output. Use this operand if a program produces two or more card files on two or more card-punch devices. Specify this operand in the * $$ PUN statement for each of the files.

Specify the address in either of the following forms:

SYSxxx
This form requires a corresponding logical unit assignment at the time the statement is being processed. For xxx in SYSxxx, you can specify PCH or any valid programmer logical unit.

cuu This is the channel and unit number of an actual or a dummy card punch.

If you omit this operand, the default is the first punch address specified during partition start-up. This is similar to the device selection by VSE/POWER in case of a list output with an LST=listaddr specification in the * $$ LST statement.

PWD=password
Specify this operand if you wish to have your output protected by a password. The password you define must be specified by anyone who wants to access this output from another partition.

If you do not define a password, VSE/POWER uses the job-level password protection for protecting the output.

A password, if used, must be defined as an alphameric string of up to eight characters.

RBC=norbc
The operand specifies the number of cards (punch records) to be processed before VSE/POWER takes a checkpoint. When the system fails, the partial output on the PUN queue will not be discarded; instead, the output (up to the last recorded checkpoint) is added to the appropriate class chain with disposition 'X'.

The operator can query such queue entries by issuing the PDISPLAY PUN,CDISP=X command and then take appropriate action.

For norbc, you can specify a number from 1 to 999999. If you specify 0, no checkpoint is taken. The output (incomplete queue entry) will then be discarded when VSE/POWER starts again.

The operand is ignored when DISP=T, I or N is specified.

RBM=(norbm1,norbm2)
Specify for:

norbm1
The number of cards (punch records) to be intercepted before the operator gets a "limit exceeded" message. Specify a number from 1 to 999999.

If you do not specify this number, VSE/POWER uses the value that was defined as the first one of the STDCARD operand of the POWER generation macro.

norbm2
The additional number of punch records to be intercepted each time before VSE/POWER reissues the above mentioned message. Again, you can specify a number from 1 to 999999.

If you do not specify this value (in this case you can omit the preceding comma and the parentheses), VSE/POWER uses the value defined as the second one of the STDCARD operand of the POWER generation macro.

RBS=norbs
The operand specifies the number of punch records intercepted before the output is segmented (this is referred to as count-driven output segmentation). The operand is ignored if your output is to be spooled to tape.

For norbs, specify a number from 1 to 999999. If you specify 0, no segmentation of the output takes place. The value you specify overrides the value that was specified in the RBS operand of the POWER generation macro.

The operator gets a message whenever a count-driven segmentation occurs.

If output on an IBM 3525 is to be segmented and print commands occur, then either all segments must be punched in a row or separator cards must be specified. This ensures that the last two punch records will not be lost.

REMOTE=remid
For remid, specify the remote identifier, a three-digit number, to which the output is to be routed.

Specify 0 for remid if the associated output is to be routed to the central location.

If you omit this operand, VSE/POWER routes the output to the central location if the job was submitted locally; it routes the output in accordance with the PUNROUT specification in the PRMT generation macro if the job was submitted remotely.

In a shared spooling environment, VSE/POWER ignores the SYSID operand if you specify a value other than 0 in the REMOTE operand.

If you specify the DEST operand after the REMOTE operand, then the DEST operand specification overwrites the REMOTE-operand specification.

SYSID={n|N}
This operand applies to shared spooling. Specify

SYSID=n
If your output is to be available on a certain one of your sharing systems. For n, give the number with which the system's VSE/POWER was initialized (by SYSID=n in the POWER generation macro).

SYSID=N
If the output is to be available on any of the sharing systems.

If this operand is omitted, VSE/POWER uses your SYSID specification, if any, in the * $$ JOB statement.

TADDR={tapeaddr|(tapeaddr,mode)}
The operand indicates that the affected job's output is to be spooled to tape. Specify for:

tapeaddr
The channel and unit number of the tape drive to be used.

mode
The applicable recording density of the tape drive. A list of valid mode specifications is given in the publication VSE/Advanced Functions, System Control Statements.

The streaming-mode specifications for the IBM 8809 (30 and 90) are not supported by VSE/POWER.

The operand is ignored for the IBM 3480.

The specification of tapeaddr forces a disposition of T.

If DISP=T is specified and tapeaddr is omitted, VSE/POWER prompts the operator for a tape-unit address when your job is being executed.

Two or more files can be written to a single tape. Multivolume files are also supported.

USER=user-information
For user-information, you can specify up to 16 alphameric characters. If this character string includes any blanks, the entire string must be enclosed in a pair of apostrophes.

The specified character string replaces any character string specified in the * $$ JOB statement (if there is one) for the job. The new character string is also included in the punch-account record for the job.

* $$ RDR: Adding 3540 Files to Input Jobs

In a writer-only partition, this statement is ignored.

The * $$ RDR statement may be used to insert data from a diskette into an input stream for spooling to the reader queue under control of a local task. The data may be inserted, for example, into an input stream from one of the following:

A card reader or an IBM 3540
A remote work station
Another partition

A diskette file inserted by a * $$ RDR statement may not itself contain another * $$ RDR statement.

Format of the Statement

Operation	Operands
* $$ RDR	[DEV=diskette-cuu] [,FEED={NO|YES}] [,FID='file-id'] [,NOD={1|number-of-diskettes}] [,VER={NO|YES}] [,VSC={NO|YES}]

Description of Operands

DEV=diskette-cuu

The operand does not apply to an * $$ RDR statement for a SYSIN file.

For diskette-cuu, specify the channel and unit address of the diskette unit from which input data is to be read. If VSE/POWER is to insert more than one diskette file, then your job may contain as many * $$ RDR statements as there are insertion files. These RDR statements must specify the same diskette unit address in the DEV operand.

The specified device address may be different from the diskette unit address that is given in the PSTART command. No VSE/POWER error occurs when the input is being read, but the data must be read from the unit whose address is given in the PSTART command. VSE/POWER checks the unit address given in the * $$ RDR statement during program execution, and this address must be known to your system's supervisor.

FEED={NO|YES}
Specify FEED=YES if you want VSE/POWER to eject a diskette on end-of-file and to insert a new diskette.

This operand overrides the default which you may have set up (by the FEED operand of the POWER macro) during VSE/POWER table generation. Your specification by this operand is valid only for the input associated with the * $$ RDR statement.

FID='file-id'
For file-id, specify (within a pair of apostrophes) the name of the file as it is recorded in the HDR1 label on the diskette.

If you omit this operand, VSE/POWER reads the file defined by the first non-secured HDR1 label found on the currently mounted diskette. VSE/POWER gets the length of the file's records from the HDR1 label.

NOD={1|number-of-diskettes}
For number-of-diskettes, specify the maximum number of diskettes to be read. You can specify a value from 1 to 255. Diskette reading stops when either:

- The specified number of diskettes are processed, or
- VSE/POWER has read a diskette whose HDR1 label indicates that it was the last volume of the file.

VER={NO|YES}
Specify VER=YES if you want VSE/POWER to check that file verification was performed on the diskette file (the verify field in the HDR1 label of the file is set to 'V'). VSE/POWER does not read the file if you specify VER=YES and file verification was not performed.

Specify VER=NO or omit the operand if you do not want VSE/POWER to do this checking.

VSC={NO|YES}
Specify VSC=YES if you want VSE/POWER to do volume sequence checking.

Specify VSC=NO or omit the operand if no such sequence checking is desired.

Sequence numbers (on the diskette volumes of a file) must start with 1 and must be incremented by 1 up to a maximum of 99.

Reading Data From the Diskette

To have VSE/POWER spool a job stream completely contained in a diskette file, you can do either of the following:

- Issue a PSTART RDR,duaddr,,'file-id' command.

  This causes VSE/POWER to spool the specified file from the diskette unit at the defined address.

- Insert an * $$ RDR FID='file-id' statement in the input stream.

  This causes VSE/POWER to search for the specified file depending on the PSTART command used for the reader task.

  - The command's format is: PSTART RDR,uraddr,,duaddr

    In other words, the command specifies the address of a diskette unit (duaddr). VSE/POWER searches for the file on the diskette unit specified for duaddr. VSE/POWER spools this file's contents when it has located the file; else it prompts the operator to mount the correct diskette or to cancel the job.

  - The command's format is: PSTART RDR,uraddr

    In other words, the command does not specify the address of a diskette unit. VSE/POWER, in this case:

    1. Assigns (to its partition) the first free diskette unit.

    2. Searches the diskette mounted on this unit for the specified file and, if this file is not there, requests the operator to mount the correct diskette or to cancel the job.

    If VSE/POWER cannot find a free diskette unit, it terminates input spooling for the currently processed job and starts spooling for the next job.

There are two modes of inserting a diskette file into a job stream:

- SYSIN-mode processing

  The diskette data to be inserted consists of records that the system reads from the devices assigned to SYSRDR and SYSIPT.

  The file to be read may contain records of a length of up to 128 bytes. However, any SYSRDR input records (VSE JCL statements, for example) must have the same length as the records of the initial SYSRDR input. In other words, if you insert the * $$ RDR statement into input from an 80-column card reader, then the SYSRDR records in the diskette file must also have a length of 80 bytes.

Note: VSE/POWER passes only the last 80 bytes of an 81-byte record to the reader queue.

Do not specify the diskette unit's device address in the * $$ RDR statement.

- Data-mode processing

  The diskette data to be inserted consists of records that your program reads from a device assigned to a programmer logical unit.

  You must specify the device address of the diskette unit from which the application program requests input data.

  The reader task can read records of a length of up to 128 characters.

  VSE/POWER does not examine the records for control statements; it writes them into the spool file exactly as read. The JECL and VSE JCL statements are read from the spool-input source, which may be any of those listed above.

  If a VSE/POWER job includes two or more * $$ RDR statements for data-mode insertion, these statements must specify the same diskette-unit address.

* $$ SLI: Add a Library Member to a Job Stream

The statement causes VSE/POWER to search for the named library member (subsequently referred to as SLI member) and to include this member into the currently processed job stream. VSE/POWER searches for an SLI member in either:

- The sublibraries defined as a search chain for the VSE/POWER partition, or
- The VSE/ICCF library if one is specified.

VSE/POWER includes the requested SLI member at the point where it encounters the * $$ SLI statement and at the time the spooled job is being executed.

Continuation of an * $$ SLI statement is not allowed. A VSE/POWER job may include more than one * $$ SLI statement.

Use of this statement requires that either of the operands MEMTYPE and SUBLIB was specified in the POWER generation macro. Your job is canceled if this was not done; it is also canceled if VSE/POWER cannot find the requested SLI member.

Format of the Statement

The statement can be used with distinct sets of operands as shown below:

Operation	Operands
* $$ SLI	{MEM=membername[.membertype]|[sublib.]bookname}
	ICCF=(membername[,password]), LIB=(libnumber[,libnumber,libnumber])

Description of Operands

MEM=membername[.membertype]

For membername, specify the name you used to catalog the SLI member. If this name begins with $$, VSE/POWER replaces the second dollar sign by the applicable partition code: $$ becomes $0 for BG, $1 for F1, ... $B for FB (foreground 11).

If your system's access control function has been activated, any member to be retrieved must be defined with universal read access.

For membertype, specify the type you assigned when you cataloged the member. If you omit this specification, VSE/POWER uses the

type you specified or defaulted to by either or both of the POWER (generation) macro operands MEMTYPE and SUBLIB.

Notes:

1. This operand must be coded only once. (Usually, keyword operands can occur more than once, in which case VSE/POWER uses the last specification).

2. You may not use PHASE or DUMP as a member type.

[sublib.]bookname
: This operand is supported for compatibility reasons only. If you use the operand, VSE/POWER uses your specification for bookname as membername specification; it uses your sublib. specification, if present, as .membertype specification.

ICCF=(membername[,password])
: This operand indicates to VSE/POWER that the SLI member to be included resides in a VSE/ICCF controlled library. Specify for:

  membername
  : The name of the member that is to be included into the currently spooled job stream. VSE/POWER does not retrieve the member until the job stream is being executed.

    An * $$ SLI statement with this operand may be part of a job that is being read by a local reader task (from a diskette unit or another partition, for example). In this case, VSE/POWER requires also a valid // ID statement if the ICCF library to be accessed is neither a common nor a public one.

  password
  : The password by which the member is protected, if there is such a password.

LIB=(libnumber[,libnumber][,libnumber])
: The operand names the VSE/ICCF controlled library (or libraries) to be searched for the named SLI member.

  For libnumber, specify the number of each of the libraries to be searched by VSE/POWER. You can specify the numbers of up to three different VSE/ICCF libraries.

Rules for Coding an SLI Member

A library member for inclusion may contain JECL statements, but not an * $$ JOB or * $$ EOJ statement.

An * $$ LST or * $$ PUN statement read from a unit-record input device cannot be continued in an SLI member. An * $$ LST or * $$ PUN statement within an SLI member may not be continued from one line to the next.

If the member is to include * $$ LST, * $$ PUN, or * $$ SLI statements, submit them as * $x LST, * $x PUN, and * $x SLI, respectively. In the statements, x can be any character other than blank or $.

Update statements and also * $$ DATA statements (if used) must follow the
* $$ SLI statement in the same sequence as that of the corresponding statements in the member. An * $$ SLI statement cannot be used as an update statement.

Data that you insert by means of an * $$ DATA statement must be delimited by a /*$SLI statement or by /* or /&.

Nesting of SLI statements is possible; that is, a retrieved SLI member may include SLI statements. While nesting is possible to a virtually unlimited level, be aware that each additional level adds to the complexity of your application. In addition, for each level of nesting, VSE/POWER needs a certain amount of virtual storage.

The statements of a nested SLI member may be updated; however, all of the nested segments are viewed as one statement for the purpose of sequence number checking.

An SLI member may invoke a VSE procedure, but a VSE procedure may not invoke an SLI member.

Changing an SLI Member

Statements may be added to or deleted from an SLI member; individual statements of a member may be replaced if they contain sequence numbers in columns 78 through 80. To do such changes, submit update statements immediately following the * $$ SLI statement. These update statements must have a code in columns 73 through 80 as follows:

```
$ S L I x n n n
```

where x = One of the following characters:

A The statement is to be inserted behind the statement with the same sequence number or before the statement with the next higher sequence number.

B The statement is to be inserted before the statement with the same or higher sequence number.

D The member statement with the same sequence number is to be deleted.

Other than A, B, or D
The member statement with same sequence number is to be replaced, or this statement is to be inserted.

nnn = A statement-sequence number.

Cataloging an SLI Member

How to catalog a member in a sublibrary and how to define this sublibrary in a sublibrary-search chain for a partition is described in the publication VSE/Advanced Functions, System Management Guide.

If you catalog an SLI member under control of VSE/POWER, use VSE/POWER specific job-step and job delimiters within these members. This ensures correct processing of end-of-data, end-of-job-step, and end-of-job conditions. The delimiters are:

```
* $$ /*  for the VSE delimiter /*
* $$ /&  for the VSE delimiter /&
```

Examples for Using * $$ SLI and * $$ DATA Statements

The following examples show the use of the * $$ SLI and * $$ DATA statements; these examples apply accordingly to members that are to be retrieved from a VSE/ICCF controlled library.

Example 1: Job to retrieve a member named ASSM and to insert data into this member:

Job from card reader

```
* $$ JOB    INSERT
* $$ SLI    MEM=ASSM
* $$ DATA   INPUTA
            POWER ACCOUNT=YES
            END
/*
* $$ EOJ
```

Cataloged member ASSM

```
// JOB INSERT
// EXEC ASSEMBLY
* $$ DATA INPUTA
* $$ /&
```

Resulting job stream

```
* $$ JOB INSERT                  (From card reader)
// JOB INSERT                    (From member 'ASSM')
// EXEC ASSEMBLY                 (From member 'ASSM')
   POWER ACCOUNT=YES             (From card reader)
   END                           (From card reader)
/*                               (From card reader)
/&                               (From member 'ASSM')
* $$ EOJ                         (From card reader)
```

Example 2: Job to retrieve a member named XMPL and to update this member by submitting update statements.

Job from card reader — Cols. 73-80

```
* $$ JOB    ASSEMSLI
* $$ SLI    MEM=XMPL.B
// ASSGN    SYSPCH,182            $SLI0020
* $$ DATA   INPUTA
DEFAULT POWER ACCOUNT=YES
        END
/*
// ASSGN    SYSIPT,182            $SLI0100
// ASSGN    SYS010,183            $SLIA100
// MTC      RUN,182               $SLI0140
* $$ EOJ
```

Cataloged member XMPL (of type B) Cols. 73-80

```
// JOB      ASSEMBLE                  XMPL0010
// ASSGN    SYSPCH,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,180                XMPL0100
// EXEC     LIBR                      XMPL0110
   CATALOG DEFAULT.OBJ                XMPL0120
* $$ /*                               XMPL0130
// MTC      RUN,180                   XMPL0140
* $$ /&                               XMPL0150
// JOB      LINK                      XMPL0160
// OPTION   CATAL                     XMPL0170
        INCLUDE DEFAULT               XMPL0180
/*                                    XMPL0200
// EXEC     LNKEDT                    XMPL0210
/&                                    XMPL0220
```

Resulting job stream

```
* $$ JOB    ASSEMSLI
// JOB      ASSEMBLE                  XMPL0010
// ASSGN    SYSPCH,182                $SLI0020   updated (replaced)
// EXEC     ASSEMBLY                  XMPL0030
DEFAULT POWER ACCOUNT=YES                         inserted
        END                                       inserted
/*
// MTC      WTM,SYSPCH                XMPL0050
// MTC      WTM,SYSPCH                XMPL0060
// MTC      REW,SYSPCH                XMPL0070
/&                                    XMPL0080   modified
// JOB      CATALOG                   XMPL0090
// ASSGN    SYSIPT,182                $SLI0100   updated (replaced)
// ASSGN    SYS010,183                $SLIA100   updated (inserted)
// EXEC     LIBR                      XMPL0110
   CATALOG DEFAULT.OBJ                XMPL0120
/*                                    XMPL0130   modified
// MTC      RUN,182                   $SLI0140   updated (replaced)
/&                                    XMPL0150   modified
// JOB      LINK                      XMPL0160
// OPTION   CATAL                     XMPL0170
        INCLUDE DEFAULT               XMPL0180
/*                                    XMPL0200
// EXEC     LNKEDT                    XMPL0210
/&                                    XMPL0220
* $$ EOJ
```

CHAPTER 6. VSE/POWER AUTOSTART

This chapter describes how to define automatic start-up of VSE/POWER by using its autostart function.

Throughout this chapter, the information and specifications given for the IBM 3800 apply also to the IBM 3200 if not specifically stated otherwise.

VSE/POWER AUTOSTART STATEMENTS

VSE/POWER offers an automatic start-up function, called autostart. The function makes VSE/POWER start-up easier, particularly so if VSE/POWER must often be stopped and restarted, frequently with identical spool-device assignments. The following autostart statements are available to start VSE/POWER automatically:

DEFINE The statement defines any additional (user-defined) JECL output operands.

FORMAT The statement indicates whether the VSE/POWER files are to be formatted.

PRINTERS The statement defines list-output spooling devices.

PUNCHES The statement defines punch-output spooling devices.

READER The statement defines an input spooling device.

SET The statement defines various control values.

Autostart statements always start in column 1. They must be submitted via the SYSIPT device assigned to the VSE/POWER partition.

If an error is detected in an autostart statement, VSE/POWER expects the corrected statement to be entered from the console. An error in a SET or DEFINE statement cannot be corrected by the console operator.

Autostart statements are normally submitted as part of a cataloged procedure. Such an autostart procedure normally includes one or more PSTART commands. The procedure may include, in addition, any of the following VSE/POWER commands: PACT, PALTER, PHOLD, and PRELEASE.

In an autostart procedure, the autostart statements have to be arranged in a certain sequence, following the // EXEC POWER statement:

1. SET, if any[7]
2. DEFINE, if any[7]
3. FORMAT
4. PSTART for a partition, followed by these statements, in the sequence as shown:

   READER
   PRINTERS
   PUNCHES

   Other meaningful VSE/POWER commands may be given instead of a PSTART command for a partition.

5. PSTART for an additional partition to be used under control of VSE/POWER, also followed by the statements as shown under 4.

This section describes the autostart statements; their use is shown and explained in Chapter 3.

[7] If any SET or DEFINE statements are placed erroneously behind the FORMAT statement, VSE/POWER ignores them. However, the operator will be warned by message 1Q06I.

DEFINE: Specify User-Defined Output Operands

The statement allows you to define additional JECL keyword operands for future use. VSE/POWER accepts such operands in the * $$ LST or * $$ PUN statement. The value of the keyword operand is syntax-checked during job execution against the specification made in the DEFINE statement.

Newly defined operands might be required, for example, to support certain output devices, such as page mode printers. Currently, IBM-supplied component programs (like PSF) make use of this support. VSE/POWER passes the keyword values in coded form to the component, requesting the queue entry via the spool-access support interface. The component, in turn, interprets the coded values at its print time.

For each output operand to be defined you must specify a separate DEFINE statement. No continuation statement is allowed. One or more values can be associated with a keyword. If more than one value is allowed, the characteristics specified in the DEFINE statement apply to each of them.

If an output operand is valid for both the * $$ LST and the * $$ PUN statements, you must code a separate DEFINE statement for each of them.

Format of the Statement

```
┌────────┬──────────────────────────────────────────┐
│ DEFINE │ {L|P},keyword,id,rep,len                 │
│        │  [,*|A|B|C|H|N[,minval,maxval]]          │
└────────┴──────────────────────────────────────────┘
```

Description of Specifications

{L|P} Specify the type of output: L indicates that the operand is valid for a * $$ LST statement, while P indicates that it is valid for a * $$ PUN statement.

keyword Specify the name of the keyword. It can be up to eight alphameric characters long.

id Specify the registered keyword identifier, as defined in the manual Network Job Entry: Formats and Protocols for System/370 Program Products. You can specify a hexadecimal value in the range of 1 - X'FFFF'. Leading zeros may be omitted.

rep Specify a repetition factor indicating how many values of the defined characteristics can be associated with the specified keyword.

The repetition factor may be any number from 1 to 15, where 1 defines a single keyword value. It must be specified in the * $$ LST or * $$ PUN statement in the form:

keyword=value

Keywords with several values must be specified as:

keyword=(value1,value2,...valuen)

len
Specify the maximum length of the keyword value, which may range from 1 to 255 (a length of 0 is invalid).

If the keyword value is to be converted into a binary field (see type B, below), the length specification refers to the length of this binary field and may range from 1 to 4 bytes only.

If the keyword value is hexadecimal (see type H, below), the length specification describes the required number of bytes in storage and must be a multiple of 2.

*|A|B|C|H|N
Specify the type of the keyword value, where

* Indicates that the keyword value can contain any character, except a blank or a comma. This is the default.

A Indicates that the keyword value must be alphabetic.

B Indicates that the keyword value must be converted into a binary value. The valid range of a binary-type keyword value can be specified by the 'minval' and 'maxval' operands.

C Indicates that the keyword value must be alphameric.

H Indicates that the keyword value must be hexadecimal.

N Indicates that the keyword value must be numeric (character format).

minval
Determines the smallest value that is valid for a binary-type keyword value. The default is zero.

maxval
Determines the largest value that is valid for a binary-type keyword value. The default depends on the length ('len') specification. The following table shows the maximum value, depending on the length of the binary field:

Length	Max. Value
1	255
2	65.535
3	16.777.215
4	99.999.999

Keyword Definition Table

The possible keyword operands that can be specified in the DEFINE statement are described in Network Job Entry: Formats and Protocols for System/370 Program Products. The keyword definition table shown in Figure 17 shows some of the registered keyword ids and other information needed to code a VSE/POWER DEFINE statement.

ID	Max. Repet.	Length (dec.)	Keyword	Meaning
0003	1	2	CKPTLINE	Integer: Range 0 - 32767 Default: 0
0004	1	2	CKPTPAGE	Integer: Range 1 - 32767 Default: Not specified
0005	1	2	CKPTSEC	Integer: Range 1 - 32767 Default: 1
0007	1	8	COMPACT	Alphameric Default: Blanks
0008	1	1	CONTROL	X'80' Force single space X'40' Force double space X'20' Force triple space X'10' Use first character in line as CC Default: X'80'
000B	8	1	DEFAULT	(Not used at output nodes)
0013	1	1	JESDS	(Not used at output nodes)
0014	1	1	LINDEX	Integer: Range 1 - 31
001C	1	8	WRITER	Alphameric or national
001D	1	1-6	FORMDEF	Name: Alphameric or national Default: Not specified
001F	1	1-6	PAGEDEF	Name: Alphameric or national Default: Not specified
0021			PIMSG	Mixed data format Default: Not specified

Figure 17 (Part 1 of 2). Extract from Keyword Definition Table

0022	1	4	THRESHLD	Integer: Range 1 - 99999999 Default: 1
3F00	1	1-128	CICSDATA	Alphameric or national Mapped by CICS
F000 -- FFFF			Reserved for Customers	May be assigned by products for customer use.

where:

alphameric = the characters A through Z, a through z, 0 through 9
national = the characters #, @, and $

Other keyword ids are reserved for private use by JES2/JES3. For further information refer to NJE: Formats and Protocols for System/370 Program Products.

Figure 17 (Part 2 of 2). Extract from Keyword Definition Table

Examples of the DEFINE Statement

If you want to use the FORMDEF operand in your * $$ LST statement, for example, you must code the following DEFINE statement:

```
DEFINE L,FORMDEF,1D,1,6,C
```

Other specifications might be:

```
DEFINE L,PAGEDEF,1F,1,6,C
DEFINE L,CKPTSEC,5,1,2,B,0,32767
DEFINE L,PIMSG,21,1,1,H
```

FORMAT: Specify File-Formatting Options

The statement tells VSE/POWER whether it is to perform a warm start or which of the VSE/POWER files are to be formatted. The statement must follow immediately your SET and/or DEFINE statement(s), if supplied. You use the FORMAT statement in a way similar to a response to message 1Q11D if VSE/POWER autostart is not available.

Only one FORMAT statement may be included in a VSE/POWER autostart procedure.

If you use an autostart procedure and this procedure does not include a FORMAT statement, then VSE/POWER prompts the operator for required formatting instructions. Autostart processing continues after the operator has responded to this prompt. Consider using this approach because a single autostart file procedure can thus provide both warm start and cold start capabilities.

Format of the Statement

```
FORMAT={NO|
       {D|Q}[,A]|
       A}
```

Description of Specifications

NO — Specify NO if the VSE/POWER files are not to be formatted. This is a VSE/POWER warm start.

{D|Q} — Specify D or Q if both the VSE/POWER queue and data files are to be formatted.

A — Specify A if the VSE/POWER account file is to be formatted.

PRINTERS: Define Output-Spooling Devices for List Output

The statement defines the printer(s) for spooling list output from the partition that is being placed under control of VSE/POWER by the preceding PSTART command. To have spooled list output printed, start a list task by another PSTART command. For a description of the PSTART command, refer to "PSTART: Start a Task or a Partition" on page 216.

In your autostart procedure, the PRINTERS statement must follow the READER statement and precede the PUNCHES statement for the same partition.

Your specification in this statement is similar to a response to message 1R50D if VSE/POWER's autostart function is not available.

Format of the Statement

```
PRINTERS={NO|
          cuu[,cuu][,...]}
```

Description of Specifications

NO
: Specify NO if no printer is to be assigned to the partition for spooling list output. If you specify also PUNCHES=NO (see "PUNCHES: Define Output-Spooling Devices for Punch Output" on page 303), you indicate that a reader-only partition is to be started.

cuu[,cuu][,...]
: For cuu, specify the channel and unit numbers of a printer that is to be used. Leading zeros may be omitted.

  You can specify up to 14 printers for spooling list output. If you specify two or more printers, any two cuu specifications must be separated by a comma. The same device(s) may be specified in several partitions.

PUNCHES: Define Output-Spooling Devices for Punch Output

The statement defines the punch device(s) for spooling punch output from the partition that is being placed under control of VSE/POWER by the preceding PSTART command. To have spooled output punched out, start a punch-writer task by another PSTART command. For a description of the PSTART command, refer to "PSTART: Start a Task or a Partition" on page 216.

In your autostart procedure, the PUNCHES statement must follow the READER and PRINTERS statements for the same partition.

Your specification in this statement is similar to a response to message 1R50D if VSE/POWER's autostart function is not available.

Format of the Statement

```
PUNCHES={NO|
         cuu[,cuu][,...]}
```

Description of Specifications

NO
: Specify NO if no card punch is to be assigned to the partition for spooling list output. If you specify also PRINTERS=NO (see "PRINTERS: Define Output-Spooling Devices for List Output" on page 301), you indicate that a reader-only partition is to be started.

cuu[,cuu][,...]
: For cuu, specify the channel and unit numbers of a punch device that is to be used. Leading zeros may be omitted.

  You can specify up to 14 devices for spooling punch output. If you specify two or more punch devices, any two cuu specifications must be separated by a comma. The same device(s) may be specified in several partitions.

READER: Define an Input-Spool Device

Use the statement to define the input-spool device for the partition that is being placed under control of VSE/POWER by the preceding PSTART command. To have spooled input processed, start a reader task by another PSTART command. For a description of the PSTART command, refer to "PSTART: Start a Task or a Partition" on page 216.

In your autostart procedure, the READER statement must precede the PRINTERS and PUNCHES statements for the same partition.

Your specification in this statement is similar to a response to message 1R50D if VSE/POWER's autostart function is not available.

Format of the Statement

```
READER={NO|cuu}
```

Description of Specifications

NO — Specify NO if no input-spool device is to be assigned to the partition that is being started under control of VSE/POWER. This indicates that a writer-only partition is to be started.

cuu — For cuu, specify the channel and unit address of the partition's input-spool device. Leading zeros may be omitted.

Only one input-spool device may be specified per partition. The same device may be specified in several partitions.

SET: Set VSE/POWER Control Values

You can use the statement for the following:

- The definition of a new system identifier for your VSE/POWER.
- Have VSE/POWER use a new network definition table (NDT).

If defined by way of a SET statement, these definitions override the definitions made in the POWER generation macro.

- Have VSE/POWER insert a skip-to-channel 1.
- Have VSE/POWER assign a disposition of X to a reader queue entry that is active when the system fails.

SET statements, if used in your autostart procedure, must follow immediately the // EXEC POWER statement.

Format of the Statement

```
SET {SYSID=sysid|
     PNET=newname|
     SKIP=YES|
     NORUN=YES}
```

SYSID=sysid
: The operand applies to a shared-spooling environment; it defines a new system identifier.

  For sysid, specify a value from 1 to 9.

  If the specification is in error, VSE/POWER issues a message and cancels VSE/POWER start-up.

PNET=newname
: The operand applies to networking.

  For newname, specify the name of the NDT that is to be used instead of the NDT defined in the PNET operand of the POWER generation macro.

  If the specification is in error, VSE/POWER issues a message, logs the statement, and continues start-up processing; VSE/POWER uses, in this case, the NDT defined in the POWER generation macro.

SKIP=YES
: The operand causes VSE/POWER to insert a skip-to-channel 1 if the user's output does not start with one.

NORUN=YES

The operand applies to locally submitted jobs. It causes a disposition of X to be assigned to any reader queue entry that was active when the system failed. You can obtain a list of the affected entries by submitting the command

```
PDISPLAY RDR,CDISP=X
```

Subsequently, you can use the PALTER command to change the affected queue entry's disposition from X to the original one.

Note: All partitions started with autostart are placed in 'pause' mode to avoid that a subsequent job starts to run.

APPENDIX A. VSE/POWER DISPOSITION CODES

The disposition of a queue entry is user defined. VSE/POWER disposes of a queue entry based on the defined disposition. VSE/POWER assigns a disposition different from the specified one if, during the creation or processing of a queue entry, certain conditions arise.

This appendix should help you understand the purpose and reason for the disposition codes that you can use or which VSE/POWER assigns.

Following is a summary of the disposition codes that may be displayed by VSE/POWER in response to a PDISPLAY command:

Code	Meaning/Remarks
A	Appendable. Spool data may be added to the queue entry via the spool-access support.
D	Process the queue entry and delete it after processing. This is the default if you do not specify a disposition.
H	Hold in queue until released.
K	Process the queue entry and keep it in the queue after processing.
L	Leave in queue until released.
X	Hold until the disposition is changed to D or K. A temporary disposition assigned by VSE/POWER to certain queue entries when processing fails.
Y	Hold until the disposition is changed to D or K. Applies only to output being retrieved via the GET service of the spool-access support. Assigned by VSE/POWER either on request by the retrieving program or, to certain queue entries, when processing fails.
*	Indicates that a queue entry is being processed.

The following disposition codes may be specified for an output entry, but they are effective only while the entry is being created.

I	Spool this output to the input (reader) queue with disposition D. Applies to punch output.

N Suppress the spooling of the referenced output when the job entry is being processed.

T Spool the referenced output to tape. Applies to output.

Creation of a Queue Entry:

Specified Disposition	Disposition During Processing	Disposition After Processing — Successful Completion	Disposition After Processing — Unsuccessful Completion
===== DISP=D ==>	n/a	Queued, D	Lost or held with A or X (See Note 2)
===== DISP=H ==>	n/a	Queued, H (See Note 1)	Lost or held with A or X (See Note 2)
===== DISP=K ==>	n/a	Queued, K	Lost or held with A or X (See Note 2)
===== DISP=L ==>	n/a	Queued, L (See Note 1)	Lost or held with A or X (See Note 2)

Processing of a Queue Entry:

Disposition is			
===== DISP=D ==>	*	Deleted	Lost or held with X or Y (See Note 2)
===== DISP=H ==>	---------- n/a (See Note 1) ----------		
===== DISP=K ==>	*	Held, L	Held; may get X or Y (See Note 2)
===== DISP=L ==>	---------- n/a (See Note 1) ----------		

n/a = not applicable.

Note 1: To have the queue entry processed, use the PALTER command to change the entry's disposition to D or K, or issue a PRELEASE command.

Note 2: To process a queue entry that has a disposition of A, X, or Y, use the PALTER command to change the entry's disposition to D or K.

Figure 18. Overview of Disposition Assignments by VSE/POWER

Figure 18 on page 310 shows how VSE/POWER handles a queue entry based on the specified or assigned disposition. The Figure gives you an idea of what may cause a disposition assignment to be changed.

Figure 19 maps the reasons for unsuccessful completion against the disposition of the involved queue entry; it shows the new disposition code assigned by VSE/POWER where applicable. In this Figure:

A dash (-) in a column indicates that the corresponding condition does not apply.

A change of the assigned disposition is indicated by the new disposition code behind the action taken by VSE/POWER.

The word "held" in a column means that the involved queue entry is retained in its queue unchanged.

The words "as is" in a column indicate that a partially created output is placed into the applicable output queue.

Creation of a Queue Entry:

	Reader Queue		Output Queue		
Reason for Failure	Local Read-in	SAS Sub-mission	Output Spooling	SAS Sub-mission	Transmit Queue Receive Function
Abnormal end and warm start recovery of VSE/POWER; PEND with FORCE:					
If RBC=n,...	-	-	As is, X	-	-
If checkpoints	-	-		As is, X	-
Neither	Deleted	Deleted	Deleted	Deleted	Deleted
PDRAIN without EOJ	-	-	-	-	Deleted
PCANCEL jobname:	-	-	As is	-	-
PFLUSH command:					
With BG or Fn			As is	-	-
With PNET	-	-	-	-	Deleted
With uraddr	Deleted	-	-	-	-

SAS = Spool-access-support interface

Figure 19 (Part 1 of 3). Mapping of Conditions to Dispositions for Unsuccessful Completion

Creation of a Queue Entry (Continued):

Reason for Failure	Reader Queue: Local Read-in	Reader Queue: SAS Submission	Output Queue: Output Spooling	Output Queue: SAS Submission	Transmit Queue: Receive Function
PSTOP:					
uraddr without EOJ	Deleted	-	-	-	-
PNET or lineaddr	-	-	-	-	Deleted
SAS if checkpoints	-	-	-	As is, X	-
SAS, no checkpoints	-	Deleted	-	Deleted	-
Abnormal end of SAS connection:					
If checkpoints	-	-	-	As is, X	-
If no checkpoints	-	Deleted	-	Deleted	-
SAS Quit	-	Deleted	-	Deleted	-
SAS Close-append	-	-	-	A	-

Processing of a Queue Entry:

Reason for Failure	Queue Entry Resides in: Reader Queue (Execution)	An Output Queue processed by: List/Punch Writer	An Output Queue processed by: SAS Get Service	Transmit Queue (Send Function)
Abnormal end and warm start recovery of VSE/POWER; PEND with FORCE:				
If SET NORUN=YES	Held, X	-	-	-
If no SET NORUN=YES	Held	Held	Held	Held
Entry protected	-	-	Held, Y	-

SAS = Spool-access-support interface

Figure 19 (Part 2 of 3). Mapping of Conditions to Dispositions for Unsuccessful Completion

Processing of a Queue Entry (Continued):

Reason for Failure	Queue Entry Resides in: Reader Queue (Execution)	An Output Queue processed by: List/Punch Writer	An Output Queue processed by: SAS Get Service	Transmit Queue (Send Function)
PCANCEL jobname:				
If D	Deleted	-	-	-
If K	Held, L	-	-	-
PDRAIN w/o EOJ	-	-	-	Held
PFLUSH with BG or Fn, uraddr, or PNET:				
If D and HOLD	Held, H	Held, H	-	Held, H
If D and not HOLD	Deleted	Deleted	-	Deleted
If K	Held, L	Held, L	-	Held, L
PSTOP:				
uraddr or lineaddr	-	Held	-	-
PNET	-	-	-	Held
SAS	-	-	Held	-
Abnormal end of SAS connection:				
Entry is protected	-	-	Held, Y	-
Entry not protected	-	-	Held	-
SAS Purge request	-	-	Deleted	-
SAS Quit request	-	-	Held	-
SAS Flush-hold request:				-
If D	-	-	Held, H	-
If K	-	-	Held, L	-
SAS Quit-lock request	-	-	Held, Y	-

SAS = Spool-access-support interface

Figure 19 (Part 3 of 3). Mapping of Conditions to Dispositions for Unsuccessful Completion

APPENDIX B. RJE,SNA DATA COMPACTION

Compaction as implemented for the VSE/POWER SNA support provides for compressing non-duplicate characters. It takes advantage of the fact that most data streams use a relatively small subset of the 256 possible EBCDIC characters.

Implementing Data Compaction

For compaction, you must supply a compaction table set. This is quite simple once you know the character set of the data that is to be transmitted. To define the required tables, proceed as follows:

1. Determine the set of all possible characters that may occur in the data stream.

2. Count the number of characters that may occur.

3. Use Figure 20, to match this number with the closest (but not smaller) number in the first column. The second column gives you the number of master characters that are to be selected for the corresponding number of compact-code characters.

Number of Compact-Code Characters	Number of Master Characters
247	3
240	4
231	5
220	6
207	7
192	8
175	9
156	10
135	11
112	12
87	13
60	14
31	15
16	16

Figure 20. Master Character to Compact-Code Character Relationship

Do not specify compaction if the number of compact-code characters is greater than 247.

Assume you have a data set in which 85 different characters occur as shown in Figure 21.

Characters	Name	Hex Code
A - I		C1 - C9
J - R		D1 - D9
S - Z		E2 - E9
a - i		81 - 89
j - r		91 - 99
s - z		A2 - A9
0 - 9		F0 - F9
blank		40
[	Left bracket	AD
.	Period	4B
(	Left parenthesis	4D
+	Plus sign	4E
&	Ampersand	50
]	Right bracket	BD
$	Dollar sign	5B
*	Asterisk	5C
)	Right parenthesis	5D
;	Semicolon	5E
-	Hyphen	60
/	Slash	61
,	Comma	6B
%	Percent	6C
_	Underscore	6D
?	Question mark	6F
:	Colon	7A
#	Pound sign	7B
@	At sign	7C
'	Apostrophe	7D
=	Equal sign	7E
"	Quotation mark	7F

Figure 21. Example of a Compact-Code Character Set

In the first column of Figure 20, the number closest to 85 (but not smaller) is 87. Increase the number of characters in the character

set to equal the number given in Figure 20 on page 315. You may choose, for example:

```
<  Less than       (X'4C')
>  Greater than    (X'6E')
```

From this character subset, select the m most frequently used characters (where m is the number of master characters given in the second column of Figure 20). For a subset of 87 characters, m equals 13. For the purpose of this example, the 13 most frequently used character might be:

```
a, d, e, g, i, l, n, o, r, s, t, u, blank
```

4. Define the compaction table for VSE/POWER by coding a PCPTAB macro. For this example, you would code the PCPTAB macro with operands as follows:

```
PCPTAB MASTER=(81,84,85,87,89,93,95,96,99,A2,A3,A4,40),
       NOMAST1=(A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,
       T,U,V,W,X,Y,Z,82,83,86,88,91,92,94,97,98,A5,A6,
       A7,A8,A9,0,1,2,3,4,5,6,7,8,9,AD,
       4B,4D,4E,50,BD,5B,5C,5D,5E,60,61,6B,
       6C,6D,6F,7A,7B,7C,7D,7E,7F,4C,6E)
```

The macro builds the required tables using these specifications. For a description of the PCPTAB macro, refer to "PCPTAB Macro" on page 76.

This is all you have to know in order to use VSE/POWER's compaction support. The section "Method of Compaction" below is provided to give you an understanding of how compaction works. It may be helpful for reading storage dumps that include compacted data.

Method of Compaction

To compress and compact the data stream, VSE/POWER performs the following:

1. Scans the data, looking for consecutive blank character strings of length 2 or greater. It compresses the data of such a string into one string control byte (SCB) for up to 63 blank characters.

2. Scans the data, looking for consecutive duplicate character strings of length 3 or greater. It compresses the data of such a string into two bytes: one SCB and one duplicate character for up to 63 duplicate characters.

3. For each string of non-blank and non-duplicate characters, it checks if any character is not a member of the set of compact-code characters (master or non-master characters). If so, VSE/POWER creates an

SCB for non-compressed and non-compacted characters and continues with the next string.

4. If all characters are in the compact-code subset, VSE/POWER translates the whole string to compact code. Then VSE/POWER scans the string, looking for two consecutive master characters. Each time this occurs, VSE/POWER compacts the two bytes into a single byte and continues. A fourth kind of SCB describes the compacted string.

The String Control Byte (SCB)

The SCB as defined for SNA consists of a two-bit string identifier and a six-bit count field as follows:

SCB Bits	Explanation
xx000000	Reserved.
00xxxxxx	Non-compressed characters. The SCB is followed by up to 63 data bytes. The number of data bytes is given by xxxxxx. The next SCB follows the last data byte.
01xxxxxx	Compact-code characters. The SCB is followed by up to 63 bytes. Each byte may represent two consecutive master characters, or a single character in the compact-code subset; and both kinds of bytes may exist in the same string. The number of bytes is given by xxxxxx; and it represents the number of bytes following the SCB, not the number of data bytes represented by the string before compaction. The next SCB follows the last byte.
10xxxxxx	Duplicate blanks. The SCB represents up to 63 consecutive blanks (X'40'). The number of blanks represented is given by xxxxxx. The next SCB follows this SCB immediately.
11xxxxxx	Duplicate next character. The SCB and the character immediately following the SCB represent 1-63 consecutive duplicate characters. The duplicated character is the character following the SCB. The number of duplicate characters is given by xxxxxx. The next SCB follows the duplicated character.

An SCB and its string may not span request unit (RU) boundaries.

There need not be any relationship between the boundaries of an SCB string and a logical record. A logical record may begin or end in the middle of an SCB string. In the same manner, single-character string control characters may be compressed as well as data.

The Compaction Tables

The PCPTAB macro builds two 256-byte compaction tables, one for testing (table TRTC) and one for translation (table XLTC). In addition, the macro builds an FMH3 (function management header) ready for transmission to the work station.

The table TRTC contains zeros at the displacements corresponding to each compact-code character of the subset and nonzero values at all other displacements. The table XLTC contains meaningful entries only at displacements corresponding to the compact-code subset members. The values at those displacements are such that no compact-code character translates into a byte whose high-order digit and low-order digit are less than m (where m is the number of master characters).

The value of the first master character is always X'F0'. The value of the second master character is X'F1', and so on, until the master-character subset is exhausted:

- If m is 16, then there are no non-master characters, and the XLTC is then complete. If m is
- If m is less than 16, the value of the first non-master character is X'Fm+1', and so forth, up to X'FF'.
- If m is less than 15, the next 16 non-master characters are assigned X'E0' - X'EF'. If m is less than 14, the next 16 non-masters are assigned X'D0' - X'DF'.

And so on, until the point is reached that, if X'x0' were assigned to the next non-master, both digits would be less than m. When this point is reached, the next non-master is assigned X'yz', where y=m-1 and z=m. The following non-master character is assigned X'yz+1', and so on, up to X'yF'. The next non-master characters are assigned X'y-1z', X'y-1z+1', up to X'y-1F'.

Each time X'yF' is reached, the high-order digit is reduced by 1, until the last non-master character is assigned X'0F'.

Using the example given in this appendix, the tables generated with the PCPTAB macro are as shown in Figure 22. For the TRTC table, a dash represents any nonzero value. For the XLTC table, a dash represents any value, because such a character will not occur in the string. VSE/POWER uses the TRTC and XLTC tables in the following way:

1. It scans the data and separates strings of duplicate characters from strings of non-duplicate characters. Next, it scans the non-duplicate strings for characters outside the compact-code subset using the TRTC table.

   If it finds a character outside the compact-code subset, VSE/POWER does not attempt to perform compaction. The string goes into the request unit unchanged.

If all characters are members of the compact-code subset, VSE/POWER translates them to compact code, using the XLTC table.

2. It scans the data, looking for two adjacent master characters and combines these characters into one byte.

TRTC Table:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4x	00	-	-	-	-	-	-	-	-	-	00	00	00	00	00	-
5x	00	-	-	-	-	-	-	-	-	-	00	00	00	00	00	-
6x	00	00	-	-	-	-	-	-	-	-	-	00	00	00	00	00
7x	-	-	-	-	-	-	-	-	-	-	00	00	00	00	00	00
8x	-	00	00	00	00	00	00	00	00	00	-	-	-	-	-	-
9x	-	00	00	00	00	00	00	00	00	00	-	-	-	-	-	-
Ax	-	-	00	00	00	00	00	00	00	00	-	-	-	-	-	-
Bx	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Cx	-	00	00	00	00	00	00	00	00	00	-	-	-	-	-	-
Dx	-	00	00	00	00	00	00	00	00	00	-	-	-	-	-	-
Ex	-	-	00	00	00	00	00	00	00	00	-	-	-	-	-	-
Fx	00	00	00	00	00	00	00	00	00	00	-	-	-	-	-	-

XLTC Table:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3x	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4x	FC	-	-	-	-	-	-	-	-	-	6D	6E	FD	6F	5D	-
5x	5E	-	-	-	-	-	-	-	-	-	5F	4D	4E	4F	3D	-
6x	3E	3F	-	-	-	-	-	-	-	-	-	2D	2E	2F	FE	1D
7x	-	-	-	-	-	-	-	-	-	-	1E	1F	FF	0D	0E	0F
8x	-	F0	E0	E1	F1	F2	E2	F3	E3	F4	-	-	-	-	-	-
9x	-	E4	E5	F5	E6	F6	F7	E7	E8	F8	-	-	-	-	-	-
Ax	-	-	F9	FA	FB	E9	EA	EB	EC	ED	-	-	-	-	-	-
Bx	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Cx	-	EE	EF	D0	D1	D2	D3	D4	D5	D6	-	-	-	-	-	-
Dx	-	D7	D8	D9	DA	DB	DC	DD	DE	DF	-	-	-	-	-	-
Ex	-	-	CD	CE	CF	BD	BE	BF	AD	AE	-	-	-	-	-	-
Fx	AF	9D	9E	9F	8D	8E	8F	7D	7E	7F	-	-	-	-	-	-

Figure 22. VSE/POWER Compaction Tables

The example below shows how this algorithm works for the following sentence:

'2. Scans the data, looking for consecutive duplicate character strings of length 3 or greater.'

This would appear in storage (in hexadecimal) as:

F24B40E2 838195A2 40A38885 408481A3 816B4093 96969289
95874086 96994083 9695A285 83A4A389 A5854084 A4979389
8381A385 40838881 998183A3 859940A2 A3998995 87A24096
86409385 9587A388 40F34096 99408799 8581A385 994B

VSE/POWER scans the string, looking for consecutive duplicate character strings of length 3 or greater. No such string exists in the first 63 characters, and VSE/POWER establishes the 2 (X'F2') as the beginning of a non-duplicate string, and the blank (X'40') between the words "character" and "strings" as the end of the non-duplicate string. VSE/POWER tests this string, using the (example) TRTC table, and finds that all characters are members of the compact-code subset. VSE/POWER now translates the string to compact code, using the XLTC table. The resulting string (first 63 bytes only, because the maximum length indication in the SCB is 63) becomes:

9E6EFCCD E1F0F6F9 FCFAE3F2 FCF1F0FA F02DFCF5 F7F7E5F4
F6F3FCE2 F7F8FCE1 F7F6F9F2 E1FBFAF4 E9F2FCF1 FBE7F5F4
E1F0FAF2 FCE1E3F0 F8F0E1FA F2F8FC

Now VSE/POWER scans the string, looking for and translating pairs of adjacent master characters. The target buffer for this example then becomes:

6A9E6EFC CDE1069C FAE32C10 A02DC577 E5463CE2 78FCE176
92E1BAF4 E92C1BE7 54E10A2C E1E308F0 E1A28C

The first byte contains the SCB; the underlined bytes represent two master characters. Thus 63 characters have been reduced to 43, including the SCB.

The reduced character string is sent to the work station; there the string is re-translated into the original hexadecimal code, and then passed to the printer.

APPENDIX C. DISK DUMP PROGRAM OF VSE/POWER

When reporting a problem to IBM, you may be asked to supply a dump of one or more or all of the VSE/POWER files. VSE/POWER includes the program IPW$$DD for this purpose. The program writes a dump of the specified VSE/POWER file(s) to a line printer or tape assigned to SYSLST; it provides, on request, a dump of the queue record and DBLK groups belonging to a certain job.

How to Proceed

1. Make assignments (the same as for VSE/POWER execution) for the files to be dumped.

2. Use DLBL/EXTENT statements (the same as for VSE/POWER execution) for the files to be dumped. These statements need not be submitted if they are stored permanently in your system's label-information area.

3. Invoke the program. Figure 23 on page 324 shows an example of the control statements needed to invoke the program.

```
     // JOB name
(1)  // ASSGN SYS000,...
(1)  // ASSGN SYS001,...
(1)  // ASSGN SYS002,...
          ... ... ...
(1)  // ASSGN SYSLST,...
(2)  // DLBL IJAFILE,'VSE.POWER.ACCOUNT.FILE'
(2)  // EXTENT SYS000,SYSWK1
(2)  // DLBL IJQFILE,'VSE.POWER.QUEUE.FILE'
(2)  // EXTENT SYS001,SYSWK1
(2)  // DLBL IJDFILE,'VSE.POWER.DATA.FILE'
(2)  // EXTENT SYS002,SYSWK1
     // EXEC IPW$$DD
     /*
(3)  // CLOSE SYSLST
```

(1) Assignments as follows: SYS000 = for the account file; SYS001 = for the queue file; SYS002 = for the first or only data-file extent; SYS003 = for the file's second extent, up to SYS016; SYSLST = for a printer or tape.

(2) Supply a copy of your location's DLBL and EXTENT statements for the VSE/POWER files, if these statements are not stored permanently.

(3) Needed only if SYSLST is assigned to a tape device.

Figure 23. Control Statement Set for Disk Dump Program Invocation

Specifying Options

When the program receives control, it prompts you at the console as follows:

```
DUMP FUNCTION=
```

Respond by entering one of the following (behind the equal sign):

A To get a dump of the account file.

D To get a dump of the data file.

Q To get a dump of the queue file.

S To get a dump of the queue-control area (applies only to a shared-spooling environment).

jobname[,jobnumber[,queue[,sub-queue]]]
To get a dump of the queue record and associated DBLK groups of a certain job, specify for:

jobname
The name of the VSE/POWER job whose queue entry is to be dumped.

jobnumber
The VSE/POWER assigned job number, if this is of significance.

queue
One of the following, whichever applies:

L For LST queue (default).
P For PUN queue.
R For RDR queue.
X For XMT queue.

sub-queue
Applies only if your response includes X (for XMT queue). In that case, you can specify:

JOB For the JOB sub-queue.
OUT For the OUTPUT sub-queue.

If you omit a subqueue specification, VSE/POWER searches the entire XMT queue for the named VSE/POWER job.

EOJ To end program execution.

When the dump is completed, the program again prompts you with the message

```
DUMP FUNCTION=
```

You can now enter a new set of specifications (see above), or end execution of the dump program by entering EOJ.

Format of the Dump

VSE/POWER prints a block of four lines for every 100 bytes. The contents of the lines are as shown by the example in Figure 24 on page 326.

```
(1) C  1 BL 2016 CH // JOB POWJOB01           ...    DATE 06/30/87,
(2) HD 3         ZN 664DDC4DDEDDCFF44444444444...4444CCEC4FF6FF6FF6
(3) RC 7         NM 11016207661620100000000000...00004135006130187B
(4)                 01...5...10...15...20...25.....85...90...95.....
```

(1) The first of four lines. It displays the printable characters (CH) of the 100-byte string.

(2) The zone-part (ZN) of each byte.

(3) The numeric part (NM) of each byte.

(4) A scale indicating the position of the bytes in the string.

Figure 24. The Format of a CKD Disk Dump

GLOSSARY

The following terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the IBM Vocabulary for Data Processing, Telecommunications, and Office Systems, GC20-1699.

This glossary includes definitions developed by the American National Standards Institute (ANSI) and the International Organization for Standardization (ISO). This material is reproduced from the American National Dictionary for Information Processing, (c) 1977 by the Computer and Business Equipment Manufacturers Association, copies of which may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions of ANSI are marked with an asterisk (*); definitions of ISO include (ISO) before the definition.

ACB. Access method control block.

account file. a file on disk maintained by VSE/POWER to hold the accounting information collected by VSE/POWER for the programs which it controls.

ACF/VTAM. Advanced Communication Function for the Virtual Telecommunication Access Method. An IBM program product that provides single-domain network capability, and optionally, multiple-domain capability.

adjacent nodes. Two nodes that are connected by a data link with no intervening nodes.

alternate route. The routing path that is used for the transmission of jobs or output from the local node to the destination node if the primary route is not active and signed-on.

*** ASCII.** American National Standard Code for Information Interchange. The standard code, using a coded character set consisting of 7-bit coded characters (8 bits including parity check), used for information interchange among data processing systems, data communication systems, and associated equipment. The ASCII set consists of control characters and graphic characters.

API. Application program identifier.

ASI. (automated system initialization) A method of starting a VSE system with input from cataloged procedures.

autostart. A VSE/POWER function that starts VSE/POWER automatically based on input from SYSIPT or an ASI JCL procedure.

Binary synchronous communication (BSC). Communication using binary synchronous transmission.

BSC. See binary synchronous communication

BSC line. A data transmission line that allows binary synchronous communication (BSC).

| **channel-to-channel attachment (CTCA).** An extension of VSE/POWER's networking function that allows data to be exchanged between two virtual machines under control of VM/System Product.

class. A means of grouping output or jobs that require the same set of resources for their execution.

CMS. Conversational monitor system, a component of VM/SP

cold start. The initialization of input and output work queues. All information present in the queues before the cold start is lost.

CTCA. See channel-to-channel attachment.

data block. The unit of transfer (buffer) that VSE/POWER uses for both disk and tape.

data file. A file on disk, maintained by VSE/POWER to hold the input and output program data records required and generated by VSE programs under control of VSE/POWER.

DBLK (data block) group. The smallest unit of space that can be allocated to a VSE/POWER job on the data file. This allocation is independent of any device characteristics.

device service task. A task of VSE/POWER. The task controls the transfer of output for being printed or punched out by a device owned by a subsystem or program in another partition.

direct link. A direct link is a connection between two adjacent nodes in a network.

disposition. A means of indicating to VSE/POWER how job input and output is to be handled. A job may, for example, be deleted or kept after processing.

drain. To end the transmission of data across a network such that the transmitter (receiver) task being used is no longer available for other work.

Compare also with "flushing".

end node. A node designated as the final destination for a job or output. An end node can be the local or any other node within a network.

execution node. A node at which a job is executed; it is the end node for jobs. An execution node may be another VSE/POWER node or any other node that supports the networking protocol used by VSE/POWER.

flushing. The discontinuation of a job that is being processed by a VSE/POWER task.

intermediate storage. A storage device used by VSE/POWER for spooling job input before execution and job output during execution. In case of spooled output, VSE/POWER uses the device as input the VSE/POWER list and punch tasks. or both.

job entry control language (JECL). A control language that allows the programmer to specify how VSE/POWER is to handle a certain job.

list task. A VSE/POWER task. It controls the writing of spooled job output to a printer. See also "task."

local device. A channel-attached input/output device.

local node. The node at which a user is located and operating. Normally the own VSE system.

multitasking. Concurrent execution of one or more sub-tasks attached to a main task within one partition.

MVS. Multiple virtual storage. A short name for the OS/MVS, an IBM operating system.

name of node. See nodeid.

name of user. See user identifier.

*** network.** An interconnected group of nodes.

networking function. A VSE/POWER function that allows communication between nodes within a network. Transmission can be via BSC or SDLC lines or CTCA.

networking protocol. A set of records which must be transmitted in a set sequence in order to:

- start a network session
- control the flow of data across through the network
- terminate a network session

The protocol is an agreed standards to which all member nodes in the network must adhere.

node. In a network, an end point of a link or a junction common to two or more links in a network. Nodes can vary in routing and other functional capabilities.

nodeid. Node identifier; the name by which a node is known within a network.

OPTB. See Output Parameter Text Block.

Output Parameter Text Block (OPTB). In VSE/POWER's spool-access support, information that is contained in an output queue record if a * $$ LST or * $$ PUN statement includes any user-defined keywords that have been defined for autostart.

page fault. A program interruption that occurs when a page marked "not in real storage" is referred to by an active page. Synonymous with page translation exception.

PNET. See networking function.

primary route. The preferred routing path out of two such paths that are provided by a direct link for the transmission jobs or job output from the local node to a node with which no direct link exists.

priority. A rank assigned to each job within its class that determines its precedence in receiving system resources.

punch task. A VSE/POWER task. It controls the writing of spooled job output to a punch device. See also "task."

purge. To purge a job or output from the queues means to delete all references to this job or output from VSE/POWER spool files.

queue. A waiting line or list formed by items in a system waiting for service.

queue entry. The queue record of a certain job or output and the related data records, both stored on disk.

queue file. A file on disk, maintained by VSE/POWER to control the processing of user jobs in the system.

queue record. A record in the queue file containing descriptive information about a job or job output.

receiver task. Receives jobs, list and punch output, messages, and commands; writes the received jobs or output to the spool files; queues commands and messages for further transmission or for processing and display, respectively.

remote job entry (RJE) function. Allows jobs to be submitted at RJE work stations for execution by the system and for output to be obtained at a work station.

remote work station. An input/output control unit and one or more input/output devices attached to a system through a transmission control unit.

resource. Any code, control block, table, record, or file used by a task. More properly called "serial resource".

resource management. The mechanisms that protect serial resources from concurrent access by competing tasks.

routing. The process of transmitting jobs and output from one node to another. Routing paths and destinations can be specified in the PNODE generation macro and in JECL statements. See also "alternate route" and "primary route".

RSCS. Remote spooling communications subsystem, a component of VM/SP.

SDLC. See synchronous data link control.

segmentation. A VSE/POWER facility to break bulky list or punch output into segments so that printing or punching can be started before execution of the generating user program has been completed.

session. The period of time during which a user of a remote user can communicate with an interactive system. A session can exist between any two nodes in a network and is established by the use of ACF/VTAM or ACF/VTAME.

shared spooling. A VSE/POWER function that allows two or more VSE systems running under VSE/POWER to share a single set of VSE/POWER spool files and one account file.

spool-access support task. A VSE/POWER task that controls the access of VSE/POWER services for spooling job or output data and for queue manipulation.

| **spool files.** The queue file and data file on disk, used by VSE/POWER for spooling job input and job output.

spooling. The storing of job input and of job output on an auxiliary storage device, concurrently with job execution and in a format convenient for later processing and actual output, again concurrently with job execution.

storage page. A unit of program address space of 2K or 4K bytes and aligned on a 2K or 4K boundary.

synchronous data link control (SDLC). A discipline for managing synchronous, transparent, serial-by-bit information transfer over a communication channel.

task. The basic unit of synchronous program execution. It consists of instructions operating upon program data. Although a task is executed synchronously with respect to its own instructions, these are executed asynchronously with respect to all other tasks existing in the system.

*** transmission.** The sending of data from one place for reception elsewhere.

transmitter task. Controls the transmission, from the local node, of jobs, list and punch output, messages, and commands.

TSO. Time sharing option, a component of MVS

unit of transfer. The amount of virtual storage needed to hold a block of data that is transferred to or from an I/O device by one I/O request.

user identifier. A character string that identifies the user who submitted or is to receive a job or output.

USS. Unformatted system services.

| **VIO area.** See virtual I/O area.

| **virtual I/O (VIO) area.** An area on the page-data-set volume used by system programs.

VM/SP. Virtual Machine/System Product, an IBM program product.

VSE/ICCF. Interactive Computing and Control Facility, an IBM program product.

warm start. A restart that allows reuse of previously initialized input and output work queues. Contrast with cold start.

BIBLIOGRAPHY

Following is a list of IBM publications which you may need to consult. For additional related publications, refer to the latest IBM System/370, 30XX and 4300 Processors Bibliography, GC20-0001.

VSE/POWER:

Application Programming, SC33-6276

Remote Job Entry User's Guide, SH12-5328

Networking User's Guide, SC33-6140.

Reference Summary, SH12-5435.

VSE/System Package, Messages and Codes,[8]

- SC33-6181 plus SC33-6184 if your VSE/POWER runs on Version 2 of VSE/SP
- SC33-6310 if your VSE/POWER runs on Version 3 of VSE/SP

VSE/Advanced Functions:

System Management Guide, SC33-6191.

Planning and Installation, SC33-6193.

System Control Statements, SC33-66198

Operation, SC33-6194.

Network Program Products, Planning, SC27-0658

Network Job Entry: Formats and Protocols for System/370 Program Products, GG22-9373.

DOS/VS IBM 3800 Printing Subsystem Programmer's Guide, GC26-3900.

IBM 3200 Printing Subsystem Support Program, Programmer's Guide, SC18-0100.

[8] All VSE/POWER messages are now contained in Volume 1 of this VSE/SP manual; the former VSE/POWER Messages manual, SH12-5520 has been discontinued.

INDEX

Special Characters

* $$ CTL statement 244
* $$ DATA statement 245
* $$ EOJ statement 248
* $$ JOB (JECL) error correction 239
* $$ JOB statement 249
* $$ LST (JECL) error correction 239
* $$ LST statement 256
* $$ PUN (JECL) error correction 239
* $$ PUN statement 271
* $$ RDR statement 280
* $$ SLI statement 284
// OPTION NOLOG consideration 237

A

ABE (operand of PRMT) 65
abnormal end
  by PDRAIN request 176
  by PEND request 178
  by PFLUSH request 180
  by PGO request 185
  by PSTOP request 227
  of a sharing system (shared spooling) 86
  of VSE/POWER 122
  reset status (shared spooling) 209
  result in other sharing systems 123
  sharing systems 123
ACCOUNT (operand of POWER) 48
account file
  delete records of 129
  display of 158
  full condition 103
  organization of 25
  retrieve from punch queue 129
  save to disk 129
account file (continued)
  save to tape 130
  sharing the 9
  size of 29
  spool to punch queue 129
account-file-full condition 103
account records
  deletion of 129
  punch-out format 129
ACF/VTAM
  application name 73
  generation requirements 39
  interface, set-up 106
active time (shared spooling) 58
add statements to SLI member 286
additional buffer expansion 65
additional resources 15
ADDR (operand of PLINE) 60
align forms (see forms alignment)
ALLUSER messages (see messages)
alphameric character, definition of 17
alter job attributes
  by class 135
  by creation date 135
  by disposition 135
  by form number 135
  by from node 136
  by from user 136
  by priority 135
  by system identifier 136
  by to node 136
  by to user 136
  command for 132
  form number 138
  job class 137
  job/output destination 138
  number of copies 137
  partition input class 139
  partition priority 98
  priority 138
  remote identifier 138
  search operands for 134
  user identifier 139

amendments, summary of xiii
application identifier
  in the APPLID operand of PNODE 73
  in the SNA operand of POWER 56
application programming
  defining VSE/POWER jobs 237
  loading print buffers (FCB/UCB) 99
  logging VSE/POWER jobs 237
  output segmentation 103
  VSE/POWER job versus VSE job 237
APPLID (operand of PNODE) 73
assignments (of devices) 84
AUTH (operand of PNODE) 73
authorization (see command authorization)
automatic restart, networking 224
automatic VSE/POWER start-up
  file formatting 299
  format queues 90
  statements for 291
  using operator commands for 291
autostart (see automatic VSE/POWER start-up)
autostart statements
  DEFINE 294
  error correction 291
  FORMAT 299
  overview 291
  PRINTERS 301
  PUNCHES 303
  READER 305
  sequence of 291
  SET 307



backup for queue entries
  command for 199
  example of 117
balancing of job load (shared spooling) 7
BE (operand of PRMT) 65
benefits of shared spooling
  centralized unit-record I/O 7
  job-load balancing 7
  shared user files 7
blanks for X'01' through X'3F' 70
block size (unit of transfer) 49
break down of system (see warm start)
broadcast messages
  command for 141
  deletion of 149
BSC line
  address of control unit 60
  characteristics, definition of 60
  I/O trace for 223
  leased 61
  password for 61
  start for networking 224
  starting a 106
  starting for network use 224
  starting for RJE 223, 224
  stopping a 107, 230
  switched 61
  time out for 61
BSC work station (see remote job entry BSC)
BSC, minimum configuration for use 22
buffers (see also print buffer)
  additional expansion of. 2770 and 3741 65
  data block (unit of transfer) 49
  expansion of for 2770 65
  forms control, name of image 261
  number of for networking 74
  release of 218, 220
  size of for networking 74
  universal character set (UCB) 269
BUFSIZE (operand of PNODE) 74
building number
  specification of in * $$ JOB statement 250
BURST (operand of * $$ LST) 257
burster-trimmer control 257



cancel (see stopping)
carriage control tape 51
central operator commands
  abbreviation of 126
  account-file punch/save 129

central operator commands (continued)
activate transmitter/receiver 131
alter job attributes 132
backup queue 199
cancel (flush) a queue entry 180
cancel job/status display 144
delete job/output 146
display queue entries on tape 199
display status 151
drain (deactivate) receiver/transmitter 176
end operation 178
entering of 96
examples (see examples)
format of 128
go (reactivate) 184
hold a queue entry 187
how to enter 125
inquire line/node status 192
leave a queue entry 187
load network definition table 197
messages, routing of 141
offload queue entries 199
overview 127
PACCOUNT 129
PACT 131
PALTER 132
PBRDCST 141
PCANCEL 144
PDELETE 146
PDISPLAY 151
PDRAIN 176
PEND 178
PGO 184
PHOLD 187
PINQUIRE 192
PLOAD 197
POFFLOAD 199
PRELEASE 204
PRESET 209
PRESTART 211
PSETUP 214
PSTART 216
PSTART, use of for autostart 90
PSTART, use of for starting tasks 100
PSTOP 227
PXMIT 234
reactivate a task 184
central operator commands (continued)
release (from hold/leave) 204
remote job entry control 125
reset job status 209
restart output 211
save queue entries 199
setup of print page 214
short form of 126
start a partition/task 216
stop a partition/task 227
summary about 125
task reactivation 184
centralizing unit-record I/O equipment (shared spooling) 7
change (see alter)
channel-to-channel attachment (CTCA)
activate PNET using CTCA 224
PNODE BUFSIZE specification 74
character arrangement table (3800) 258
character specification rules (data compaction) 77
CHARS (operand of * $$ LST) 258
class assignment
by * $$ CTL 244
by * $$ JOB 250
by * $$ LST 258
by * $$ PUN 272
by PSTART 216
change job attributes by 135
concept 42
default for partition output 223
display by 155
in status display 164
job deletion by 148
job hold by 189
partition-related (example of) 226
release job by 206
CLRPRT (operand of POWER) 48
CMPACT (operand of * $$ LST) 258
CMPACT (operand of PRMT) 65
CODE (operand of PLINE) 60
cold start
automatic 90
interactive 94
with shared spooling 85
command authorization
for networking 73
commands (see central operator commands)

compaction table (see data compaction)
compatibility
  with VSE/POWER Version 1 19
component selection
  activation 65
  for message-output device 66
  for print output device 66
  for punch output device 67
compression of space 69
concepts
  define, delimit, log a job (JECL) 237
  output segmentation 103
  VSE/POWER job versus VSE job 237
configuration
  for using BSC 22
  for using SNA 21
  generation examples 79
CONSOLE (operand of PRMT) 65
continuation (of control-statement line) 15
control
  by job entry control language (JECL) 237
  by operator commands 125
  by remote operator commands 125
control tables, VSE/POWER, definitions for 45
controlling spool files from operator console (shared spooling) 8
conventions 15
copies, number of
  printed output (JECL) 259
  punched output (JECL) 272
copy grouping
  by * $$ LST 259
copy separation
  generating support 48
  request by * $$ LST 264
  request by * $$ PUN 275
COPYSEP (operand of POWER) 48
count-driven output segmentation 103
CPUIDs unique for sharing systems (shared spooling) 38
creation date
  change by 135
  display by 156
  job deletion by 148
  job hold by 189
creation date (continued)
  release job by 206
cross-partition communication (see SPOOL-macro support)
CS (operand of PRMT) 65
CSALST (operand of PRMT) 66
CSAMSG (operand of PRMT) 66
CSAPUN (operand of PRMT) 67
CTCA (see channel-to-channel attachment)

D

DASD (see disk)
data block 23
  defining size of 49
  size of 27
  to DBLK group relation 27
data buffer (see data block)
data compaction
  character specification rules 77
  compaction table set, cataloging of 78
  implementation of 315
  master/non-master character relation 77, 315
  method of 317
  performance considerations 78
  request of by PALTER 137
  table set definition in JECL 258
  table set for 315
  table set, default name for 65
  table set, generation 76
  table set, generation name 76
data-driven output segmentation 104
data file
  organization of 23
  relation to queue file 22
  size of 27
data input
  from diskette (see also diskette) 280
  library members 284
  unit record 249
data length
  data block 27
  unit of transfer 27

data, insertion of in SLI member 245
date, current, display of 158
DBLK (operand of POWER) 49
DBLK group 23, 49
  display of size 158
  size of 28
  to data-block relation 28
  to queue file relation 28
DBLKGP (operand of POWER) 23, 49
DEFINE statement (autostart) 294
delays in starting of sharing systems 85
delete
  a job 146
delimit a job (JECL) 237
department number
  specification of in * $$ JOB statement 250
destination of job, changing of 138
DEV (operand of * $$ RDR) 280
device considerations
  operation 109
  planning 40
device-independent FCB image 42
device setup
  command for, local 214
device support, external (see external device support)
devices
  assignments of for VSE/POWER 84
  dummy assignments for 84
  not supported 3
  restricted support for 3
  supported 3
DFLT (operand of * $$ LST) 260
directing output to specific I/O equipment (shared spooling) 7
disk
  dump program for VSE/POWER files 323
  requirements for VSE/POWER files 22
diskette(s)
  address specification 251, 280
  data-mode processing 112, 283
  eject and feed 49
  eject instruction 281
  file identifier 281
  file verification check 281
  input * $$ RDR controlled 280
  input during execution 251
diskette(s) (continued)
  number of 281
  start processing with 220
  support 41
  SYSIN-mode processing 111, 282
  volume sequence check 221, 281
display (see status display and status report)
disposition 4, 309
  change job attributes by 135
  change of by VSE/POWER 310
  codes 309
  concept of 4
  in status display 164
  job deletion by 148
  job hold by 189
  release job by 206
disposition X queue entries
  creation of 104
  handling of 104
disposition Y queue entries
  creation of 105
  handling of 105
disposition, assignment of
  by VSE/POWER 310
  for punch output 273
  job 251
  list output 260
disposition, changing of
  by PALTER 137
  by VSE/POWER 310
  to D or K (by PRELEASE) 204
  to H or L (by PHOLD) 187
disposition, display by 155
DLBL statements (for VSE/POWER files) 25
drain receiver/transmitter 176
dummy devices, assignment of 84
dump on abnormal end of VSE/POWER 122
dump on immediate end of VSE/POWER 123, 178
dump VSE/POWER files
  example 325
  format and contents 325
  how to proceed 323
  options 324

E

eject diskette 49
empty account file 129
end VSE/POWER
command for 178
procedure 119
environment, operating 12
error correction
autostart statements 291
JECL statements 239
examples
* $$ DATA statement 287
* $$ SLI statement 287
backup for a queue 117
data compaction implementation 316
disk dump program 324
job stream, VSE/POWER start 88
load saved queue entries 118
PACCOUNT command 130
PACT command 131
PBRDCSST command 143
PCANCEL command 144
PDELETE command 150
PDISPLAY command 162
PDRAIN command 177
PEND command 179
PFLUSH command 183
PGO command 186
PHOLD command 190
PLOAD command 198
POFFLOAD command 203
PRELEASE command 207
PRESET command 210
PRESTART command 213
PSETUP command 215
PSTART (tasks) 95
PSTART command 225
PSTOP command 233
PXMIT command 235
restore backed up queue 118
saving queue entries 117
select queue entries for reload 118
start-up, autostart 89
start-up, interactively 92
status display (PDISPLAY) 163
status report 120
examples (continued)
status report (PINQUIRE) 193
VSE/POWER generation 79
3540 operation (data mode) 113
3540 operation (SYSIN mode) 111
execution (see control)
exit (see reader exit)
expansion of space 69
EXTENT statements (for VSE/POWER files) 25
external device support
benefits of 9
start a task for 221
stop a task for 229

F

FCB (see print buffer)
FEED (operand of * $$ RDR) 281
FEED (operand of POWER) 49
feed diskette 49
FID (operand of * $$ RDR) 281
file formatting
by autostart 299
interactive start 94
files of VSE/POWER
disk storage requirements 22
dumping of 323
residence of 25
sharing of 56
FLASH (operand of * $$ LST) 262
flush (see terminate)
FNO (operand of * $$ LST) 263
FNO (operand of * $$ PUN) 275
FORCE operand
PEND command 178
PSTOP command 230
forced stop
RJE,BSC line 108
VSE/POWER operation 178
form number
change job attributes by 135
changing of 138
display by 156
in status display 165
job deletion by 148

form number (continued)
  job hold by 189
  release job by 206
format
  autostart statements 299
  conventions 15
  JECL statements 240
  of operator commands 128
format queues
  autostart 90
  interactive start 94
format spool files (shared spooling) 85
FORMAT statement (autostart) 299
forms
  alignment (see forms alignment)
  for list output 263
  for punch output 275
  set-up example 215
forms alignment
  command for, local 214
forms control buffer
  image, name of in * $$ LST 261
  image, setting default for 51
  loading of 99
forms-mount request, IBM 4248 109
from node
  change job attributes by 136
  display by 156
  job deletion by 149
  job hold by 190
  release job by 207
from user
  change job attributes by 136
  display by 157
  job deletion by 149
  job hold by 190
  release job by 207
functions of VSE/POWER
  diskette input 111
  interactive start-up 92
  overview 4
  save/restore queue entries 117
  segmentation (of output) 103
  spool from tape 116
  spool to tape 115
  start-up by autostart 87
  start-up with shared spooling 85

G

generation macros
  example for coding 79
  PCPTAB 76
  PLINE 60
  PNODE 72
  POWER 47
  PRMT 63
GETVIS-area requirements (see storage requirements)

H

hardware
  BSC line characteristics 60
  device considerations 40
  devices (see devices)
  minimum for BSC 22
  minimum for SNA 21
  processors supported 2
  requirements 21
  RJE,BSC and RJE,SNA 63
HFC (operand of PRMT) 67
hold job(s) 187
horizontal-copy printing (4248) 109
horizontal format control (2770 and 2780) 67

I

I/O processing, delay of 10
ICCF (operand of * $$ SLI) 285
identification of processor 58
idle time
  on a BSC line 61
  system (shared spooling) 58
IJAFILE (see account file)
IJDFILE (see data file)
IJQFILE (see queue file)

immediate stop
  receiver task 177
  RJE,BSC line 230
  transmitter task 177
  VSE/POWER 178
in-storage copy of queue file, obtaining 160
information (see status display and status report)
initialization (see start-up)
initializing sharable spool files 38
input
  from diskette directly 221
  insertion from diskette 220
  overview 11
  spooling of to disk 219
  spooling of to tape 220
  stopping of 228
  via 3540 diskette 111
input spool device, definition of, interactively 94
inquire (see status display)
insertion, data into SLI member 245
installing VSE/POWER
  as part of VSE/SP 44
  procedures for 44
  under VSE/Advanced Functions 44
intermediate storage (see disk)
interrupting printer flow for messages 65
INTRPT (operand of PLINE) 60
IPWPOWER (name of control table) 44

J

JECL (see job entry control language)
JLOG (operand of POWER) 49
job (see job, VSE/POWER)
job accounting
  generating support 48
  identification of output 237
  storage requirement 33
job characteristics, definition of 249
job class
  use in load leveling (shared spooling) 7
job deletion
  by class 148
  by creation date 148
  by disposition 148
  by form number 148
  by from node 149
  by from user 149
  by priority 148
  by system identifier 148
  by to node 149
  by to user 149
  by user identifier 148
  command for 146
job disposition 251
job entry control language (JECL)
  * $$ CTL 244
  * $$ DATA 245
  * $$ DATA, example 287
  * $$ EOJ 248
  * $$ JOB 249
  * $$ LST 256
  * $$ PUN 271
  * $$ RDR 280
  * $$ SLI 284
  * $$ SLI, example 287
  begin of VSE/POWER job 249
  burster-trimmer-stacker option (3800) 257
  coding conventions 240
  continuation of 15
  disposition assignment 251
  end of VSE/POWER job 248
  error correction during operation 239
  format of statements 240
  loading print buffers (FCB/UCB) 99
  output operands, user-defined 294
  output segmentation, data-driven 104
  overview of 241
  purpose of 241
  restrictions 241
  statements 244
  use of 238
  user-defined output operands 294
  versus VSE job control 237
job execution
  on specific sharing system 7
job hold
  by class 189

job hold (continued)
by creation date 189
by disposition 189
by form number 189
by from node 190
by from user 190
by priority 189
by system identifier 189
by to node 190
by to user 190
job information, logging of 49
job-load balancing (shared spooling) 7
job name
defining by * $$ JOB 250
in status display 166
job number
assignment of 125, 237
in status display 166
list output 256
punch output 271
job output segmentation (see output segmentation)
job priority (see priority)
job processing
control of by JECL 237
control of by operator 125
delay of 10
job release
by class 206
by creation date 206
by disposition 206
by form number 206
by from node 207
by from user 207
by priority 206
by system identifier 206
by to node 207
by to user 207
job separation
cards, contents of 51
default for 50
list output 264
output on IBM 3800 53
overriding through PSTART SEP|NOSEP specification 217
page, contents of 50
punch output 275
sequence number in 50
job status
display of (see also status report) 151
resetting of (shared spooling) 209
job stream (example) for VSE/POWER start 88
job suffix 103
job transmission
cancellation of 182
discontinuation of 176
starting of 224
stopping of 231
job, cancellation of (by VSE/POWER) 144
job, VSE versus VSE/POWER 237
job, VSE/POWER
// OPTION NOLOG consideration 237
alter attributes (by PALTER) 132
alter attributes (by PRELEASE) 204
control by central operator 125
control from work station 125
controlling (JECL) 237
defining (JECL) 237
delete from queues 146
display status of 151
ending of (by PEND) 178
holding of 187
JECL/JCL job stream example 238
logging (JECL) 237
multitask, support for 223
number, assignment of 125, 237
password for 254
release from hold/leave state 204
restart output of 211
routing of for execution 255
segmentation of output 103
stop (by PFLUSH) 180
stopping task for 229
submitting 96
without JECL 237
jobs, multiple VSE/POWER 237
JSEP (operand of POWER) 50

K

keyword operands 16
keyword operands, user-defined 294
KILL (see FORCE)

L

label information in spool files (shared spooling) 38
leased line connection 61
leave job(s) 187
LFCB command, use of 96
LIB (operand of * $$ SLI) 285
library-member inclusion (see source library-member inclusion)
line (see also BSC line)
  BSC, definition of 60
  charges, how to save 61
  password, BSC (in PSTART) 224
  stopping of 230
  termination of 107
  turnaround 70
line skip control (see also forms control buffer)
  by * $$ LST 265
  default 51
LIST (operand of PRMT) 67
list output
  blanks for X'01' through X'3F' 70
  definition of 256
  definition of by * $$ LST 271
  delay of 103
  disposition assignment 260
  number of copies 259
  number of lines before checkpoint 267
  number of lines before warning 267
  password for 266
  predefined data (3800) 266
  restart of 211
  routing of 268
  routing of by * $$ JOB 252
  routing of by * $$ LST 259
list output (continued)
  starting of 217
  suppress spooling of 261
loading
  network definition table 197
  print buffers (FCB/UCB) 99
  queue entries from tape 199
LOCAL (operand of PNODE) 74
local node 74
lock file (for shared spooling) 38
logging job information 49, 237
logical unit names, secondary 67
LST (operand of * $$ LST) 264
LSTROUT (operand of PRMT) 67
LTAB (operand of POWER) 51
LU (operand of PRMT) 67

M

machine requirements 13
macros
  generation 45
  PCPTAB (generation) 76
  PLINE (generation) 60
  POWER (generation) 47
  PRMT (generation) 63
magnetic tape considerations 41
mark form function (3800) 53
MASTER (operand of PCPTAB) 76
master characters 76, 317
MAXBUF (operand of PNODE) 74
maximum resource support 14
MEM (perand of * $$ SLI) 284
message page eject 68
messages
  deletion of 149
  display of 158
  notify (see notify support) 54
  space-3 command after 68
  to nodes/remote users 141
  unanswered 158
migration considerations 19
MODIFY (operand of * $$ LST) 266
MODSET (operand of PLINE) 61
MRF (operand of PRMT) 68

MRKFRM (operand of POWER) 53
MSG (operand of PRMT) 68
MSGEJCT (operand of PRMT) 68
MSGSPCE (operand of PRMT) 68
multifunction card machine
  operation of 110
  supported 40
multiple channel-12 posting 53
multiple record feature, IBM 2780 68
multiple VSE/POWER jobs in VSE job 237
multipurpose unit-record device 110
multitask job support 223
multivolume tape segmentation 104
MULT12 (operand of POWER) 53

N

n.n iii
name (see phase name)
NCP generation requirements 39
NDT (see network definition table)
network definition table
  defining 72
  display of 159
  loading of 197
  new (by autostart) 307
  phase name 72
networking function
  activate/reactivate transmitter or receiver 131
  alter destination 138
  buffers for 74
  command authorization 73
  control of operations 125
  deactivate receiver task 176
  deactivate transmitter task 176
  end for a queue entry 182
  hardware characteristics (PNET,BSC) 60
  job input, routing of 255
  line password 61, 224
  link, setting up a 224
  load network definition table 197
  messages, routing of 141
  NDT, new (by autostart) 307
  node name 224
networking function (continued)
  object deck transmission 62
  password for 75, 224
  performance/problem trace 224
  request support for 54
  route notify messages (JECL) 252
  route operator commands 234
  routing list output by * $$ JOB 252
  routing list output by * $$ LST 259
  routing punch output (by * $$ PUN) 272
  routing punch output by * $$ JOB 253
  SEND/RECEIVE trace 224
  session controls for 72
  session, starting a 224
  status of node 192
  stop a connection, link, or session 231
  stop for a queue entry 182
  storage requirement 33
  table for (see network definition table) 72
  time out 61
  transmission route 75
NOD (operand of * $$ RDR) 281
NODE (operand of PNODE) 72
node identifier (see node name)
node name (see also fromnode/tonode)
  define to VSE/POWER 72
  specification in PSTART 224
node password
  definition of 75
  specification in PSTART 224
NOMASTn (operand of PCPTAB) 76
non-master characters 76, 315
non-printable characters transmission 70
NORUN (operand of SET) 307
notify support
  definition for 54
  routing of by * $$ JOB 252
  storage requirements 33
NTFYMSG (operand of POWER) 54
number assigned to a job 237

O

object deck transmission 70
object deck, transmission over BSC line 62
offload function
  BACKUP 117
  command for 199
  LOAD 118
  processing considerations 117
  SAVE 117
  SELECT 118
  stopping of 119
onload queue files 118, 200
operating environment 12
operating with VSE/POWER
  at central system 125
  at remote work station 125
  authorization in network 73
  end of 178
  JECL statements, error correction 239
  stop a local task 228
  stop a network connection 231
  stop a partition 229
  stop a tape function 229
  stop an external device 229
  stop an RJE,BSC line 230
  stop an RJE,SNA task 231
operations
  ending of 119
  start a partition (interactively) 94
operator command routing (networking) 234
operator commands (see central operator commands)
operator console
  controlling spool files from (shared spooling) 8
options of VSE/POWER
  ACF/VTAM interface, set-up 106
  class change 98
  commands for autostart 291
  commands for start 84
  commands, how to enter 96
  error correction, autostart statements 291
options of VSE/POWER (continued)
  FCB loading 99
  formatting files (by autostart) 299
  job-identification display 97
  jobs, submission of 96
  multifunction unit-record device 110
  networking function 83
  output segmentation 103
  output segmentation, data-driven 104
  overview 4
  partition priority change 98
  planning of 38
  reader task, start of 100
  RJE,BSC line, start of 106
  RJE,SNA start 106
  RJE,SNA, stopping of 108
  shared-spooling function 83
  spool-tape processing 114
  start a partition (by autostart) 91
  start-up (by autostart) 87
  start-up (of VSE/POWER) 83
  start-up (with shared spooling) 85
  start-up interactively 92
  system console 83
  task control 100
  tasks, stopping of 101
  with 3540 diskette (data mode) 112
  writer task, start of 101
  3540 diskette (SYSIN mode) 111
organization
  account file 25
  data file 23
  queue file 22
output (see also list/punch output)
  copy separation 48
  device selection, remote work station 65
  directed to specific I/O equipment (shared spooling) 7
  from tape, starting of 218
  job separation, specifying 217
  job status, requesting 151
  list, definition of 256
  list, disposition for 260
  overview 11
  punch, definition of 271
  punch, disposition for 273
  restart of 211
  segmented (staged) 103

output (see also list/punch output)
(continued)
spooling of to tape 268, 278
status report, cancellation 144
stop (by PFLUSH) 180
stopping of 228
to device under VM, starting of 217
to external device, start of 221
to unit record, starting of 217
output operands, user-defined 294
output segmentation
card count for 277
count-driven 103
data-driven 104
list output, data-driven 256
macro SEGMENT 104
number of cards/pages per 55
overview 103
page count for 267
program-driven 104
punch output, data driven 271
spool tape at EOV 115
types of segmentation 103
output spool device, definition of,
interactively 94
for using VSE commands 96
output transmission
cancellation of 182
discontinuation of 176
starting of 224
stopping of 231
overviews
central operator commands 127
disk dump for VSE/POWER files 323
functions, JECL controlled 241
generation macros 45
JECL statements 241
output segmentation 103
storage requirement 33
VSE/POWER concept and purpose 1

P

PACCOUNT command 129
PACT command 131
page buffer, IBM 3800, clearing of 48
page count
for restart 212
in status display 164
PALTER command 132
partition priority 12
changing of 98
partition, VSE/POWER
alter input class 132
device assignments for 84
priority assignment for 12
release to VSE operation 178
start of 93
storage for options 33
partition, VSE/POWER-controlled
priority assignment for 12
reactivation of 184
relationship to class 226
starting (by autostart) 91
starting (interactively) 94
starting a, command for 222
stop/restart procedure for 98
stopping of, command for 229
password
BSC line, definition by PLINE
macro 61
for list output (in * $$ LST) 266
for punch output (in * $$ PUN) 276
job (in * $$ JOB) 254
log-on, BSC work station 223
log-on, SNA work station 68
network connection 224
node, definition of 75
RJE,SNA log-on 68
PAUSE (operand of POWER) 54
PBRDCST command 141
PCANCEL command 144
PCPTAB (generation macro) 76
PDELETE command 146
PDISPLAY command 151
PDRAIN command 176
PEND command 178

performance consideration
  data-block size 28
  data block, size of 27
  DBLK group, size of 28
  storage vs. throughput 15
PFLUSH command 180
PGO command 184
phase name (see also node name)
  compaction table set 76
  FCB image, device independent 42
  for POWER macro assembly 47
  network definition table 72
  reader exit 55
PHOLD command 187
PINQUIRE command 192
planning
  account-file size 29
  data-file size 27
  device considerations 40
  options (for VSE/POWER) 38
  overview 21
  queue-file size 25
  VIO space 25
PLINE (generation macro) 60
PLOAD command 197
PNET (operand of POWER) 54
PNET function (see networking function)
PNODE (generation macro) 72
POFFLOAD command 199
polling time (shared spooling) 58
positional operands 16
POWER (generation macro) 47
POWER (name of control table) 44
pre-generated VSE/POWER 44
PRELEASE command 204
PRESET command 209
PRESTART command 211
PRI (operand of POWER) 55
print band verification for IBM 4248 109
print buffer
  FCB image name (JECL) 261
  FCB, device independent 42
  loading of 99
  page, clearing of 48
  UCB image name (JECL) 269
  UCB, loading of 99
print positions
  per line 67
print positions (continued)
  per message 68
print setup pages 214
printer assignment
  autostart 91
  by * $$ LST 264
  interactive start 94
printer select character 66
PRINTERS autostart statement 301
priority
  change by 135
  default 55
  display by 156
  in status display 164
  job deletion by 148
  job hold by 189
  of a job, alter 138
  of partitions 12
  release job by 206
  setting of by * $$ JOB 253
  setting of by * $$ LST 266
  setting of by * $$ PUN 276
PRMT (generation macro) 63
problem situation 323
procedures
  commands, how to enter 96
  generate VSE/POWER 81
  installation 44
  loading FCB and UCB 100
  shut down (of VSE/POWER) 119
  start-up by autostart 87
  start-up interactively 92
  start-up with shared spooling 85
  stop/restart partition 98
  stopping a BSC line 107
processor identification 58
processor storage (see storage requirements)
processors supported 2
program class (see class)
program-driven output segmentation 104
programmer name
  specification of in * $$ JOB statement 253
programming requirements 13
protected output entry (spool-access support) 105
PRTY command, use of 96

PSETUP command 214
PSTART command 216
PSTOP command 227
PSWRD (operand of PLINE) 61
PSWRD (operand of PRMT) 68
PUN (operand of * $$ PUN) 276
PUN (operand of PRMT) 69
punch assignment
  autostart 91
  by * $$ PUN 276
  interactive start 94
punch output
  definition of by * $$ PUN 271
  delay of 54, 103
  disposition assignment 273
  number of cards before checkpoint 276
  number of cards before warning 277
  number of copies 272
  password for 276
  restart of 211
  routing of by * $$ JOB 253
  routing of by * $$ PUN 272, 277
  routing of to work station 69
  starting of 217
  suppress spooling of 274
punch-writer task delay 54
PUNCHES autostart statement 303
PUNROUT (operand of PRMT) 69
purpose of VSE/POWER 1
PWD (operand of PNODE) 75
PXMIT command 234

Q

queue
  backup of on tape 199
  control of 125
  delete job in 146
  entry 24
  formatting of (autostart) 90
  formatting, interactively 94
  record 24
queue entry
  display of from tape 199
  loading from tape 199
queue entry (continued)
  loading of individual 199
  saving of on tape 199
  selecting of for POFFLOAD 201
queue file
  copy of in VIO space 23
  free records in, display of 158
  organization of 22
  relation to data file 22
  size of 25
  virtual-storage copy of 23
queue management 125
quit-and-lock request (spool-access support) 105

R

RBC (operand of * $$ LST) 267
RBC (operand of * $$ PUN) 276
RBM (operand of * $$ LST) 267
RBM (operand of * $$ PUN) 277
RBS (operand of * $$ LST) 267
RBS (operand of * $$ PUN) 277
RBS (operand of POWER) 55
RDREXIT (operand of POWER) 55
READER autostart statement 305
reader exit
  defining name of 55
  storage requirement 33
real storage (see storage requirements)
receiver task
  buffers for, number of 74
  deactivation of 176
  reactivate 131
  stop for a queue entry 182
record count
  for restart 212
  in status display 164
record length
  diskette input unspooled 251
  diskette input, spooled 282
  for remote job entry 69
records before warning message
  card-image (punch) 57
  list (print) records 57

recovery of spool files (shared spooling) 85, 86, 123
REF (operand of PRMT) 69
reference definition of work station 69
relation, queue to data file 22
release number iii
remid (remote identifier) 64
REMOTE (operand of * $$ LST) 268
REMOTE (operand of * $$ PUN) 277
REMOTE (operand of PRMT) 64
remote job entry BSC
  commands for 107
  component selection 65
  hardware characteristics 60
  line password, default definition 61
  line status request 192
  line turnaround 70
  loading of support 55
  log-on password 223
  message page eject 68
  object deck transmission 62
  operating 96
  print positions per line 67
  print positions per message 68
  record length 69
  routing list output to 67
  routing punch output to 69
  space compression/expansion (2770 and 3780) 69
  space-3 command after message 68
  start a line 106
  stopping a line 107
  storage requirement 33
  suppress loading of support 55
  time out for 61
  transparency 70
  work station identification 64
  work station type 64
  work station, definition of 63
  2770 and 2780 horizontal format control 67
remote job entry functions
  control of operations 125
  job status, display of 155
  messages, sending of 141
  starting task for 223, 224
  stopping task for 230
remote job entry SNA
  ACF/VTAM considerations 39
remote job entry SNA (continued)
  commands for 107
  compaction-table definition 65
  generating support 56
  log on by logical unit name 67
  log-on password 68
  logical unit status request 192
  operating 96
  printer-flow interruption 65
  record length 69
  routing list output to 67
  routing punch output to 69
  sessions, number of 69
  starting a session 106
  stopping a session 108
  storage requirement 33
  work station identification 64
  work station type 64
  work station, definition of 63
  work stations, number of 56
  X'01' through X'3F' translation 70
remote operator commands 125
reports (see status report)
requirements (see also storage requirements)
  disk storage for VSE/POWER files 22
  for ACF/VTAM and ACF/NCP 39
  for shared spooling 38
  hardware 21
  machine 13
  programming 13
  storage, minimum 14
  storage, processor 29
  storage, virtual 29
  when using BSC 22
  when using SNA 21
reserved RJE user identifiers 222
reset job status (shared spooling) 209
residence (in library) 43
restart
  by PGO 184
  of output by PRESTART 211
  partition, example for 98
restart, automatic (networking) 224
restrictions
  accessing a job 254
  alter attributes/input class 132
  backup queues 200
  CCB-CCW address 12

restrictions (continued)
COPY and RBS operands of * $$ LST 268
copy-group handling, IBM 3800 137
data areas 12
Data chaining 12
EXCP real 12
IDA flag 12
JECL statements, writer-only partition 243
messages, routing of 141
multifunction-card assignments 40, 84
multivolume SYSIN-tape 116
nesting of SLI statements 286
PACCOUNT command 129
PNET,SNA use 39
queue-entry backup 200
remote job entry 19
saving queue entries of the transmit queue 199
segmentation on a 3525 55
shared spooling 19
stacker selection 51
SYSBUFLD under VSE/POWER 109
3800 copy-group handling 137
RJE (see remote job entry)
RJE work station type, specifying 64
RJEBSC (operand of POWER) 55
room number
specification of in * $$ JOB statement 254
ROUTE1 (operand of PNODE) 75
ROUTE2 (operand of PNODE) 75
routing
list output, RJE 67
operator commands (networking) 234
punch output, RJE 69

S

save account file
to disk 129
to tape 130
save queue entries 199
save queue files 117
SCE (operand of PRMT) 69
secondary logical unit, name of 67
segmentation (see output segmentation)
selecting of queue entries for POFFLOAD 118, 201
selector channel interrupt 60
separation of output copies
default specification 48
list output 264
punch output 275
separator cards/pages (see job separation)
sequence number
of job separator 50
session (between nodes), controls for 72
sessions, SNA, number of 69
SESSLIM (operand of PRMT) 69
SET CPUID command (shared spooling) 38
SET statement (autostart) 307
set up forms (see forms alignment)
sharable spool files 38
SHARED (operand of POWER) 56
shared DASD (for shared spooling) 38
shared spooling
abnormal end 123
benefits of 7
cold start 85
controlling spool files from operator console 8
generating support 56
initialization with 85
job class selection 8
processor identification 58
recovery of spool files 86
reset job status 209
selecting job classes 8
setting active/idle/polling time 58
sharing the job account file 9
start-up with 85
storage requirement 33
SYSID for job execution 254
SYSID for list output 268
SYSID for punch output 278
system requirements 38
updating control information in spool files (shared spooling) 8
warm start 86

sharing systems
  delays in starting 85
sharing user files 7
sharing VSE/POWER libraries 38
shipment of VSE/POWER 43
shut down (of VSE/POWER)
  command for 178
  procedure for 119
single character string 70
size of
  account file 29
  data block 27
  data file 27
  DBLK group 28
  queue file 25
  transmission buffers 74
  VIO space 25
  VSE/POWER partition 29
SKIP (operand of SET) 307
skip-to-channel 1 262
  by VSE/POWER 307
  specification in PSTART command 219
SLI (see source library-member inclusion)
SNA (operand of POWER) 56
SNA work station (see remote job entry SNA)
SNA, minimum configuration for 21
source library-member inclusion
  add statements to member 286
  cataloging member 287
  delete statements from member 286
  generation of support 52
  JECL statements in member 286
  nesting a member 286
  rules for coding a member 286
  specifications for 284
  storage requirement 36
space compression/expansion 69
space-3 commands after message 68
SPLDMOHP field (spool-access support) 105
SPLIM (operand of POWER) 57
SPOOL (operand of POWER) 57
spool-access support 9
spool account file to punch queue 129
spool device
  definition (by autostart) 90
  definition by interactive start 94
spool device (continued)
  multifunction unit-record device 110
  PRINTERS autostart statement 301
  PUNCHES autostart statement 303
  READER autostart statement 305
  tape unit 114
  3540 diskette 111
spool devices, specification of 91
spool file formatting 85
spool file recovery (shared spooling) 86, 123
spool files (shared spooling)
  controlling from operator console 8
  formatting 85
  updating control information in 8
SPOOL-macro support
  generation of support 57
  storage requirement 33
spooling to tape 114
staged output 103
start-up
  automatic 87
  commands used for 84
  interactive 92
  of VSE/POWER 83
  with shared spooling 85
starting (see also start-up)
  a partition (by autostart) 91
  a partition (interactively) 94
  reader task 100
  writer task 101
status (see status display/report)
status display
  account file 158
  active task 168
  active tasks 159
  appendable jobs 154
  BSC line 192
  burster indication 166
  by class 155
  by creation date 156
  by disposition 155
  by form number 156
  by from node 156
  by from user 157
  by priority 156
  by system identifier 156
  by to node 157
  by to user 157

status display (continued)
by user identifier 156
cancel output of 144
copy-group count 166
copy(-transmission) count 165
current date/time 158
DBLK and DBLK group size 158
disposition 164
example, PDISPLAY A 168
example, PDISPLAY LST 164
example, PDISPLAY LST,...,
FULL=YES 172
example, PINQUIRE ALL 193
example, PINQUIRE DEV 196
example, PINQUIRE NODE 195
external device 192
fixed pages 158
for FULL=YES request 172
for PDISPLAY A 168
for PDISPLAY LST 164
for PDISPLAY PNET,ALL 174
form number 165
forms overlay 166
free DBLK groups 158
free queue records 158
header for 164
incomplete jobs 154
job characteristics 151
job class 164
job name/job number 154, 166
local (PDISPLAY) 151
local-destination jobs 155
logical unit 192
messages 158
network definition table 159
node 192
non-local (PINQUIRE) 192
of POFFLOAD or spool tape 161
of virtual-storage copy of queue
file 160
output characteristics 151
output class 164
page/record count 164
pages fixed 158
priority 164
processable jobs 154
requesting full 156
RJE submitted jobs 155
stop output of 144
status display (continued)
system identifier 164
task, active 168
tasks, active 159
trace area addresses 158
trimmer indication 166
status report
at start up 88
on shut down (by PEND) 120, 178
status, reset (shared spooling) 209
STDCARD (operand of POWER) 57
STDLINE (operand of POWER) 57
STOP command, use of 96
stopping
a local task 228
a network connection 231
a partition 229
a tape function 229
an RJE,BSC line 230
an RJE,SNA task 231
external device 229
job processing 144, 180
job transmission 176
output processing 180
output transmission 176
status-report output 144
VSE/POWER operation 119, 178
storage (see processor or virtual
storage)
storage pages PFIXed 158
storage requirements
estimating 29
fixable area 35
GETVIS area 35
job accounting 33
networking support 33
notify support 33
reader-exit support 33
RJE,BSC support 33
RJE,SNA support 33
shared spooling 33
spool access support 37
SPOOL-macro support 33
VSE/POWER nucleus 33
3800 printer support 33
store queue entries on tape 199
structure of VSE/POWER (see tasks)
SUBLIB (operand of POWER) 58

sublib.bookname (operand of * $$ SLI) 285
suffix, job 103
summary of amendments xiii
support
  hardware 2
  maximum number of resources 14
SWITCH (operand of PLINE) 61
switched line
  definition for 61
  time-out for 61
SYSBUFLD program, use of 96
SYSID (operand of POWER) 58
SYSID, new (by autostart) 307
SYSIN input from diskette
  specification for 282
  specification in * $$ JOB 251
SYSRDR assignment 93
system considerations 12
system identifier
  change job attributes by 136
  define to VSE/POWER 58
  display by 156
  display of 158
  in status display 164
  job deletion by 148
  job hold by 189
  new (by autostart) 307
  release job by 206
  selecting by * $$ JOB 254
  selecting by * $$ LST 268
  selecting by * $$ PUN 278

T

table generation (see generation macros)
tape processing
  backup a queue 199
  cancel for a queue entry 180
  display of queue entries 161, 199
  input from 116
  load a queue 199
  output segmentation 115
  output to 115
  save a queue 199
  stopping task for 229
tape spooling
  list output 268
  punch output 278
tape support considerations 41
tape SYSIN function 116
task (see also receiver or transmitter task)
  cancel a queue entry 180
  control of 125
  controlling of 100
  display of 159
  management 125
  reactivation of 184
  reader task, start of 100
  starting of 94, 216
  stopping of 101, 228
  writer, start of 101
tasks, VSE/POWER, overview 10
terminate (see stopping)
termination (of VSE/POWER) 119
terms, definition of (see Glossary)
throughput considerations 15
TIME (operand of POWER) 8, 58
time, current, display of 158
TIMEOUT (operand of PLINE) 61
to node
  change job attributes by 136
  display by 157
  job deletion by 149
  job hold by 190
  release job by 207
to user
  change attributes by 136
  display by 157
  job deletion by 149
  job hold by 190
  release job by 207
trace area addresses, display of 158
trace, BSC and SNA
  define buffer for 59
  network SNA I/Os 225
  RJE,BSC I/Os 223
  SNA network SEND/RECEIVE 224
  start dynamically 225
  stop dynamically 233
  stop storing of 233
  store event records 225
TRACESZ (operand of POWER) 59

transmission
  buffers, number of 74
  buffers, size of 74
  code 60
  queue, backup for 201
  queue, saving of 201
  start of 224
  stop for a queue entry 182
transmitter task
  activate/reactivate 131
  buffers for, number of 74
  deactivation of 176
  stop for a queue entry 182
transparency
  define support 62
  work station with 70
TRNSP (operand of PLINE) 62
TRNSP (operand of PRMT) 70
turnaround, line 70
TURNEOJ (operand of PRMT) 70
TYPE specification
  of work station (in PRMT) 64

U

UCB (see print buffer)
unanswered messages 158
unit of transfer (see also buffers)
  data file, local spooling 49
  local spooling, size of 27
unit-record I/O equipment
  centralization of (shared spooling) 7
universal character set buffer
  load request by * $$ LST 258, 269
  loading of 99
updating control information in spool files (shared spooling) 8
user-defined output operands 294
user identifier
  alter 139
  deletion by 148
  display by 156
  for list output 270
  for punch output 279
  format of specification 16
user information
  specification of 254
user-written channel programs for IBM 4248 110
userid (see user identifier)
users, types of 16

V

VER (operand of * $$ RDR) 281
version number iii
VIO space
  copy of queue file in 23, 160
  dump of 122, 160
  estimating size of 25
virtual storage (see storage requirements)
virtual-storage copy of queue file, obtaining 122, 160
VM output task 217
volume checking, diskette 221
volume sequence, diskettes 281
VSC (operand of * $$ RDR) 281
VSE job, VSE/POWER jobs in 237
VSE/ICCF notify support (see notify support)
VSE/POWER files (see files)
VSE/POWER job 237
VSE/POWER table generation
  examples 79
  macros for 45
  procedure for 81
  requirements for PNET,SNA 39
  requirements for RJE,SNA 39
VSE/POWER, pre-generated 44
VSE/POWER, Version 1 and 2 compatibility 19

W

warm start
  example (autostart) 90
  interactively 94
  job number after 83
  specification (in FORMAT statement) 299
  with shared spooling 86
work stations
  defining of to VSE/POWER 63
  definition by reference 69
  log-on password, SNA 68
  number of, logged on 56
  record length 69
  routing list output to 67
  routing punch output to 69
  time out for 61
  transparency feature 70
  type, defining of 64

X

X (see disposition X queue entries) 104
X'01' through X'3F' as blanks 70
XDEST operand (of * $$ JOB) 255
XLATE (operand of PRMT) 70
XMT (transmission) queue
  backup for 201
  direct output to by * $$ PUN 274
  directing job input to by * $$ JOB 255
  directing list output by * $$ LST 259
  directing list output to by * $$ JOB 252
  directing punch output to by * $$ JOB 253
  saving of 201

Y

Y (see disposition Y queue entries) 105

Numerics

2560 considerations 40
2701 control unit specifications (in PLINE) 61
2770
  buffer expansion 65
  buffer expansion, additional 65
  horizontal format control 67
  print positions per line 67
  space compression/expansion 69
2780
  horizontal format control 67
  multiple record feature for 68
  print positions per line 67
3200 relation to 3800 iii
3525 considerations 40
3540 diskette I/O unit
  input * $$ JOB controlled 251
  input * $$ RDR controlled 280
  scope of support 41
3741, additional buffer expansion 65
3741, print positions per line 67
3780
  horizontal format control 67
  print positions per line 67
  space compression/expansion 69
3800 printing subsystem
  burster-trimmer control 257
  character arrangement tables 258
  copy grouping for 259
  default setting for (JECL) 260
  FCB-image name for 261
  forms-overlay frame (JECL) 262
  job separation 53
  output to, definition of 256
  page buffer, clearing of 48
  predefined data on 266
  relation to 3200 iii
  restart consideration 212

3800 printing subsystem (continued)
   storage requirement 33
4248
   device considerations 41
   forms-mount request 109
   horizontal-copy printing (4248) 109
   print band verification 109
   user-written channel programs 110
5424/5425 considerations 40

IBM Virtual Storage Extended/
POWER
Installation and Operations Guide
Order No. SH12-5329-06

READER'S COMMENT FORM

This form may be used to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Your comments:

Note: **Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.**

If you wish a reply, give your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page).

SH12-5329-06

CUT
OR
FOLD
ALONG
LINE

Reader's Comment Form

Fold And Tape **Please Do Not Staple** Fold And Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold And Tape **Please Do Not Staple** Fold And Tape

If you would like a reply, please print:

Your Name _____

Company Name/Department _____

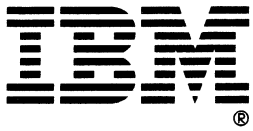
Street Address _____

City _____

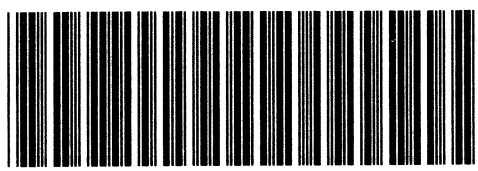
State/Zip Code _____

IBM Branch Office serving you _____

IBM®



SH12-5329-06



IBM Virtual Storage Extended/
POWER
(File No. S370/4300-36)
Printed in U.S.A.