IBM z/VSE
VSE Central Functions

VSE/POWER
Administration and Operation

Version 7 Release 1

SC33-8247-00

IBM z/VSE
VSE Central Functions

VSE/POWER
Administration and Operation

Version 7 Release 1

SC33-8247-00

Note!
Before using this information and the product it supports, be sure to read the general information under “Notices” on page xiii.

First Edition (March 2005)

This edition applies to Version 7 Release 1 of IBM VSE/POWER, which is part of VSE Central Functions, Program Number 5609-ZVS, and to all subsequent releases and modifications unless otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the addresses given below.

A form for readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

```
IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany
```

You may also send your comments by FAX or via the Internet:

```
Internet: s390id@de.ibm.com
FAX (Germany): 07031-16-3456
FAX (other countries): (+49)+7031-16-3456
```

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Notices

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 product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

Any pointers in this publication to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement. IBM accepts no responsibility for the content or use of non-IBM Web sites specifically mentioned in this publication or accessed through an IBM Web site that is mentioned in this publication.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

```
IBM Deutschland GmbH
Department 0790
Pascalstr. 100
70569 Stuttgart
Germany
```

Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Programming Interface Information

This manual is intended to help the customer with planning, installation, operation, and diagnostic tools of IBM VSE/POWER.

It also documents intended programming interfaces that allow the customer to write programs to obtain the services of VSE/POWER. Such information is enclosed in brackets as follows:

Programming Interface Information

End of Programming Interface Information

Trademarks and Service Marks

The following terms are trademarks of International Business Machines Corporation in the United States or other countries, or both:

ACF/VTAM
Advanced Function Printing
AFP
AS/400
CICS
CICS/VSE
Enterprise Systems Architecture/370
Enterprise Systems Architecture/390
ES/9000
ESA/390
IBM
OS/390
Print Services Facility
System/370
VM/ESA
z/VSE
VTAM

About This Book

z/VSE is the successor to IBM's VSE/ESA product. Many products and functions supported on z/VSE may continue to use VSE/ESA in their names.

z/VSE can execute in 31-bit mode only. It does not implement z/Architecture, and specifically does not implement 64-bit mode capabilities.

z/VSE is designed to exploit select features of IBM eServer zSeries hardware.

This manual introduces IBM VSE/POWER and is intended to help the reader with planning, installation, operation, and diagnostic tools of IBM VSE/POWER, the spooling system of z/VSE. In addition, the manual contains all reference information for VSE/POWER.

Who Should Use This Book

This manual addresses system administrators, system programmers and operators who plan, install and operate VSE/POWER. The intended audience of this manual consists therefore of users who have familiarity in the following areas:

- User level knowledge of z/VSE
- Knowledge of IBM assembler language.

How to Use This Book

The following list tells where you find information on various aspects of VSE/POWER described in this manual.

Chapter 1, "Understanding Syntax Diagrams," on page 1 describes how to read the syntax diagrams in this manual.
Chapter 2, "Planning for VSE/POWER," on page 5 introduces you to VSE/POWER and shows what to plan for.
Chapter 3, "Tailoring VSE/POWER," on page 55 describes the VSE/POWER generation macros.
Chapter 4, "Operating with VSE/POWER," on page 97 gives operator information; it explains how to start and stop VSE/POWER and how to use its operator commands.
Chapter 5, "VSE/POWER Operator Commands," on page 191 gives a complete reference of the VSE/POWER operator commands. The symbols of command language syntax are listed at the beginning of this chapter.
Chapter 6, "JECL Statements," on page 397 gives a complete reference of the JECL statements.
Chapter 7, "VSE/POWER Autostart Statements," on page 471 gives a complete reference of the VSE/POWER autostart statements.

The following information is included in the appendixes:

Appendix A, "VSE/POWER Disposition Codes," on page 497 explains the purpose of disposition codes and lists the conditions that cause VSE/POWER to change a disposition code.
Appendix B, "VSE/POWER Diagnostic and Service Aids," on page 503 introduces the available VSE/POWER diagnostic tools.

Appendix C, "RJE, SNA Data Compaction," on page 533 discusses data compaction, a function to reduce line traffic under VSE/POWER.

Additional help is provided at the back of the book:
The glossary explains technical terms used in the manual.
The index helps you to locate information.

Where to Find More Information

The following IBM manuals also describe aspects of VSE/POWER:
- *VSE/POWER Remote Job Entry*, SC33-6734
- *VSE/POWER Networking*, SC33-8249
- *VSE/POWER Application Programming*, SC33-8248

The VSE/POWER messages are listed in *z/VSE Messages and Codes*, SC33-8226, SC33-8227, SC33-8228.

For z/VSE you might occasionally need the following IBM manuals:
- *z/VSE Planning*, SC33-8221
- *z/VSE Installation*, SC33-8222
- *z/VSE Administration*, SC33-8224
- *z/VSE Operation*, SC33-8239
- *z/VSE Networking Support*, SC33-8235
- *z/VSE Guide to System Functions*, SC33-8233
- *z/VSE System Control Statements*, SC33-8225
- *z/VSE System Macros Reference*, SC33-8230
- *VSE/ESA Extended Addressability*, SC33-6621
- *VSE/ICCF Administration and Operation*, SC33-6738

For information on VTAM, see
- *Planning for NetView, NCP, and VTAM*, SC31-7122

For more information on TCP/IP, see
- IBM Redbook *Getting Started with TCP/IP for VSE/ESA 1.4* , SG24-5626.
- *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information* , SC33-6601.

For a book on High Level Assembler, refer to *HLASM V1R4 Programmer's Guide* , SC26-4941.

For the layout of NJE control records, refer to *Network Job Entry Formats and Protocols* , SC23-0070.

z/VSE Home Page

z/VSE has a home page on the World Wide Web, which offers up-to-date information about VSE-related products and services, new z/VSE functions, and other items of interest to VSE users.

You can find the z/VSE home page at:

`http://www.ibm.com/servers/eserver/zseries/zvse/`

VSE/POWER Home Page

VSE/POWER also has a home page that offers current information on VSE/POWER. You can find the VSE/POWER home page at:

`http://www.ibm.com/servers/eserver/zseries/zvse/support/power/power.htm`

Abbreviations

```
ACB      =  access method control block
ACF      =  Advanced Communication Function
API      =  application program interface
ASA      =  records with American National Standard control characters
ASCII    =  American National Code for Information Interchange
ASI      =  automated system initialization
BAM      =  Basic Access Method
BDT      =  Bulk Data Transfer (an IBM licensed program)
BMS      =  basic mapping support (used by CICS)
BSC      =  binary synchronous communication
BSD      =  Berkeley Software Distribution
BSM      =  Basic Security Manager
CCB      =  channel control block
CCW      =  channel control word
CICS/VSE =  Customer Information Control System/VSE (an IBM licensed program)
CPDS     =  composed page data stream (also 'all-point addressable records')
CSI      =  Connectivity Systems Incorporated
CTC      =  channel-to-channel
CTCA     =  channel-to-channel adapter
DBLK     =  data block
DSHR     =  data set header record
DTF      =  Define the File (BAM control block)
EBCDIC   =  extended binary-coded decimal interchange code
ESC      =  escape mode (the format is user defined)
ESM      =  External Security Manager
FBA      =  fixed-block architecture (disk format)
FCB      =  forms control buffer (for printer control)
ICCF     =  Interactive Computing and Control Facility
ID       =  identifier/identification
IP       =  Internet Protocol
IPL      =  initial program load (of operating systems)
JCL      =  job control language
JECL     =  job entry control language
JES      =  Job Entry System (the MVS spooling component)
KB       =  Kilobyte (=1024 bytes)
LTA      =  logical transient area
MB       =  Megabyte (=1024KB)
MCC      =  machine command code
MVS      =  Multiple Virtual Storage (a base element of OS/390)
NAT      =  node attached table
NCP      =  Network Control Program
NDT      =  network definition table
NJE      =  network job entry
NJI      =  network job-interface
OPTB     =  output parameter text block
OS/390   =  mainframe operating system (formerly known as MVS)
PNET     =  Power NETworking
PSF      =  Print Services Facility (an IBM Licensed Program)
RJE      =  remote job entry
RSCS     =  Remote Spooling Communication Subsystem
SAS      =  Spool Access Support
SCB      =  string control byte (data compaction)
SCS      =  standard character string
SDLC     =  synchronous data link control
SEH      =  Spool Environment Header record
```

```
SER      =  Spool Environment Record
SNA      =  system network architecture
SSL      =  Secure Sockets Layer
SVA      =  system virtual area
TCP/IP   =  Transmission Control Protocol/Internet Protocol
TOD      =  time-of-day
TRTC     =  test compaction table
UCB      =  Universal character set buffer
USS      =  unformatted system services (a VTAM service)
VIO      =  Virtual I/O storage space (used for queue file copy)
VM       =  Virtual Machine (a type of IBM operating systems)
VSE/SAM  =  Virtual Storage Extended/Sequential Access Method
VTAM     =  Virtual Telecommunications Access Method
```

Throughout this manual:

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

Summary of Changes, VSE/POWER 7.1

The following summarizes changes made to VSE/POWER 7.1.

Extension of Functional Support

This section summarizes the changes made to provide enhanced functionality.

Reducing System Down Time

Now finally with VSE/POWER 7.1 existing functions have been improved and new functions have been added to close all known gaps where a cold start[1] of the VSE/POWER spool files has been required in the past, and where costly time had to be spent in offloading of queues, reformatting spool files extents and in reloading the queues. Instead, during a VSE/POWER **warmstart** the following functions can be performed without a noticeable loss in system-up time:

- "**Release Migration During Warmstart**" for VSE/POWER 6.7 (and later) spool files. For details see "Upward Migration to VSE/POWER 7.1" on page 18.
- "Extending the Data File During Warmstart" — introduced with Version 6.6 for for one extent, expanded in Version 7.1 to "any" number of data file extents. For details see "Extending a Queue or Data File During Warm Start" on page 38.
- "Re-Allocate Queue File During Warmstart" — introduced with Version 6.7. For details see "Re-Allocate Queue File During Warm Start" on page 41.
- "Change Local Node Name During Warmstart" — introduced with Version 6.7. For details see *VSE/POWER Networking*.

At the same time, switching the spool file processing mode from shared to non-shared, or vice versa, during a VSE/POWER warmstart has been readdressed by improved operator guidance via messages 1QB3A/1QB3D and 1QBGD. Even a defensive coding technique of anticipating an incorrect operator response has been provided to terminate "shared spooling" operation as soon as concurrent "non-shared" operation has been detected that would eventually corrupt the spool files.

These steps may be invoked individually at separate warmstarts or combined during one warmstart incident, with the exception of "Release Migration" which cannot be combined with "Re-Allocation" of the queue file.

General Functional Enhancements

Automatic Journal Creation for Offload BACKUP/PICKUP/SAVE

When requesting the POFFLOAD BACKUP/PICKUP/SAVE command to write VSE/POWER entries to tape, the POFFLOAD Journal function automatically creates a journal list queue entry named $OFJnnnn, which records:

- all entries saved successfully to tape and individually numbered
- the offload command format with start and end time stamps
- tape attributes and volumes required

1. One last situation may occur where a cold start formatting of the data file can occur, but is generally less important and not requiring immediate attention, namely when the system administrator wishes to alter the DBLK or DBLKGP size to modify spooling utilization.

- most important console messages accompanying the offload process, relieving the operator's effort to search the hardcopy in case problems are noticed later

Thus the journal reflects the tape contents and increases the operator's productivity when later producing individual tape reports using the PDISPLAY command:

- the report already exists in the LST queue for all tapes used in the process
- the re-mounting of individual tapes for each report is no longer required

For details about the Journal function and for samples of the journal layout, refer to "POFFLOAD Journaling" on page 339.

Delayed Deletion of Queue Entries

When VSE/POWER queue entries are to be deleted, they are no longer processed immediately by the given task, for example a command processor task or an execution reader. Instead these tasks merely remove the queue entries from the RDR/LST/PUN/XMT queue and flag them "to be deleted", meaning that they are put in the DELetion queue. Then the termination task is posted to finally free the entries, that is, delete them from the DEL queue individually, return the queue record to the free queue record chain, write its DBLK groups free on the data file (a function of DBLK group tracing) and add these groups to a free DBLK group subchain.

When a sizable number of queue entries (for example, all of a given class) or a big queue entry has to be deleted, the 2–stage deletion process has great performance advantages:

- The deleting task is immediately able to continue — callers need not wait unexpectedly long
- "final deletion" is done with the lowest priority by the termination task which need not lock spooling resources during deletion of chained queue entries or during writing of DBLK groups to the free DBLK group subchain.

Consequently, when deleting a large spool entry, the default DBLK group tracing does no longer reduce VSE/POWER responsiveness. Hence DBLK group tracing will no longer be an option controlled by the PSTOP DBLKTR command and will always be available for the sake of spool data integrity. See also "Life Cycle of VSE/POWER Queue Entries" on page 30

Additional Improvements Implemented by Field APARs

The following changes have been implemented by field APARs:

- Suppress frequently occurring informational VSE/POWER messages on console

  When the console is flooded by unimportant messages, you can use the PVARY MSG command to disable them for display on the console, but have them available on the hardcopy console file for review purposes. See "Format 4: Varying VSE/POWER Messages" on page 391 for disabling, displaying and re-enabling of such messages.
- Avoid immediate processing of dispatchable entries reloaded from tape

  Using POFFLOAD LOAD/SELECT you may introduce dispatchable queue entries which are executed/processed unwillingly unless you // PAUSE the partition, disable dynamic classes, or stop processing tasks. Specifying the new HOLD=YES operand you can introduce dispatchable entries from tape with their disposition set to the corresponding "hold" value (disposition 'D' is set to 'H', and 'K' is set to 'L'). See "Format 3: Loading All or Selected Entries from Tape" on page 326.
- Recording of printed/punched output entries on the console

For PNET any transmitted queue entry is recorded on the console by the message 1RA0I, but for local LST/PUN tasks (especially with the' VM' option) there exists no automatic recording message for individually processed queue entries. With the LOG=YES operand, the "successfully processed" message 1Q8KI can be requested and, using the PVARY MSG facility, this message may even be disabled for the console, but appears in the hardcopy file. See "Format 1: Processing Disk-Spooled Output" on page 359.

- Establish operator confirmation for critical commands

  Inadvertently entering a PDELETE LST,ALL (for example using the short form L LST,ALL) command may have drastic unwanted effects. By default, a new SET statement SET CONFIRM=PDELETE,QUEUE,ALL has been established requesting an initial confirmation with the message

  ```
  1QZ3D  PROCESS 'L LST,ALL'? CONFIRM WITH 'YES', ELSE 'NO'
  ```

  Similarly, to protect against unwanted bulk execution, you may now provide the SET CONFIRM=PRELEASE,QUEUE,ALL autostart statement within your VSE/POWER startup procedure. See "SET: Setting VSE/POWER Startup Control Values" on page 473.

- Support for IBM 3592 WORM (Write Once Read Multiple) tape unit

  All VSE/POWER tape writing functions (POFFLOAD BACKUP/SAVE/PICKUP and DISP=T spooling) support the new archiving IBM 3592 tape unit, which cannot "backout" incomplete user data already written at tape end, forcing VSE/POWER to create "invalid" trailing queue entries on tape instead. The tape reading functions (POFFLOAD LOAD/SELECT and tape PDISPLAY) can identify "invalid" entries and ignore them. The tape writer task reading from a D/3592 is rejected by message 1Q7FI because of the possible presence of "invalid" data whose eventual presence could only be detected after the data had been processed. See "Layout of VSE/POWER Tapes" on page 527 for a description of ″invalid" queue entries on spool tapes.

- Execution pausing for dynamic partitions

  Following abnormal termination with SET NORUN=YES:
  – active jobs are held with DISP=X, and
  – static partitions enter the // PAUSE mode

  but dynamic partitions simply start to execute the next job. With the SET NORUN=YES,DYN1=PAUSE autostart statement, VSE/POWER handles static partitions and dynamic classes with **MAX ACTIVE=1** equally, that is, it modifies the dynamic partition behavior of such class(es) to cause the partition to enter the // PAUSE mode and allows for rearrangement of jobs to be executed in such class(es). See "SET: Setting VSE/POWER Startup Control Values" on page 473 for defaults on the improved NORUN= operand.

- Adapt single sheet separator pages for Spool Access tasks

  Customizing separator pages by the SET ISEP= or the SET DLSEP= and SET xxLINE= autostart statements for adapted single sheet printing until now has been effective only for local printing, not as, for example, AUTOLPR printing via Spool Access Support (SAS) tasks or Device Service Tasks (DST). This gap is now closed by the SET DLSEPSAS=YES|FORCE and the SET ISEPSAS=YES|FORCE autostart statements. For details see "SET: Setting VSE/POWER Startup Control Values" on page 473.

- Protection against unexpected number of Spool Access Support (SAS) task

  Failing XPCC applications (that CONNECT but don't DISCONNECT) may loop on connecting to VSE/POWER by ever increasing numbers of SAS tasks until all VSE/POWER partition Getvis or SETPFX storage is exhausted and spooling comes to a halt. Now a default threshold number of 250 current SAS tasks has

been introduced. When exceeded, the XPCC application fails during CONNECT with XPCC RETC/REAS=X'19/40' accompanied by VSE/POWER's RETCD/FBKCD=X'10/07', and the operator is informed by message 1Q3JA. She may then check the failing XPCC application and/or modify the threshold via the PVARY MAXSAS,nnn command. See "Format 5: Varying the Maximum Number of SAS Tasks" on page 393 for further information.

- Identify all possible segment attributes at IPWSEGM request

  Upon invocation of the IPWSEGM macro to establish the spooling attributes of a new segment and thereby overwriting the attributes of the previous segment, it may be necessary to re-establish the attributes of the old segment. Until now the program needed to remember the attributes of the old segment. With the previous IPW$MXD macro version "VS10" (see field $MXVRS) there was no complete support for this. When recompiling with VSE/POWER 7.1.0 the new IPW$MXD macro (version "VS20") an extension area provides 14 additional segment attributes which give information about previous segment characteristics. See *VSE/POWER Application Programming* "IPW$MXD Mapping Macro" for extended segment attributes.

  Note: The user must reassemble his application code with the new PNODE macro to benefit from these performance improvements.

- Allow PNET exploitation of "external-IP-address" with firewall

  For TCP/IP networking, when the local node resides behind a firewall (known externally by an "external-IP-address" but internally by an "internal-IP-address", the local node will return its "internal-IP-address" in the OPEN CONTROL record to remote node, which fail with the message:

  ```
  1RT3I ... UNKNOWN ... IP-ADDR  (=internal-IP-address of local node)
  ```

  Now the PNODE macro provides a new operand IPEXTRAD=external-IP-address (of local node) for PNODE entries of remote nodes to ensure successful session establishment. For details refer to:

  – PNODE macro. See "Format 4: Defining a Directly Linked TCP Node" on page 84
  – display of external address. See "Format 9: PDISPLAY PNET" on page 275.
  – VSE/POWER Networking. See *VSE/POWER Networking* "Change Local Node Name During Warm Start".

- Accelerate PNET TCP and SSL connections

  The previous PNODE macro established the standard number of two receiver buffers for communication with a remote TCP or SSL node. As soon as the TCP/SSL layer provided data to PNET receiver tasks (which have to spool the data to disk), the two receiver buffers may not have been sufficient and led to a "wait-a-bit" indicator to be returned to the transmitting node, slowing performance. This is now avoided by the extended PNODE macro, which defines a default MAXBUF=(4,4) operand, namely 4 receiver and 4 transmitter buffers, and which allows to specify a maximum MAXBUF=(16,16) configuration.

  Note: The user must reassemble his NDT with the new PNODE macro to benefit from these performance improvements.

Summary of Changes, VSE/POWER 6.7

The following summary of changes for the previous VSE/POWER 6.7 is included to help those readers that did not install that release to keep informed of functional changes.

Extension of Functional Support

This section summarizes the changes made to provide enhanced functionality.

Relief for High End Spooling Systems

Number of Spool Entries Increased to 100,000: The number of spooling entries has been extended from 32,768 (32,766 usable) to 100,000 (99,998 usable). The externally visible highest wrap-around job number of 65,535 has not been changed[2]. The structure of the increased queue file IJQFILE on disk remains unchanged, however the following internal changes should be considered with respect to user exit or OEM code:

- the disk management block (DMB), as mapped by the macro IPW$DQC, has had a major rearrangement and code references will require a recompile
- the 2–byte pairs of class anchors of the DMB for the queue classes (RDR/LST/PUN/XMT), as mapped by the macro IPW$DCT, have been extended to 4–bytes pairs, possibly requiring code changes by non-standard interfaces
- the internal macro IPW$WFQ (wait for class posting) no longer checks the standard ECB bit 16 (0,1,2...) for class anchor activity, and instead checks bit 32, possibly requiring code changes by non-standard interfaces
- the spool environment header record (SEH), as mapped by the macro IPW$DSP SEH=YES, and spool environment trailer record (SER), as mapped by the macro IPW$DSP SER=YES, have been extended in size from X'70' to X'C0' bytes as indicated by their internal 4–byte length field. These records precede the first, respectively last, data block (DBLK) of every data block group (DBLKGP) of the VSE/POWER data file (see "Analyzing Queue File and Data File in Dumps" on page 523 for the record layouts). Code of non-standard interfaces referring to these records may require verification

For details on the external specifications of a standard size or large queue file, refer to "The Size of the Queue File" on page 33.

New Internal Validity Checking "Data Ownership Test": With this release VSE/POWER has activated the "data ownership test", whereby all queue entries being fetched from a RDR/LST/PUN/XMT queue are verified whether spool data contained in the individual data block groups (DBLKGP) are correctly chained to the owning spool entry queue record. In an incorrect case, the entry is flagged by the message 1Q6UA, isolated by the message 1Q6JI, and finally the processing task is terminated. The checking is very fast with no visible effect on system performance, and can help greatly to reveal the presence of data corruption, especially in large spooling systems. See "Data Ownership Tracing" on page 527.

2. Although the highest job number remains unchanged and wraps around to the job number 00001, as in previous VSE/POWER releases, commands using both the job name and job number to identify a spool entry to be processed, will for all practicable purposes be adequate. Otherwise a spool entry can be uniquely identified by the internal queue entry number QNUM (see "Format 1-4: PDISPLAY LST,FULL=YES" on page 257

Display List of Largest Spool Entries

In dealing with the problem of insufficient spooling capacity, VSE/POWER supports new commands and operands to identify those spool entities consuming the greatest spooling space, for example:

- PDISPLAY LST,CPAGES>nnnnn or PDISPLAY PUN,CCARDS>nnnnn search the RDR/LST/PUN/XMT queues for the spool entries larger than the specified page or card count. These "C"-type search operands may also be specified for the queue manipulation commands (PALTER, PDELETE, PHOLD and PRELEASE). For a complete list of "C"-type operands, see the section "Keyword Search Operands" of the specified command.
- PDISPLAY BIGGEST,LIMIT=nn searches the complete VSE/POWER spool space queues CRE/RDR/LST/PUN/XMT/DEL for the 'nn' largest queue entries and presents them in the descending order of DBLK groups used. For details on the command format, see "Format 6: Displaying Information for the 'nn' Largest Entries" on page 237 and details on the display format, see "Format 6: PDISPLAY BIGGEST,LIMIT=7" on page 267.

Reducing System-Down Time

Although the VSE/POWER 6.6 function "Data File Extension During Warm Start" helped customers to avoid a perhaps lengthy system cold start, there remained two further important cases where a spooling system needed modification(s) requiring formatting (queue file and all data file extents) and a perhaps lengthy down time, namely when 1) the queue file has to be enlarged, or 2) when the local PNET node name must be changed. The consequence is that the operator must offload/backup the spool file, format the queue/data files via cold start, and reload the offloaded spool entries, perhaps costing several hours. Now both of the above cases can be performed by a much faster warm start.

Re-allocate Queue File During Warm Start: To further minimize system down-time, VSE/POWER can now re-allocate the queue file, that is, relocate it to **another** disk position, and further **extend** the file size at the same time. This transition during warm start requires only a short formatting step for the new queue file since already existing queue records of the old queue file are transported to the new queue file, thus leaving the linkage between a spool entry queue record and its spool data untouched. The queue file re-allocation is triggered when the STDLABEL.PROC and DTRPOWR.PROC have been changed as follows for the subsequent VSE/POWER warm start:

- the existing queue file is named IJQFOLD and assigned to the logical unit SYS034
- the newly extended queue file is named IJQFILE and assigned to the logical unit SYS001

Following operator prompting and extent verification the extended IJQFILE is formatted and is primed with all queue records of IJQFOLD. Finally the VTOC entry of IJQFOLD is overwritten by a work file named "VSE.POWER.DUMMY.FILE". For details see "Extending Existing VSE/POWER Spool Files" on page 38.

PNET Node Name Change during Warm Start: The local PNET node name is not only contained in the spool entry queue record, but also in spool entry control records (job header and data set header records) within the spool data on the data file. Converting the spool entry node name previously could only be achieved by performing a cold start as described in *VSE/POWER Networking* under "Changing the Name of the Local Node".

This conversion process can now be requested during a VSE/POWER warm start, whenever a different local node name is introduced via the network definition table (NDT) as specified by the PNET= operand of the POWER macro generation, or by the SET PNET autostart override statement. After operator confirmation, the conversion process is started for all existing queue entries. "Work-in-progress" messages will accompany the processing as the VSE/POWER initialization is possibly delayed due to this new function, until finally the dynamic conversion of the spool files has completed. For details see *VSE/POWER Networking*.

General Functional Extensions

SLI Member JECL Statement Continuation Enablement

Source Statement Library Inclusion (SLI) for a member originating from a VSE/AF or VSE/ICCF library had previously never allowed continuation of a * $$ LST or * $$ PUN from one statement to another according to VSE/POWER JECL continuation rules. Instead the * $$ LST or * $$ PUN JECL statement was rejected by message 1Q49I INVALID DELIMITER followed by 1R33D CORRECT FULL STATEMENT, due to the reason that VSE/POWER disallowed continuation and required that the last operand value not end with a delimiting comma.

Now VSE/POWER provides support for processing of an "unlimited" number of JECL operands by enabling the continuation of the * $$ LST and * $$ PUN statements which appear within an SLI member. For details see "* $$ SLI: Including a Library Member into a Job Stream" on page 463, especially "Rules for Coding the SLI Statement" on page 463.

Display Spool Entry Number and Distribution

The PDISPLAY Q command has been enhanced to display the current status of spool entries with respect to number and distribution of used entries:

- the total number of current used spool entries (used queue records)
- the distribution of the spool entries among the logical in-creation (CRE) and deletion (DEL) queues
- the distribution of the spool entries among the physical queues (RDR/LST/PUN/XMT)

For an example see"Format 7: PDISPLAY Q" on page 268.

Additional Improvements Implemented by Field APARs

The following changes have been implemented by field APARs:

- Eliminate spooling bottle neck during console decision message 1Q55D reply period.

  When a VSE/POWER controlled partition was performing tape spooling or VSE/POWER was printing from tape, and the task did not provide a correct tape address (cuu), VSE/POWER previously requested the operator to specify a tape device via the decision message 1Q55D, and the VSE/POWER partition console support went into a wait state until the reply was entered, and spooling activity began to grind to a halt as tasks attempting to issue a console message had to wait. Now the message 1Q55A action message is issued prompting the operator to issue a PGO command response specifying the "cuu" device. This solution does not block the VSE/POWER console nor spooling activity. See "Format 5: Reactivating a Tape Spooling Task in 1Q55A Wait State" on page 302 and "Format 6: Reactivating a Tape WriterTask in 1Q55A Wait State" on page 302.
- Suppress segmentation of PDUMPS's when spooling to 3800 printer.

Segmentation caused by a 3800 printer SETPRT requests of the PDUMP macro may now be suppressed using the

```
SET PDUMP=NOSEGMENT
```

autostart statement. See "SET: Setting VSE/POWER Startup Control Values" on page 473

- Redisplay ignored startup statements.

  Previously misplaced SET or DEFINE autostart statements were ignored during VSE/POWER startup as indicated by message 1Q06I. Now the command:

  ```
  PDISPLAY AUSTMT
  ```

  has been enhanced to display ignored autostart statements, which may have been overseen by the operator. Ignored statements are prefixed with the text:

  ```
  >> IGN'D:
  ```

  and erroneous statements are prefixed with the text:

  ```
  >> ERROR:
  ```

  See "Format 14: Displaying Information About Used Autostart Statements" on page 245
- Remove command length limitations of 72 bytes.

  VSE/POWER commands submitted via the Attention Routine interface or via the Spool Access Support (SAS) CTL request previously were limited to 72 bytes. Now the maximum command length has been increased to 130 bytes, allowing for more operands or longer operand values.
- Preserve $LSTnnnn or $TAPnnnn entries in the LST queue.

  The $LSTnnnn LST queue entry created by a

  ```
  PDISPLAY ALL|TOTAL,LST
  ```

  command, or similarly the

  ```
  PDISPLAY entry,TAPE=cuu,OUT=LST
  ```

  command has a default disposition 'D' and class 'A'. It may easily be selected for printing by a local list task before the operator can browse and view this entry. Now the

  ```
  SET HOLDCL=class
  ```

  autostart statement allows to pre-determine the default class of these entries, and at the same time keep them with disposition 'H' in the LST queue for browsing. See "SET: Setting VSE/POWER Startup Control Values" on page 473
- Allow the specification of the BLDG=, DEPT=, ROOM= and PROGR= operands in the * $$ LST or * $$ PUN statements and thereby override any * $$ JOB specification.

Summary of Changes, VSE/POWER 6.6

The following summary of changes for the previous VSE/POWER 6.6 is included to help those readers that did not install that release to keep informed of functional changes.

Extension of Functional Support

This section summarizes the changes made to provide enhanced functionality.

Data File Extension During Warm Start

To minimize the long system downtime (caused by formatting all extents of the data and queue files during a VSE/POWER cold start), VSE/POWER now offers extension of the data file during a warm start. The data file extension will not affect already spooled data and is triggered when VSE/POWER detects that one more extent has been appended to the existing IJDFILE DLBL/EXTENT statements in the label procedure (STDLABEL.PROC) of a VSE/POWER warm start. After operator prompting and file verification, the formatting of the additional extent is done in parallel with ongoing spooling. For details see "Extending Existing VSE/POWER Spool Files" on page 38.

Along with this new function, the PDISPLAY Q command (see sample under "Format 7: PDISPLAY Q" on page 268) has been enhanced to present all currently used extents of the queue, data, and account files and their detailed physical locations.

Extension of Networking Capabilities

VSE/POWER networking has been enhanced to use the Secure Sockets Layer (SSL) feature for TCP/IP connections. Data (jobs, list and punch output, messages and commands) which is sent between two VSE/POWER nodes via a TCP/IP connection is encrypted according the private key of the customer. Network control records, which contain for example a node password, are encrypted as well. This support is referred to as 'PNET SSL'. The usage of the SSL feature is optional and is selected for each node separately, which means that there are henceforth

- SSL nodes using a TCP/IP connection with the SSL feature
- TCP nodes using a TCP/IP connection without the SSL feature.

For this purpose, VSE/POWER initiates an application interface to the local TCP/IP host partition and communicates via Assembler asynchronous EZASMI SOCKET requests that enter the LE environment for C-SOCKET calls to the 'TCP/IP for VSE/ESA' host partition. This application interface is established in addition to an application interface used for TCP nodes.

The following enhancements, changes, and rules accompany this networking enhancement:

- The PNODE macro offers new operands to define an SSL node by specifying the dotted decimal IP address or the symbolic IP address of the remote node, and optionally a port number. In addition, new operands are available to specify the private key of the customer.
- A Network Definition Table (NDT) is generated from PNODE assemblies. PLOADing an NDT with at least one SSL node will immediately open VSE/POWER's TCP/IP interface, so that any incoming CONNECT request can be responded to, although no PSTART command was given for the connecting node.
- The new PNODE macro generates a Version 06.0 NDT, which is incompatible with NDTs (Version 05.0) generated by previous VSE/POWER releases.

  Note: Existing NDTs must be re-assembled under VSE/POWER 6.6.
- The display of a Version 06.0 NDT has a new arrangement.
- The PSTART PNET command for an SSL node has the same format as used for TCP nodes. All other PSTOP, PDRAIN, PACT, PFLUSH, and PINQUIRE networking commands are independent of the node type and therefore transparent.
- The PINQUIRE command has been enhanced for

  - subselection via PNETSSL
  - new display of local node attributes
  - improved display of remote nodes
- The so-called PNET Console trace of BSC/CTC/TCP/SSL communications has been extended for
  - subselection of PNETSSL
  - comprehensive trace information for SSL nodes

Additional Improvements Implemented by Field APARs

The following changes have been implemented by field APARs:

- Request confirmation for critical VSE/POWER commands

  Use the new SET CONFIRM=power-command to request confirmation of the command. For example, SET CONFIRM=PEND will then prompt the operator with the new message 1QZ3D. See "SET: Setting VSE/POWER Startup Control Values" on page 473.
- Request identical separator pages

  Use the new SET DLSEP=YES|FORCE command to force identical separator pages and drop the 'last' separator page for *all* list tasks to be started.

  This is similar to the existing 'DLSEP' operand of the PSTART LST command. See "SET: Setting VSE/POWER Startup Control Values" on page 473. For the differences between 'YES' and 'FORCE', see Table 7 on page 185.
- Ignore SET NORUN=YES for selected jobs

  For CICS and VTAM startup jobs, which should come up in any case, it may not be desirable to enter DISP=X at VSE/POWER recovery due to SET NORUN=YES. The new * $$ JOB operand NORUN=IGN allows ignoring such general SET NORUN processing. This special job attribute is reflected
  - during PDISPLAY RDR,...,FULL=YES by NORUN=IGN
  - via Spool Access GET-OPEN reply SPL by SPLDFLG.SPLDFRUN X'08'
  - via Spool Access CTL (Fixed Format) Display Request by PXFMFLG2.PXFM2RUN X'01'
- Search for U='user_information'

  The user information as specified by the UINF= (or USER=) operand of a * $$ JOB/LST/PUN statement or of the PALTER command is displayed (with FULL=YES) via the U=' 16 bytes ' field. It is used by TCP/IP for VSE to contain the dotted decimal or symbolic IP address of the IP host owning the target printer or the name of a script file. Use the new CUINF='...' search operand of the queue manipulation or display commands to select queue entries according to Current User Information content.
- Verify successful command processing

  Existing message 1R88I OK, issued for completed central operator queue manipulation commands (PALTER, PDELETE, PHOLD, PRELEASE), has been extended to identify the number of processed queue entries together with the repeated command, e.g.

```
1R88I  OK : 6 ENTRIES PROCESSED BY PALTER,LST,*SSL,DISP=K
```

Chapter 1. Understanding Syntax Diagrams

This section describes how to read the syntax diagrams in this manual.

To read a syntax diagram follow the path of the line. Read from left to right and top to bottom.

- The ►►— symbol indicates the beginning of a syntax diagram.
- The —► symbol, at the end of a line, indicates that the syntax diagram continues on the next line.
- The ►— symbol, at the beginning of a line, indicates that a syntax diagram continues from the previous line.
- The —►◄ symbol indicates the end of a syntax diagram.

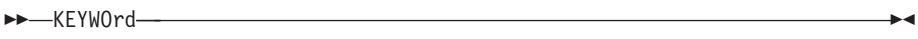
Syntax items (for example, a keyword or variable) may be:

- Directly on the line (required)
- Above the line (default)
- Below the line (optional)

Uppercase Letters

Uppercase letters denote the shortest possible abbreviation. If an item appears entirely in uppercase letters, it can not be abbreviated.

You can type the item in uppercase letters, lowercase letters, or any combination. For example:



In this example, you can enter KEYWO, KEYWOR, or KEYWORD in any combination of uppercase and lowercase letters.

Symbols

You **must** code these symbols exactly as they appear in the syntax diagram

*	Asterisk
:	Colon
,	Comma
=	Equal Sign
-	Hyphen
//	Double slash
()	Parenthesis
.	Period
+	Add
$	Dollar sign

For example:

```
 * $$ LST
```

Variables

An *italicized* lower-case word indicates a variable that you must substitute with specific information. For example:

Here you must code `,USER=` as shown and supply an ID for user_id. You may, of course, enter `USER` in lowercase, but you must not change it otherwise.

Repetition

An arrow returning to the left means that the item can be repeated.

A character within the arrow means you must separate repeated items with that character.

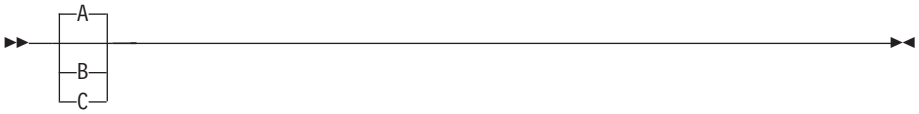
A footnote (1) by the arrow references a limit that tells how many times the item can be repeated.

Notes:

1 Specify *repeat* up to 5 times.

Defaults

Defaults are above the line. The system uses the default unless you override it. You can override the default by coding an option from the stack below the line. For example:



In this example, A is the default. You can override `A` by choosing `B` or `C`.

Required Choices

When two or more items are in a stack and one of them is on the line, you **must** specify one item. For example:

Here you must enter either `A` or `B` or `C`.

Optional Choice

When an item is below the line, the item is optional. Only one item **may** be chosen. For example:

Here you may enter either `A` or `B` or `C`, or you may omit the field.

Required Blank Space

A required blank space is indicated as such in the notation. For example:

```
* $$ EOJ
```

This indicates that at least one blank is required before and after the characters $$.

Chapter 2. Planning for VSE/POWER

What VSE/POWER Does

Queue Control

VSE/POWER is the spooling system of z/VSE. As such it performs the following functions:

- It reads jobs from various input devices, including an RJE workstation, and stores them in the input queue (RDR queue).
- It starts jobs from the input queue in one of the partitions which it controls.
- It stores output from various jobs in one of the output queues (LIST, PUNCH, or XMIT) or on tape and, if required, controls the writing of it on a printer.
- On request, it transfers spooled output to a subsystem in another partition. The subsystem then can print, display, or punch this output.
- It maintains a transmit queue for jobs or output to be transmitted to another node.

Classes and Priorities

Classes and priorities control the processing of jobs within VSE/POWER. VSE/POWER processes jobs according to their class assignments and within the classes according to their priority specification.

For every job, VSE/POWER supports different classes and within that class different priorities. VSE/POWER supports:

- 36 job classes (A-Z, 0-9)
- 36 output classes (A-Z, 0-9)
- 10 levels of priority per class (0-9).

The *system programmer* uses classes at system start-up to define which jobs are to execute in a certain partition or on which printer(s) the job output is to be processed.

The *operator* uses the CLASS operand of a VSE/POWER command or a JECL statement to, for example, change existing or standard class assignments. By changing classes and priorities, the operator can schedule and control the processing of jobs and output on the system.

For detailed information on classes, see “VSE/POWER Classes” on page 53.

Submitting Jobs and Routing Output

Formerly, jobs were often entered as card packs and read in from a card reader. Now, they are usually written at a terminal and then submitted to the input queue by the ICCF command SUBMIT or via the Interactive User Interface panels of z/VSE.

With the SYSIN tape reader function, POWER also can read input from tape.

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 z/VSE job control language (JCL) may be used. For examples of the use of VSE/POWER JECL, please see Figure 65 on page 400.

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.

With options of the LST and PUN JECL statements, you can route printer or punch output to tape rather than to an output queue on disk and later print or punch this output directly from tape.

With the POFFLOAD command, you can save spooled data to tape and load it again into existing queues.

Remote job entry under VSE/POWER and the types of supported RJE terminals are described in detail in *VSE/POWER Remote Job Entry*. For details of exchanging jobs and output data in a network, see *VSE/POWER Networking*.

Disposition of Jobs and of Output

When you submit a job, you can instruct VSE/POWER what to do with it. You can, for example, spool it and hold it in the queue file unprocessed until your operator releases it. Job input as well as job output may be held in its queue for processing 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 the job or output for processing later when the system is less occupied.

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 unrecoverable I/O error occurs.

See Appendix A, “VSE/POWER Disposition Codes,” on page 497 for a list of disposition codes that you can specify and which VSE/POWER assigns by default or for unexpected events.

Time Event Scheduling of Jobs

VSE/POWER supports the scheduling of jobs for processing in a partition at a date, which can be defined.

1. If the job has to be scheduled once, you specify the time and the date.
2. If the job has to be scheduled repetitively, you can specify that the job is scheduled
   - daily
   - every weekday (for example, every Monday)
   - every specific day of every month (for example, every first)

- every specific day of a certain month (for example, every first of January, June and December).

3. If the job specifies scheduling every day or on days of a weekday list, the job may even be scheduled to run more than once per day.

Segmentation of Output

Turnaround time for jobs with much printed or punched output can be improved by segmenting the output; consequently, certain parts of the output are 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).
- By command request (command-driven segmentation)
- By Spool-Access PUT-OUTPUT segmentation request

For details on how to segment output, please see "Output Segmentation" on page 117.

Count-driven segmentation is always done on logical boundaries. 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.

Job Accounting

The ACCOUNT operand of the POWER generation macro allows you to specify whether or not you want to collect accounting information for every job running under its control. It stores the information in an account file on disk. By means of a VSE/POWER command, you can request that this information be copied into a separate disk file or to tape. You also can request that the account information be spooled as punched output and queued into the punch queue.

VSE/POWER also stores Advanced Function Printing account records into the account file for device driving systems that process VSE/POWER output entries according to their own printing rules. For information on this account record, see section "Put Account Record Order" in the manual *VSE/POWER Application Programming*.

Tape Support

VSE/POWER has 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 later by a VSE/POWER list or punch task, whichever applies, or reloaded on disk.

Supporting Various Console Commands

Commands which enter the system via the VSE Attention Routine are processed by VSE/POWER depending on the console role. The commands issued on a *system* or

master console are unrestricted. Note that in this manual, the terms system and master console will be referred to as system console only.

Commands issued on a *user console* will be limited in the scope of the commands that will be allowed and in the set of VSE/POWER queue entries that can be manipulated and/or displayed. The scope of limitation is identical to that of a cross-partition user. For values see the PWRSPL FUNC=COMMAND table in *VSE/POWER Application Programming*. The commands that manipulate or display queue entries, however, will be allowed only for the entries for which the user's console name matches that of the origin or target user ID of the job entry. Any entry password for the manipulation will not be required. To enjoy the full benefit of VM/CMS user console support, do the following:

- Activate VSE security. This is required; see the note below.
- Use the * $$ JOB FROM= operand. This is recommended.
- Use the * $$ JOB ECHO= or ECHOU= operand. This is optional.
- Use the SUBVSE and VSECMD commands. The VSECMD command is required for issuing commands. For details see *z/VSE Operation*.

Note: VM/CMS console users will have user console command authority only when z/VSE Access Control (security) is activated (IPLed with SEC=YES). If VSE security is not activated, then CMS users will have system console authority.

For a detailed description of the console function and the various consoles refer to *z/VSE Operation*.

An *operator* is anyone issuing a command to VSE/POWER. Those commands issued on a system console will be designated as **central operator commands**. Central operator commands may also be entered by a system administrator, for example a 'type1' user. For details, refer to *z/VSE Operation*.

Networking Function (PNET)

This function permits the connection of a VSE system that includes Version 7 of VSE/POWER to nodes within a network with any of the following:

1. VSE/POWER, Version 2 and following
2. SSX/VSE
3. OS/390 MVS with JES2
4. OS/390 MVS with JES3 (and BDT)
5. VM with RSCS
6. AS/400

supporting the NJE protocols.

Transmission can be via a synchronous half-duplex multileaving BSC line, an SDLC session, a virtual CTCA under VM, or via TCP/IP or TCP/IP SSL (TCP/IP Secure Sockets Layer).

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

Jobs
Output
Messages
Commands

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

This manual presents some PNET-related reference information. The manual *VSE/POWER Networking* shows how you can use networking functions.

Remote Job Entry (RJE) Function

This function allows jobs to be entered from remote terminals and output to be obtained either centrally or at a terminal.

This manual includes RJE-related reference information; the manual *VSE/POWER Remote Job Entry* shows how you can use the remote job entry function.

Shared Spooling

This function allows up to nine VSE systems using VSE/POWER of the same version and release to share a single VSE/POWER spool file (which consists of the queue and data file). The VSE/POWER account file also may be shared with the restriction that all systems sharing the VSE/POWER account file also must share the VSE/POWER spool file. This sharing of files applies to VSE systems running on different processors or on the same processor under different VM systems if the z/VSE Access Control function is not activated. 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. If VSE security is active, the job activation may be controlled by the security zone of the job origin. For details, see "Data Security and Shared Spooling" on page 13.

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

- *Job-Load Balancing* The queuing priority of jobs in VSE/POWER is valid for all sharing systems. If a high-priority job is entered into the spool files, it runs 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 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.

A summary of how shared spooling works follows:

Controlling Spool Files from the Operator's Console

Operators of sharing systems can control all jobs in the shared VSE/POWER queues. They cannot, however, control partitions or devices that do not belong to their 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.

JECL statements allow you to define which sharing system is to process a job or output.

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.

Hint: To prevent one sharing system from retaining write access indefinitely, specify a time limit in the TIME operand of the POWER generation macro. For available TIME values, refer to page 71.

While a 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 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.
- When spooling job 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 systems to wait if they need to update the control information in the account file. VSE/POWER of each 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.

For more details on shared spooling, refer also to:

- "Shared Spooling Requirements" on page 49.
- "Shared Spooling: Startup Considerations" on page 98 and following pages.
- "Shared Spooling Considerations" on page 188.

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 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.
- Retrieve or delete VSE/POWER job event messages.

Normally, IBM-supplied components make use of this kind of support. For details of the "spool-access support" refer to *VSE/POWER Application Programming*.

External Device Support

This support is an 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, as offered by the IBM Print Service Facility (PSF) or the Report Controller feature of CICS. 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*.

Data Security Considerations

VSE/POWER complements user data protection as offered by VSE by giving the user extra data security and by protecting some of the user's system functions. In particular, VSE/POWER has security protection in the following functional areas:

- **Job execution and resource access protection** Together with VSE security, the user can specify a job's security userid and password for accessing protected resources, using either the SEC operand of the * $$ JOB statement or corresponding fields in the PWRSPL macro when submission is via the spool-access support. Jobs which specify invalid security values are flushed (disposition 'D' jobs are deleted, and disposition 'K' jobs are changed to disposition 'L') and message 1Q4EI appears.

  Refer to *z/VSE Guide to System Functions*.

- **Output access protection**

  The spool-access support retrieval of output is granted only to the originating or target user. In addition, the output may be protected by an optional VSE/POWER password from access by other users through the PWD operand in either the * $$ JOB or * $$ LST/PUN statements, or the corresponding field of the PWRSPL macro.

  If the output is spool access protected, access is further reduced to those users who are authorized (i.e., have performed a security logon). See "VSE/POWER Spool Access Protection" on page 14.

- **Job access protection**

  A job in the RDR queue is protected the same way as output (see above).

- **Queue entry manipulation**

  Jobs and output are protected from manipulation (for example, altering or changing job disposition) in the same way as output (see "Output access protection" above). For a *user console*, however, additional password checking will not be performed; see "Supporting Various Console Commands" on page 7.

- **Logon of Remote Job Entry (RJE) stations**

  Protection from unauthorized logon by an external remote job entry station is offered through use of a signon password. For details, refer to *VSE/POWER Remote Job Entry*, "Security".

- **Logon of PNET nodes**

  Protection from unauthorized logon by an external spooling system is offered through use of a signon password. Refer to *VSE/POWER Networking*, "Providing Data Security".

- **Non-central operator command authorization**

  VSE/POWER spooling command users are divided into different authorization groups. RJE and NJE users should refer to *VSE/POWER Networking*, "Command Authorization". Spool-access support users should refer to section "Scope of GET/CTL Access to Queue Entries" in the *VSE/POWER Application Programming* manual.

Except for security protection offered by the VSE function, the central operator and any system administrators have unlimited rights. System administrators are recognized by their password when accessing via the spool-access support interface. For details, refer to *VSE/POWER Application Programming* manual.

Authenticated Jobs

VSE/POWER jobs for which both the VSE security userid and password are not explicitly specified may inherit through propagation VSE security authorization from a ″parent″ job (for details see the section below). Such jobs are designated as ″authenticated″ with the security zone of the parent job (see the SECNODE operand of the POWER macro on page 68). The password is dropped so that the original password can change, but the child job retains its authorization.

Such jobs will be recognized as authenticated even when they are passed via:

- A network to another VSE/POWER system
- A shared DASD to another system
- A POFFLOAD tape to another system

as long as the executing system shares the same SECNODE value.

Priority of Job Security Specifications

VSE/Advanced Functions jobs which are executed under VSE/POWER will obtain security authorization values in the following order of importance:

1. The values of the VSE/Advanced Functions // ID statement, if specified.

   Note: If the // ID statement is rejected, the next order of priorities is used.
2. The values of the * $$ JOB SEC operand, if specified
3. The values of the PWRSPL SPLXSID field, if specified
4. The ″propagated VSE security values″, if any (see section below).

Job Execution with Propagated VSE Security Values

If a VSE/POWER job does not explicitly specify security values by the * $$ JOB SEC operand, the job can inherit VSE security values if the following is true:

1. Execution occurs on a secured VSE system (z/VSE Access Control is activated) and
2. The job is submitted internally by a parent job which itself is
   - An IBM 'authorized program' (for example the Interactive User Interface, CICS).
   - Or an 'authorized job', for example one that itself had been security authorized at execution time by VSE.

   and submission is via the:
   - VSE/POWER spool-access support
   - VSE/POWER spooled punch output (* $$ PUN DISP=I)

Jobs that are not submitted to VSE/POWER as stated above, do not acquire any propagated security authorization.

Note: A job can lose its propagated security values during execution if a * $$ EOJ statement is processed at that time. See "Handling Read-in JECL Statements at Execution Time" on page 403.

The values propagated by the parent job or program are basically the parent job security userid, a 'trusted' indicator, and the security zone. The exceptions are:

1. For a child job submitted via the spool-access support:
   - If the PWRSPL parameter list of the parent job explicitly contains the security userid and password, then the default security values are the specified userid and password.
   - If the parent job is an 'authorized program' and the PWRSPL parameter list contains a security userid without a password, then the default security values are the specified userid and the security zone of the parent job.
2. For a child job submitted via * $$ PUN DISP=I :
   - The security values are those in effect for the VSE job which produced the punch output at the time when the * $$ PUN DISP=I statement is processed.

With these propagated values the child job may run successfully in the same security zone as the parent job. If the zone is not the same, message 1Q4HI is issued and the job runs without security values. Likewise, if the child job is transferred to a system on which the userid is unknown (not in the DTSECTAB), then the message 1Q4JI is issued and the job runs without security values.

Data Security and Shared Spooling

If a system is running in a shared environment with the z/VSE Access Control function activated, VSE/POWER assigns to that system a security zone through the SECNODE operand of the POWER generation macro. Any jobs entering that system which use propagated VSE security authorization from a parent job (for example because an explicit security userid and password are not specified) executes according to the following rules:

- If a specific shared system is specified (by means of the SYSID operand), the job will be selected to execute on this very system.
- If no specific shared system is specified, the job will be selected to execute on a system in the same security zone (SECNODE operand) as that of the parent job.
- If no specific shared system is specified and no system has a matching SECNODE value, the job can be selected by any system for execution.

In all of the above cases, if the executing system does not have a matching SECNODE value, the job will lose its security authorization at execution time, and message 1Q4HI is issued.

Do not dispatch a job that has inherited security authorization when only one shared CPU has the indicated SECNODE value and that secnode can not be started. Instead, use the PRESET SECNODE command as described on page 351. Note that the job loses its security authorization if it does not execute on a CPU with the same SECNODE value.

Hint: PDISPLAY command with FULL=YES.

Data Security and Multi-Tasking (MT) Partition

In the case that a partition has been started as multi-tasking (MT) partition, then one job will be allowed in a single process step (// EXEC) to read further jobs from the VSE/POWER RDR queue. The passing of a complete RDR queue entry to the job will cause some VSE/POWER end-of-job processing for each entry read. This in turn will cause the process step security authorization to be cancelled when the process step completes and z/VSE Job Control becomes active again. If it is necessary to reestablish the job's security authorization, this may be done by following the // EXEC statement with a // ID statement.

Data Security with PNET (SSL)

To provide message integrity, authentication, and confidentiality, VSE/POWER uses the Secure Sockets Layer (SSL) feature of the TCP/IP discipline for communication between TCP/IP nodes. Support exists for either the SSL version 3.0 or Transport Layer Security (TLS) version 3.1. Furthermore, it is possible to specify the encryption type to be used (either "weak" or "strong") depending on the needs or geographic requirements.

- For information on SSL support see *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*, SC33-6601.
- For information on PNET SSL, see *VSE/POWER Networking*, SC33-8249.

PNET SSL support requires additional generation steps by the system administrator. Generation operands specific to user security can be found under "PNODE Generation Macro for Networking Support" on page 81:

- PNODE macro for the local node (LOCAL=YES), specification of:
  - SECTYPE= security protocol type (SSL 3.0 or TLS 3.1)
  - KEYRING= key database sublibrary name
  - DNAME= key database sublibrary member name to be used as a key for the local node
  - PWD= node password (default, overwrite by PSTART PNET) for transmission to other nodes
- PNODE macro for a directly linked SSL node, specification of:
  - ENCRYPT= encryption level to be used (WEAK or STRONG)
  - DNAME= key database sublibrary member name to be used as a key for this node
  - PWD= node password used to verify received connection request

Data Security and Tapes

Due to the tape functional support via VSE/SAM, spool entries and accounting data can be additionally protected with tape labels, using VSE/SAM and VSE/AF internal interfaces to Access Control.

VSE/POWER Spool Access Protection

Overview

This feature, which can optionally be activated, provides enhanced security to control the access and manipulation of spool file entries in the RDR, LST, PUN, and XMT queues. It eliminates spool file security exposures in VSE/POWER whereby a non-authenticated program (including a device driving system (DDS) program) using VSE/POWER Spool-Access Support or the spool macros GETSPOOL/CTLSPOOL could identify itself as an arbitrary user ID and access or manipulate another user's spool entries.

This access control can, depending on IPL, be activated at VSE/POWER startup and applies to eligible spool entries (generally speaking, those with an origin or target user ID) accessed internally via the Spool-Access Support (SAS) function or the GETSPOOL/CTLSPOOL macros. In this case, such entries can be accessed only by authenticated users or programs (those having performed a security logon), system administrators, or the central operator.

Individual spool entry protection can be deactivated by the JECL operand SECAC=NO in the * $$ JOB, * $$ LST, and * $$ PUN statements and by setting the PWRSPL flag SPLX1SNO when submitting spool output entries via the SAS interface.

For more information, see "Scope of GET/CTL Access to Queue Entries" and "Spool Access Protection Considerations" in *VSE/POWER Application Programming*.

Note: VSE/POWER Spool Access Protection is available with either BSM or ESM.

With the PDISPLAY FULL=YES command on a system where Spool Access Protection is active, the operator can display whether a spool entry has been specified to be ineligible for Spool Access Protection.

Spool Access Protection is ignored if a user or a program accessing spool entries has System Administrator status, provided this status was attained within a VSE component such as IUI, ICCF, or CICS, or by specifying the master password (MPWD=) in the POWER macro.

The VSE/POWER status report (see Figure 6 on page 45) shows the Spool Access Protection mode currently set.

Activating Spool Access Protection

VSE/POWER Spool Access Protection will be *activated* if the user has specified SEC=YES in the IPL SYS command and the SET SECAC=SYS VSE/POWER autostart statement.

Spool Access Protection Eligibility

A spool entry is *Spool Access Protection eligible* if it:

- Possesses either an origin or target userid, and
- Was not specified with a `* $$ JOB SECAC=NO` statement and did not inherit such a specification as output from such a job, and
- Is not an output spool entry specified with a `* $$ LST/PUN SECAC=NO` statement or with the PWRSPL flag SPLX1SNO.

If Spool Access Protection is active and a spool entry is Spool Access Protection eligible, then the entry will be referred to as a *Spool Access Protected entry*, and the origin and/or target user IDs, if any, become the *Authorized Access User IDs*.

If VSE/POWER Spool Access Protection is active, then every attempt will be made to tag a job spool entry with an origin user ID to obtain Spool Access Protection eligibility. As normally, the job entry (and its output by inheritance) will be tagged with the origin userid obtained according to the following sequence:

1. the FROM= operand, if any, from the `* $$ JOB` statement
2. the USERID=userid operand of the PWRSPL or the PUTSPOOL macros.

However, in the case of a PUTSPOOL macro, the USERID=userid operand of the PUTSPOOL macro is optional and the origin user ID may not be available (field SPUS). In this case VSE/POWER will search for a security logon user ID for the XECB PUTSPOOL program (available from the PUTSPOOL program's `* $$ JOB SEC=(userid,pwd)` or `// ID USER=userid,PWD=password` statement (if any). If the security logon userid is available, then this will be used as the job origin userid.

Authenticated User ID

An *authenticated user or program* has performed a security logon with a password. The associated *security logon user ID* for the *authenticated user or program* will be decided according to the following order of priority:

1. System component logon user ID (e.g., Interactive User Interface)
2. User program security logon user ID via the statement:

   `// ID USER=`*`cccccccc`*`,PWD=`*`password`*

3. User program security logon user ID via the statement

```
* $$ JOB SEC=(userid,pwd)
```

Programs or users will have their queue entry access via spool-access support (GET/CTL) or spool macro (GETSPOOL/CTLSPOOL) tagged with their security logon user ID.

Note: Jobs submitted to VSE/POWER from the Interactive User Interface (which is done via the SAS PUT function) are tagged with the Interactive User Interface user's logon userid.

Access Rules

A *Spool Access Protected entry* is protected against read/write access in case of:

- SAS PUT (Open-Append)
- SAS PUT (Open-Restart)
- SAS GET
- GETSPOOL

and against command manipulation in case of:

- SAS Commands
- CTLSPOOL

Note: The PALTER command cannot be used to alter the job or output SECAC= operand. Therefore, a spool entry's specification via the SECAC= operand is protected against manipulation.

If Spool Access Protection is active:

The users allowed to access a spool access protected entry are:

- A system administrator or central operator or user specifying the Master Password.
- A SAS GET/CTL or GETSPOOL/CTLSPOOL requestor who specifies the master password from the POWER macro (MPWD=).
- A SAS GET/CTL or GETSPOOL/CTLSPOOL requestor who has performed a security logon and whose security logon userid matches any of the spool entry's Authorized Access User IDs.

  Note: Entries with a target remote userid (REMOTE=remid or DEST=(,userid) which indicates a RJE remote userid) can be accessed by a user with a matching remote userid regardless of the Authorized Access User IDs (RJE users can also be password authorized similar to the Interactive Interface and VM/CMS users). However, it should be noted that RJE,BSC users cannot be protected as closely as RJE,SNA users since for RJE,BSC only a BSC line can be password protected, but not the RJE,BSC remote userid.
- Any authenticated requestor accessing (but not manipulating) an output entry with the target userid 'ANY'.

  Note: If the output is to be accessible by any user, including those not authenticated, then you must additionally specify SECAC=NO for the output.
- A user console (CMS, for example) whose console userid matches the spool entry's Authorized Access User ID.
- Local RDR/LST/PUN/XMT tasks.

If Spool Access Protection is not active:

The former rules of access apply as documented in "Scope of GET/CTL Access to Queue Entries" in *VSE/POWER Application Programming*.

Protection of Jobs/Output Arriving via PNET or POFFLOAD Tape

When jobs or output arrive via PNET or a POFFLOAD tape on a system where Spool Access Protection is active, from another system where Spool Access Protection was not active (including VSE/POWER 6.5 and earlier releases), Spool Access Protection will only be activated for a spool entry if the job or output origin userid and/or output target userid are available (can be seen with PDISPLAY).

Protection of Jobs Arriving Via Local RDR

Jobs arriving via local RDR device (e.g., the VM/CMS reader, a local RDR card or tape device where spool entries are read via the PSTART RDR command) generally do not have an origin user ID and are therefore not eligible for Spool Access Protection. Such jobs can be protected by adding a `* $$ JOB FROM=userid` statement to the job (this is recommended for VM/CMS users). This results in an origin user ID being tagged to the job and its output.

Migration Considerations

When VSE/POWER Spool Access Protection is activated:

- existing spool entries in the VSE/POWER queues from an earlier VSE/POWER release will be Spool Access Protected if they have either an origin userid (job or output) or target userid (output only) displayed as the FROM= or TO= userid of the entry.
- *vendor products and/or customer-written programs* that access VSE/POWER spooling queues may have some difficulty when Spool Access Protection is activated:
  - Programs issuing the GETSPOOL/CTLSPOOL macro previously could access any spool entry without regard to the spool entry's matching origin or target userid(s). Now, if a spool entry does have an origin or target userid, and the GETSPOOL/CTLSPOOL program does not have system administrator authority, the program must perform a security logon with the same origin or target userid to gain access, e.g. via `* $$ JOB SEC=(userid,pwd)` or via `// ID USER=cccccccc,PWD=password`. To obtain system administrator authority, the use of the VSE/POWER Master Password should be considered.
  - Likewise, for programs issuing a SAS GET/CTL request to access a spool access protected entry using an origin or target userid, where the SAS program does not have system administrator authority, the program must perform a security logon with the same origin or target userid. To obtain system administrator authority, the use of the VSE/POWER Master Password should be considered. Programs running in jobs submitted from IUI will use the security logon userid of the IUI user.
- when the local system is networking via PNET, all spool access commands (PALTER, PDELETE, etc.) commands from external NJE system users will be considered to be ″authenticated″ (the userid came from a security logon) if the non-local system:
  - is not VSE
  - is down-level from VSE 2.4
  - is VSE 2.4 and its Spool Access Protection option is inactive.

  This is to aid the migration to a VSE/POWER 6.4 or higher system. However, it may compromise system security. It is the responsibility of the system programmer to decide whether this is a security risk and to configure his network as required for his security needs.

Migration to or from VSE/POWER

Upward Migration to VSE/POWER 7.1

The following VSE/POWER Version.Release may have to be migrated to VSE/POWER 7.1 contained in z/VSE 3.1:

- VSE/POWER 5.x contained in VSE/ESA 1.x
- VSE/POWER 6.x contained in VSE/ESA 2.x

Jobs that were processed under VSE/POWER versions 4, 5 or 6 will run under VSE/POWER version 7.1. That means that you can offload (BACKUP, PICKUP or SAVE) the queues to tape under the old release and reload them again under version 7.1 of VSE/POWER.

VSE/POWER output files, spooled to tape (DISP=T) under an old version or release, can be printed or punched or reloaded to the queues under VSE/POWER 7.1.

Existing user-written exit routines must be recompiled under z/VSE 3.1 because the layout of VSE/POWER 7.1 control blocks has changed.

In a shared spooling environment, all participating VSE/POWER systems must be of the same version and release level because all systems access the same queue/data file. A VSE/POWER 7.1 system cannot coexist with a previous version or release. An unintentional VSE/POWER 7.1 cold or warm start with a startup phase generated by a POWER.A macro of a pre-7.1 release is flagged by message 1Q0GA. VSE/POWER startup continues successfully, but re-compilation of the startup phase with the 7.1 POWER.A macro is recommended to avoid message 1Q0GA in the future.

The following restrictions applying to an upward migration to VSE/POWER 7.1 from **pre-6.4** versions were introduced in 6.4 by extending the length of a queue record compartment from 256 to 384 bytes:

- In order to provide room for the same number of queue entries as in a previous release, the number of tracks of the VSE/POWER queue file (see //&rbl;EXTENT of IJQFILE) must be increased by a factor of 1.5. For example, 6 tracks should be increased to 9 tracks.
- As a result, the copy of the queue file in VSE/POWER partition Getvis (when private) or in VIO (when shared partition) also requires 1.5 times more storage. As a rule of thumb, use 'number-of-tracks times tracksize' for 'storage needed'.
- For programs that read VSE/POWER 7.1 offload tapes but which are not prepared to handle the longer queue record of 368 bytes, the downward migration facility of POFFLOAD can be used when creating tapes.

For details, see "Organization of the Queue File" on page 26.

Release Migration by Cold Start of the Spool Files

Due to format changes, a migration of VSE/POWER 5.x or 6.1 to 6.6 spool files to VSE/POWER 7.1 requires a cold start of the queue-, data- and account files. For startup options, see "FORMAT: Specifying File-Formatting Options" on page 492 which informs you how system down time can be reduced, as soon as two or more data file extents have to be formatted.

Release Migration During Warm Start

Migration of VSE/POWER 6.7 (and later) spool files to VSE/POWER 7.1 can be done during a VSE/POWER warm start. That means the lengthy process of offloading queues, reformatting existing queue-, data- and account file extents, and reloading the queues is no longer required. Instead during the first warm start of VSE/POWER 7.1 of a 6.7 (or later) queue file the operator is prompted as follows:

```
1Q0HI CURRENT LEVEL V7R1 OF VSE/POWER DIFFERENT FROM LEVEL VvRr OF QUEUE STARTING WARM
1Q0HD IF SPOOL FILE MIGRATION TO V7R1 IS INTENDED REPLY 'YES', ELSE 'NO'
```

When the reply is 'YES', the old spool files are accepted and are branded on disk as version 7.1 as soon as the initialization has completed by message 1Q12I. From this point in time those spool files are only accepted by VSE/POWER 7.1 and subsequent releases.

In case of shared spooling and the operator reply to 1Q0HD is 'YES' but VSE/POWER detects other shared spooling SYSID's still active, the warm start attempt is rejected by

```
1Q0JA SPOOL FILE MITRATION FAILED DUE TO OTHER SHARING SYSID(S) ACTIVE: n1,n2,...
```

Warm start may be retried after the other sharing SYSID's have been PEND'ed. Then a successful warm start migration brands the shared queue/data file by the first (and only) running system as version 7.1, and subsequently starting sharing systems can come up successfully only under VSE/POWER 7.1.

Note: Release migration during warm start is generally rejected, when

- 'Re-Allocation of the Queue File' has been requested for the **same** warm start
- 'Extension of the Data File' has not been completed during the last session and must be resumed during this warm start

Downward Migration of Tapes from Current Release

During the installation of a new current release there may be the need for passing queue entries on tape to a lower level VSE/POWER system. This works as long as the tape queue record has the same length as the queue record in use by the reading low level system. (For information on queue record length refer to "Queue Entries on Spool Tape" on page 528.) If, however, queue record lengths are different, the down level release rejects a tape with queue records longer than expected by message

1Q5AI **INVALID TAPE MOUNTED ON** *cuu* **FOR R-OFF,***cuu* RC=0002

You may overcome this limitation and request your current higher level VSE/POWER to produce an Offload tape suitable for a low level system by specifying the target level in the POFFLOAD BACKUPxx/PICKUPxx/SAVExx command. For details refer to "Format 4: Backup, Pickup, or Save for a Previous Release of VSE/POWER" on page 327.

Note: Only unlabeled tapes may be migrated downward to releases prior to VSE/POWER 6.1.

Controlling VSE/POWER Using Macros, Commands, and Statements

You control VSE/POWER by the following means:

Generation macros
are used when modifying the IBM-defined options of VSE/POWER to meet your installation needs. For a description of these macros see Chapter 3, "Tailoring VSE/POWER," on page 55.

Autostart statements
are used by the operator to define an automatic VSE/POWER startup for frequent use. For a description of the autostart statements see Chapter 7, "VSE/POWER Autostart Statements," on page 471.

Commands
are used by the operator to make a request to the system.

Central operator commands
allow to start and stop processing of a job or its output, display and change processing parameters, control the transmission of jobs and output between nodes in a network, or control the queues at another node in the network. For a description, see Chapter 5, "VSE/POWER Operator Commands," on page 191.

Remote job entry commands
are used to enter and manipulate jobs from a remote workstation. Many of the requests made by the remote terminal operator are similar to those of the central operator, as described above. The RJE commands are described in the *VSE/POWER Remote Job Entry* manual.

Job entry statements (JECL)
To spool a job and its output under VSE/POWER, you can either run it in a VSE/POWER controlled partition with its normal JCL delimiters (// JOB and /&), or you can use, in addition to the JCL statements, the VSE/POWER-oriented job entry language JECL, which allows to specify when and how your job will be processed.

Spool-access support macros and statements
are used to access VSE/POWER input and output queues from a user program. For a description, see "Spool-Access Support Macros" in *VSE/POWER Application Programming*.

How VSE/POWER Works

VSE/POWER Tasks

VSE/POWER is structured as 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 an input device.
- *Punch or list task* – It writes spooled output to an output 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 workstation.
- *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 queuing 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, shared-spooling timer control, and dynamic partition scheduling.

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

Processing I/O Requests

VSE/POWER intercepts input and output requests issued by programs running in partitions under control of VSE/POWER. VSE/POWER reads input from input devices or other connected processors; it queues this input into its reader 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 spool files (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, a card punch, a tape drive, a remote workstation, or another connected processing unit.

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

Restrictions for Programs Running under VSE/POWER

1. VSE/POWER does not support spooling for programs that include any of the following:
   - EXCP real
   - CCWs with indirect data addressing (IDA flag set)
   - CCWs with data chaining (CD flag set)
   - Format-1 CCWs (Format Control flag in ORB set).

   If these are used in a program, the affected job might be canceled.
2. The CCB-CCW addresses and data areas for the 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 LTA if this area is owned by the program
   - In the SVA (24-Bit)– applies only to write-request CCWs and I/O areas and not to CCBs.

Hardware Support

VSE/POWER supports the same devices as z/VSE. For a list of the devices supported, please refer to the section "Device Type Codes" in *z/VSE System Control Statements*.

However, there are these exceptions:

Devices NOT Supported
Optical- and magnetic-character equipment
Terminal printers
The IBM 7770 Audio Response Unit
The Advanced Function Printing (AFP) devices

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.
The Advanced Function Printing (AFP) devices[3]

Supported Telecommunication Devices

In general, VSE/POWER supports all BSC devices and systems that emulate the line discipline of the
IBM 2770,
IBM 2780,
IBM 3741, or
IBM 3780

For details, see *VSE/POWER Remote Job Entry*.

VSE/POWER supports the following BSC and SDLC terminals:
IBM 2770
IBM 2780
IBM 3741, Model 2 in 2780 emulation mode
IBM 3741, Model 4
IBM 3770 in BSC or SNA mode [4]
IBM 3780
IBM 3790 with the RJE facility

How VSE/POWER Supports Certain Devices and FCBs

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

3. AFP devices are not supported by VSE/POWER as a spooled printer device or as a local list task device, but spooling of Composed Page Data Stream (CPDS) for AFP printing is supported. For details see "CPDS Record Spooling and Page Counting" on page 145.

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

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 I/O Unit

VSE/POWER supports the 3540 only as 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.

For more details on the VSE/POWER support of the 3540 device, see "Using an IBM 3540 Diskette Unit" on page 142.

IBM 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 141.

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 you chose) VSE/POWER replaces the dollar signs by a character string depending on the printer:

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
$$$$ For any other printer type

Therefore, you can use for the actual printing a printer of a different type than the one specified at creation time (spooling time of output). But you must supply FCB images with phasenames according to the above conventions and with channel positions describing an identical page layout.

Using device-independent FCBs is recommended when spooling - for performance reasons - to a 3800 printer but actually printing on a non-3800 printer. For 3800-spooled printers, the sequence of searching and activating FCB specifications is the following:

1. FCB= ... in a * $$ LST statement
2. FCB= ... in a SETPRT command/statement
3. FCB= ... in a SETDF command
4. LTAB= ... in a * $$ LST statement
5. LTAB= ... of the POWER macro (or its default)

Note: The first and last half-inch of a 3800 page cannot be used for printing, and this is already observed at spooling time. Therefore, any FCB or even LTAB used to describe a 3800 page should cover the additional inch.

System Considerations

IBM VSE/POWER is a function of VSE Central Functions 7.1. VSE Central Functions is part of IBM z/VSE 3.1.

VSE/POWER operates in the static partition F1 of the generated z/VSE system. It can service as many other static and dynamic partitions as are available. VSE/POWER is allocated in a private address space. See also “Starting VSE/POWER in Private or Shared Address Space” on page 98.

The serviced partitions have traditionally been spooled static partitions (BG, F1,...,FB). The serviced partitions operating under VSE/POWER control may also be dynamic partitions. For detailed information, please see “Dynamic Partition Support” on page 132.

In most cases, the processing priority of the VSE/POWER partition should be higher than those of the partitions under its control. In fact, this simplifies startup, reduces confusion, improves problem/data accumulation, furnishes VSE/POWER account information and generally is a much cleaner, clearer operational environment.

However, for performance reasons, you may need to give the VTAM and/or CICS partition a higher dispatching priority. For standard spooled output, however, VSE/POWER should have a higher priority.

The NPC option on the PSTART command allows you to bypass the priority check and to start a partition under control of VSE/POWER with a higher dispatching priority.

If the PSTART command is issued for a dormant partition (VSE/POWER controlled partition in z/VSE STOP state) no optional operand specified in the PSTART command can take effect. Message 1R80I is issued to inform the operator.

Program Product Requirements

VSE/POWER can communicate with any system which supports IBM NJE protocols, for example JES2 and JES3 for an OS/390 node, and RSCS for a VM node.

If SNA sessions are required for terminal job entry or networking, you need IBM VTAM release, which is part of the z/VSE base package. If the terminal equipment is linked via a channel-attached controller, you need IBM NCP, which is an optional product of z/VSE. For information on the contents of z/VSE and its available optional products, see *z/VSE Planning*.

Processing Requirements

VSE/POWER has the following requirements:

1. A minimum SETPFIX LIMIT storage, EXEC SIZE specification, and partition ALLOC amount as listed in Table 1. In most cases, the following is sufficient for the VSE/POWER partition:
   - a SETPFIX LIMIT=150K
   - an EXEC powerphase,SIZE=1000K
   - a partition ALLOC of 1,800 KB with room for the queue file copy in partition Getvis (assuming 1000 queue entries requiring about 400 KB) and room for Getvis workspace of VSE/POWER functions.
2. Space on disk storage for its spool files.

The storage sizes in Table 1 allow a minimum set of VSE/POWER phases to be loaded into virtual storage and at least one task to be active.

Table 1. Minimum Storage Requirements for VSE/POWER Startup

	Partition Requirements (in KB)			
	SETPFIX LIMIT storage[1]	POWER phases storage	EXEC... ,SIZE= storage	Partition ALLOC storage for - 400 KB queue file Getvis copy[3] - 100 KB Getvis workspace[2]
Requirements for PNET, RJE, Shared Support	74	836	910	1410

Notes:

1. This minimum must be increased to provide fixable workspace for tasks to run.
2. This lower limit must be increased to provide Getvis workspace for tasks to run.
3. This amount provides space for about 1000 queue entries and must be increased for a larger queue file.

An additional 32 KB is required from the System Getvis area for the IPW$$NU phase.

If you plan to use the RJE,SNA (remote job entry with SNA) function or the PNET,SNA (networking with SNA) function, you must choose VTAM as your telecommunication access method.

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

An IBM 37xx Communications Controller
A communication adapter for the IBM ES/9000 (9221)

The attachment of BSC terminals 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 37xx Communication Controller in 270x emulation mode.
- A communication adapter for the IBM ES/9000 (9221)

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 terminals or BSC connected nodes concurrently. Through switched operation, more terminals may be in the network non-concurrently.
- Up to 250 terminals, SNA in synchronous data link control (SDLC) mode and BSC. 100 of these terminals may be BSC. Through switched operation, more terminals may be in the network non-concurrently.
- For every SNA terminal: one console, one logical reader, one logical punch, up to three logical printers, and up to six simultaneous sessions.
- The maximum SETPFIX LIMIT storage that VSE/POWER can use is 2048KB.
- Up to 99,998 queue entries on spool files.
- Up to 32 data file extents and a total of 2,147,483,647 DBLKs.
- A highest job number 65,535. Job numbers are assigned in wraparound mode.
- Up to 127 uniquely addressable output segments, before a new job number is assigned to the next group of 127 segments.
- The execution account record reflects up to 303 Start I/O Table entries. If I/O requests to more devices have been started during a job step, the surplus Start I/O Table entries are dropped.
- Up to 250 concurrent Spool Access Support (SAS GET/PUT/CTL/GCM) tasks — this limit may be increased by the command PVARY MAXSAS,nnnn.

Spool File Requirements

When modifying the IBM-supplied spool file amounts, consider that 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.

Organization of the Queue File

The queue 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. You can inquire about the extent characteristics of the active queue file by

using the PDISPLAY Q command (see “Format 7: Displaying Various Status Information” on page 238). The file contains, besides internal records, one queue file record per VSE/POWER job [5].

The queue file is organized in queue record blocks whose size depends on the device being used (see Figure 4 on page 34). Normally, a queue record block contains 32 queue records, each of which resides in a compartment of 384 bytes. If the queue file resides on an FBA device, the queue record block comprises 24 FBA blocks. Thus, the number of FBA blocks allocated for the queue file should be a multiple of 24; otherwise the remaining FBA blocks are wasted. For further details on queue records, refer to “Analyzing Queue File and Data File in Dumps” on page 523.

Method of Operation

During its first startup, the queue file must be formatted (also called ″cold start″) to create the chain of ’free’ queue records. At the same time, a copy of the queue file on disk is placed into the VSE/POWER partition storage to speed up all queue record access during spooling operations. Whenever free queue records are used to house ″used″ reader/list/punch queue entries, or whenever important queue record attributes are altered by the operator command, or when finally used entries are deleted and returned to the free chain, VSE/POWER also reflects all these modifications in the queue file on disk. Chaining information, on the other hand, is only maintained in the storage copy of the queue file until (by PEND command) VSE/POWER terminates in an orderly fashion and commits the total queue file from storage back to disk. A subsequent warm start copies the queue file again from disk to partition storage, so that VSE/POWER can immediately resume with the same queue contents as before shutdown.

When VSE/POWER terminates abnormally or is re-IPLed without an orderly shutdown, however, a subsequent warm start automatically switches to ″recovery warm start″. All queue records are then re-collected one by one from the queue file on disk, and their chaining information is reconstructed up to the point of VSE/POWER termination.

Copy of Queue File in Partition Getvis Area - VSE/POWER in Private Space

This method is chosen by VSE/POWER if it is started in a partition allocated in private space. In this environment, ample partition Getvis space can be made available. When not enough partition Getvis space is provided at startup time, VSE/POWER attempts to put the queue file into the VIO area and warns the operator by message 1Q24I.

The queue file in partition Getvis is a contiguous amount of storage that competes with active VSE/POWER tasks, when they request I/O areas, etc., from the partition Getvis-24 area. Whenever you increase the queue file or you want to provide more dynamic Getvis storage for active tasks, you should increase the ALLOC specification and/or adjust the SIZE value for the VSE/POWER partition. For details, refer to “The Size of the VSE/POWER Partition” on page 44.

When the partition Getvis-24 area is exhausted, and all available storage below the 16 MB line has been allocated to the VSE/POWER partition, you can still extend its ALLOC beyond the 16 MB line. The contiguous amount of the queue file in

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

storage is then requested from the (upper end of the) partition Getvis-31 area. This leaves all partition Getvis-24 to control blocks requested dynamically by active VSE/POWER tasks.

Copy of Queue File in Virtual Storage (VIO) - VSE/POWER in Shared Space

VSE/POWER startup in a shared partition places by default a copy of the queue file into the virtual storage area (VIO space). The VIO space can be seen as an extension of the page data set and is used as system work space. When not enough VIO space is defined at startup time, VSE/POWER attempts to put the queue file into the partition Getvis area and warns the operator by message 1Q1EI.

For how to provide VIO space, see "Virtual Storage (VIO Space)" on page 48.

Tracking the Queue File in Storage

The statistics status report (see "The Size of the VSE/POWER Partition" on page 44) reflects the storage consumption in partition Getvis by:

```
          QUEUE FILE STOR. COPY IN PART. GETVIS (TOTAL)    nnn KBytes
(optional) QUEUE FILE STOR. COPY, PART IN GETVIS-31        mmm KBytes
```

and in the VIO area by:

```
          QUEUE FILE STORAGE COPY VIO SPACE                nnn KBytes
```

Tracking the Queue File on Disk

The spool file characteristics report (see sample under "Format 7: PDISPLAY Q" on page 268) shows the currently remaining free queue records, the distribution of used queue records over the various queues, and the disk location of the VSE/POWER queue file by:

```
1R49I  QUEUE FILE 008% FULL - 1295 FREE QUEUE RECORDS
1R49I  USED QUEUE RECORDS: 111, CRE-Q: 3, DEL-Q: 0
1R49I  RDR-Q: 49, LST-Q: 51, PUN-Q: 2, XMT-Q: 6
1R49I  QUEUE FILE EXTENT ON CKD-150, SYS001, 1005, 15
```

Organization of the Data File

The data file, named IJDFILE, consists of up to 32 extents, and all extents of the file must reside on devices of the same type. If the file resides on an FBA device, an extent can start on any block boundary. You can inquire about the extent characteristics of the active data file by using the PDISPLAY Q command (see "Format 7: Displaying Various Status Information" on page 238).

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, and the same volume must not be addressed with different logical units. The VSE/POWER system:

- Uses the logical units SYS002 to SYS033 for accessing the data file extents on different volumes
- Provides its own I/O control block for each data file extent in the SETPFIX LIMIT storage.
- Can improve spooling performance during its private subtasking by
  Parallel I/O preparation, provided the data file is split into more than one extent on the same volume
  Parallel I/O preparation **and** scheduling, provided the data file extents are even spread over different volumes

Figure 1 shows sample DLBL and EXTENT statements that describe different data file extents to VSE/POWER.

One Extent Only

```
// ASSGN SYS002,230
// DLBL IJDFILE,'POWER.DATA.FILE',99/365,DA
// EXTENT SYS002,,1,0,15,90
```

Three Extents on One Volume

```
// ASSGN SYS002,230
// DLBL IJDFILE,'POWER.DATA.FILE',99/365,DA
// EXTENT SYS002,,1,0,15,90
// EXTENT SYS002,,1,1,105,60
// EXTENT SYS002,,1,2,165,90
```

Two Extents on Different Volumes

```
// ASSGN SYS002,230
// ASSGN SYS003,231
// DLBL IJDFILE,'POWER.DATA.FILE',99/365,DA
// EXTENT SYS002,,1,0,15,90
// EXTENT SYS003,,1,1,165,90
```

Two Extents on One Volume, One More on Another

```
// ASSGN SYS002,230
// ASSGN SYS003,231
// DLBL IJDFILE,'POWER.DATA.FILE',99/365,DA
// EXTENT SYS002,,1,0,15,90
// EXTENT SYS002,,1,1,105,60
// EXTENT SYS003,,1,2,165,90
```

Figure 1. DLBL and EXTENT Statements

The space available on the data file is arranged in **DBLK groups**. A DBLK group is the **smallest unit** of space that is allocated to a VSE/POWER job. Every DBLK group contains an integer number of DBLKs (data blocks). The smallest DBLK group consists of two DBLKs. 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).

For the logical arrangement of DBLKs and DBLK groups, see "Analyzing Queue File and Data File in Dumps" on page 523.

Tracking the Data File on Disk

The spool file characteristics report (see sample under "Format 7: PDISPLAY Q" on page 268) shows the status and location of the VSE/POWER data file by:

```
1R49I  DATA FILE 003% FULL - 1230 FREE DBLK RECORDS
1R49I  CURRENT DBLK SIZE=07548, DBLK GROUP SIZE=00008
1R49I  DATA FILE EXTENT 1 ON CKD-151, SYS002, 6690, 2025
```

Queue File to Data File Relation

Figure 2 shows the relationship between the queue file and the data file, assuming for example 3 DBLKs per DBLK group.

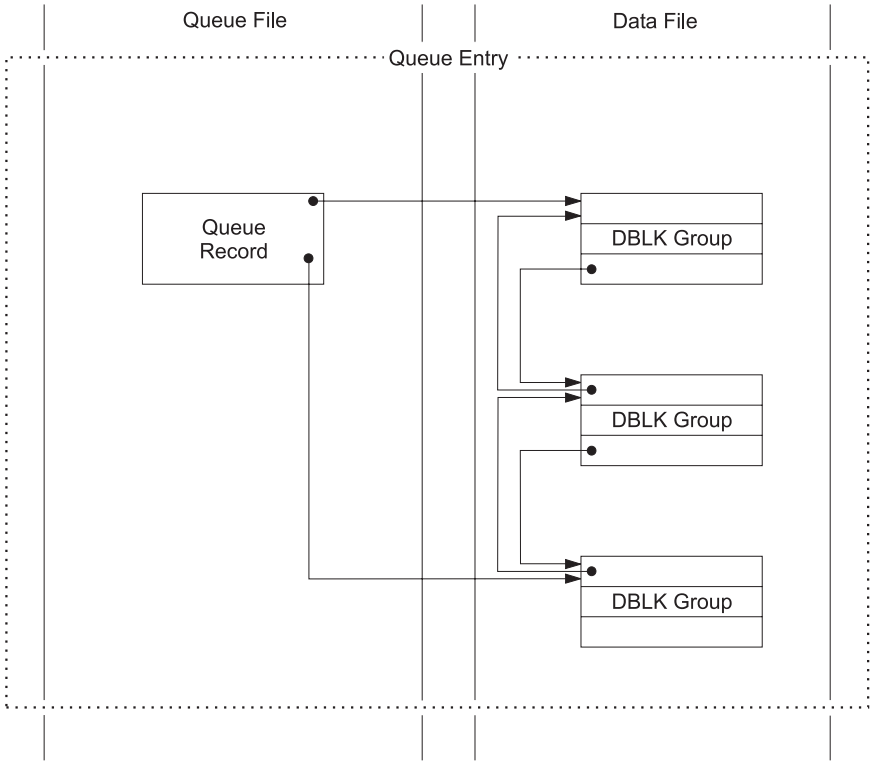


Figure 2. Relationship between Queue File and Data File

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 or output 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 linked together by a forward and backward chain. Every DBLK group contains two 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.

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.

Life Cycle of VSE/POWER Queue Entries

Whenever user data is spooled by VSE/POWER, a queue entry (see Figure 3 on page 32) is created which occupies

1. one queue record from the free queue record chain of the queue file, plus
2. one or more data block (DBLK) groups from the free DBLK group chains of the data file.

A queue entry may be created by various VSE/POWER tasks:

1. Execution writers handling WRITE requests for VSE/POWER spooled devices
2. Local or RJE card readers reading in jobs
3. Network receivers reading in jobs our output from other NJE nodes
4. SAS-PUT tasks creating jobs or output for a Spool-Access Support program

5. PUTSPOOL tasks creating jobs
6. POFFLOAD tasks LOADing jobs or output

Queue entries 'in creation' can be made visible using the PDISPLAY CRE command (see page "Format 3: Displaying Information of the Create Queue" on page 235), which presents them as members of the logical Create (CRE) queue[6].

When spooling has completed, an in-creation queue entry is added to the RDR, LST, PUN, or XMT queue depending on its specified queue entry attributes. These queues are true physical queues with entries linked together according to their disposition, processing class, and priority. Using the PDISPLAY RDR|LST|PUN|XMT commands, all queue entries 'queued' for processing can be made visible.

As soon as a VSE/POWER processing task selects a dispatchable 'queued' entry (e.g., a RDR job for execution or a LST entry for printing), the queue entry enters the 'active' state, that means it remains in its queue displayed with Disposition '*'. After processing, the queue entry may either be

- re-queued with a hold disposition in the same queue, or be
- deleted, meaning that it is either
  - flagged for later (delayed) deletion because there are still browsing (read-only) tasks that are viewing the queue entry from another partition. During this time, queue entries are in the 'delayed deletion' state. They can be made visible by the PDISPLAY DEL command, which shows them belonging to the logical Deletion (DEL) queue[6]. To list all queue records and all DBLK Groups kept 'in deletion', refer to the PDISPLAY STATUS output example (Figure 7 on page 46). As soon as all viewing has been terminated for such a queue entry, it will be 'finally freed', that means the queue record is returned to the Free Queue Record Chain and its DBLK group(s) are added to a Free DBLK Group Chain, or it is
  - passed by 'delayed freeing' to the DELetion (DEL) queue and the initiation/termination task is invoked to 'finally free' the entry..

Under all transition states, namely 'in-creation, 'queued/inactive', 'queued/active', and 'delayed deletion', the data spooled for queue entries can be viewed by Spool Access GET BROWSE tasks from other partitions. At the same time, the queue entries and their resource consumption can be tracked by the cited PDISPLAY commands.

The following diagram illustrates the queue entry management process:

6. The Create and Deletion queues are referred to as *logical queues*. They are not physically linked sets of allocated queue entries like the physical RDR/LST/PUN/XMT queues but rather simply subsets of allocated queue entries in the 'in-creation' and 'delayed deletion' states, respectively. The entries in these logical queues are located for display purposes by stepping sequentially through all allocated queue entries and examining the state indicator. Also, the PDISPLAY A(active) command can be used to show in-creation queue entries belonging to currently active VSE/POWER tasks.

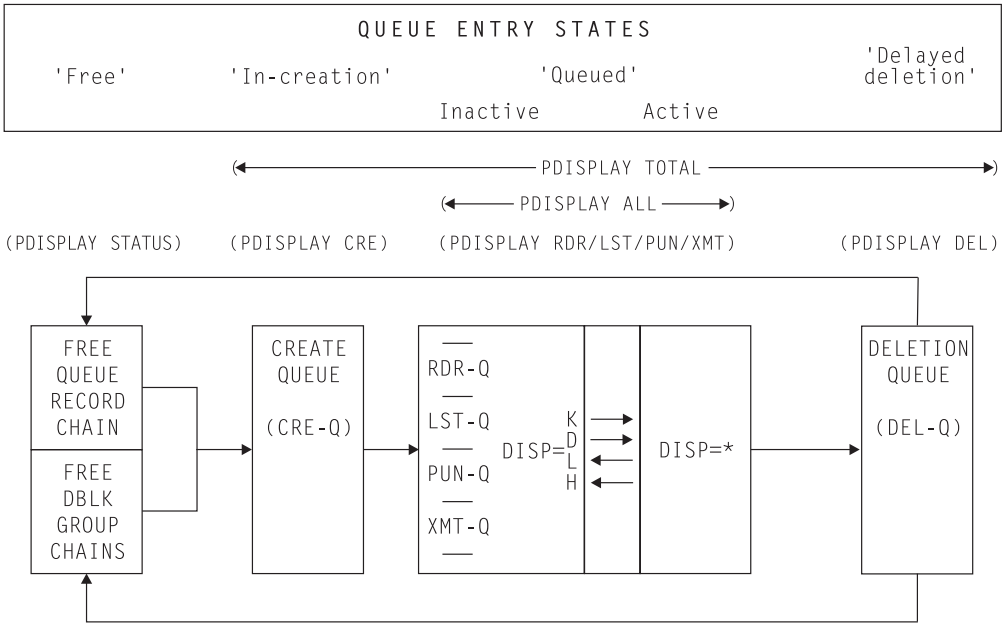


Figure 3. Life Cycle States of Queue Entries and Associated Display Commands

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. You can inquire about the extent characteristics of the active account file by using the PDISPLAY Q command (see "Format 7: Displaying Various Status Information" on page 238).

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.

Tracking the Account File on Disk

The spool file status report (see "Format 7: Displaying Various Status Information" on page 238) shows the status and location of the VSE/POWER account file by:

```
1R49I  ACCOUNT FILE 1 % FULL
1R49I  ACCOUNT FILE EXTENT ON CKD-151, SYS000, 8715, 90
```

Estimating Disk Space for the VSE/POWER Spool 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. A queue file accommodating 1000 queue records may meet operational requirements. You may need a larger queue file if you plan to share your system's spool files or process jobs from other nodes in a network or use your VSE/POWER queues as a repository.

Note that performance is not impacted by increasing the queue file, but partition GETVIS (or VIO when VSE/POWER runs in a shared address space) must be increased for the queue file copy in storage. For example, a queue file with 60 tracks on a 3380 contains 5726 queue records and needs 2.1 MB of virtual storage in VSE/POWER partition GETVIS (for the calculation, see Figure 4 on page 34).

Calculate Tracks from Number of Queue Records: Based on the selected maximum number of queue records, you can determine (1 and 2) the number of tracks needed for the queue file and (3) the necessary amount of partition Getvis or VIO space and the number of 'usable' queue records as follows:

1. Divide the chosen number of queue records by the number of queue records per block (see Figure 4 on page 34 for the appropriate values) and round up the quotient if the remainder is not zero. Add one to the quotient (for the master record which occupies 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) 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 this number of tracks by the number of queue record blocks per track, and subtract one for the master record. Now, multiply the **result** by the queue record block size (see Figure 4 on page 34), which gives you the necessary partition Getvis or VIO space in bytes. In case of VIO, the latter value must be rounded up to the next multiple of 32K/64K, depending on the mode of operation.

   If you multiply the **resulting** number of queue record blocks by the number of queue records per block (see Figure 4 on page 34) and subtract two internal ("heading" and "trailing") queue records, one obtains finally the number of **usable queue records**.

For example, assuming the total number of allocated queue records on a IBM 3380 is 1000, then

1. The number of queue record blocks is: 1000/32+1 = 33 (rounded up)
2. The number of tracks is: 33/3 = 11
3. The amount of partition Getvis or VIO space to hold the entire queue file is: (11 tracks x 3 - 1) x 12 KB = 384 KB

   The number of usable queue records is (11 tracks x 3 - 1) x 32 - 2=1022.

For an FBA device, the number of queue record blocks would also be 1000/32+1=33, assuming a total of 1000 queue records. Since a queue record block comprises 24 FBA blocks, the total number of FBA blocks required for the queue file would be 792.

Device Type	Queue Record Block Size	Number of Queue Records per Block	Number of Blocks per Track
3380	12,288	32	3
3390 as 3380	12,288	32	3
3390	12,288	32	4
FBA	12,288	32	-
SCSI (=FBA)	12,288	32	-

Figure 4. Device Type - Queue Record Block Relationship

Calculate Number of Queue Records from Tracks: Based on the selected number of queue file tracks per device type, you can determine the necessary amount of partition GETVIS or VIO space in bytes (step 1) and the number of usable queue records (step 2) as follows:

1. Multiply the chosen number of tracks by the number of blocks per track (of the selected device in Figure 4) and subtract one for the master record. Now multiply the **result** by the queue record block size (see Figure 4), which gives you the necessary partition GETVIS or VIO space in bytes to maintain the storage copy of the queue file.
2. Multiply the **resulting** number of queue record blocks by the number of queue records per block (see Figure 4) and subtract two internal (″heading″ and ″trailing″) queue records, which obtains the number of 'usable' queue records.

The following device oriented tables help one to determine easily the number of 'usable' queue records for a given number of tracks or FBA blocks on disk

Table 2. Device 3380 Tracks vs. Usable Queue Records

Tracks on Device 3380	Partition GETVIS KB or VIO Space	Number of Usable Queue Records
11	384	1022
22	780	2078
44	1572	4190
88	3156	8414
176	6324	16862
352	12660	33758
704	25332	67550
1042	37500	99998

Table 3. Device 3390 Tracks vs. Usable Queue Records

Tracks on Device 3390	Partition GETVIS KB or VIO Space	Number of Usable Queue Records
8	372	990
16	756	2014
32	1524	4062
64	3060	8158
128	6172	16350
256	12276	32734
512	24564	65502
782	37500	99998

Table 4. FBA Blocks vs. Usable Queue Records

Tracks on FBA Device	Partition GETVIS KB or VIO Space	Number of Usable Queue Records
792	384	1022
1594	780	2078
3168	1572	4190
6336	3156	8414
12672	6324	16862
25344	12660	33758
50688	25332	67550
75024	37500	99998

Large Queue File Considerations: For a queue file requiring more than 6 MB from the VSE/POWER partition GETVIS area, one should allocate the partition larger than 16 MB. Then the contiguous amount of the queue file in storage is requested from the upper partition GETVIS-31 area. This leaves more partition GETVIS-24 for control blocks which dynamically support active VSE/POWER internal tasks.

The maximum queue file is limited to 99,998 usable queue records contained in 3125 queue record blocks plus one for the master record. These 3126 blocks can be contained completely in the tracks or FBA blocks as presented in the last row of the above Device tables (see Table 2 on page 34,Table 3 and Table 4) . When more tracks or FBA blocks are specified for the queue file extent, VSE/POWER issues the message 1Q0CI informing about the extraneous amount of unused disk space.

For a queue file containing more than 20,000 entries (used or free), certain functions that do not address the class chains of the RDR/LST/PUN/XMT queues but, however, have to scan the **complete** queue file, may take some seconds of time before finishing. This performance impact of 1–2 seconds per 10,000 searched queue records, can be observed for the following functions and commands:

- reading the queue file from disk during warm start
- performing queue file recovery at abnormal warm start
- saving the queue file to disk at PEND shutdown time
- requesting the seldom-used commands
  - PDISPLAY CRE

- PDISPLAY DEL
- PDISPLAY TOTAL
- PDISPLAY BIGGEST
- PDISPLAY Q

The Size of the Data File

The size of the data file depends on the amount of space needed for every 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, if data is printed or punched immediately after creation and then deleted. If, however, you use the VSE/POWER queues as a repository, you should choose a larger size depending on the retention period.

  This approach should avoid that VSE/POWER runs out of spool file space (see "Options When Data File Is Full" on page 179). 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 you defined for a data block.

The Size of the Data Block: You define the size of a data block, the physical record size that VSE/POWER writes out to the data file, 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 **
3380	7,476	84	53
3390 as 3380	7,548	85	53
3390	7,548	85	53
FBA	7,680	87	54
SCSI (=FBA)	7,680	87	54

* 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 partition GETVIS 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. Therefore, the DBLK value takes main influence on the performance of VSE/POWER's spooling characteristics. 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.

The Size of the DBLK Group: 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 8 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 typical type of queue entries existing in your VSE/POWER environment, 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, equal to 'DBLK-value times DBLKGP-size' number of bytes. Then little data file disk space is wasted per queue entry. However, at the same time, for a big output entry, many of such small DBLK groups have to be allocated, which costs an extra read/write request on the data/queue file per DBLK group.

If, for example, you run many jobs producing large output, you can set the DBLK group size relatively high, up to the equivalent of several tracks. Then, only few of such big DBLK groups must be chained to house all data of a big queue entry. However, at the same time, any small job will also obtain a big DBLK group as a minimum, which wastes data file space. The system may then run out of spool space, although not all DBLKs are filled on the data file. Both examples demonstrate the advantage and disadvantage of either small or big DBLK groups. In the end, the product of

```
'DBLK-value bytes times DBLKGP-size'
```

has to be tailored to your specific job/output mix, provided the shipped default of e.g. 7,476 bytes times 8 does not satisfy your needs:

Table 5. Recomended DBLK Values

Device Type	Record size max(corresponding no. of records/track)
3380	4820(9), 5492(8), 6356(7), 7476(6),9076(5), 11476(4), 15476(3), 23476(2), 47476(1)
3390	4566(11), 5064(10), 5726(9), 6518(8), 7548(7), 8906(6), 10796(5), 13682(4), 18452(3), 27998(2), 56664(1)
SCSI	7680, 15360, 23040, 30720

The FULL=YES operand of the PDISPLAY command or the PDISPLAY BIGGEST command show the number of allocated DBLK groups for every queue entry. The PDISPLAY STATUS 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. In case the data file is indeed suddenly exhausted, see "Options When Data File Is Full" on page 179).

For detailed information on DBLK group chaining, see "Analyzing Queue File and Data File in Dumps" on page 523.

Size of the Account File

For every 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 help you estimate: a disk space of about 250KB can accommodate the account records for about 300 VSE/POWER jobs with an average of three account records per job.

Extending Existing VSE/POWER Spool Files

Whenever the shipped or currently existing queue, data, or account file is to be altered in your z/VSE environment, refer to "Extending VSE/POWER Files" in *z/VSE Administration*, so that all linked definitions and assignments are updated consistently. For extension of the queue file, refer to "Re-Allocate Queue File During Warm Start" on page 41. For extension of the data file, refer to "Extending the Data File During Warm Start." or to "Options When Data File Is Full" on page 179.

Two types of VSE/POWER file extensions exist:

- Extending the queue or data file during a cold start
- Extending the queue or data file during a warm start

Extending the Queue or Data File During a Cold Start

VSE/POWER may extend or relocate the queue, data, and account files by a cold start, which reformats all file extents. Therefore, jobs must be saved to tape using POFFLOAD PICKUP or BACKUP and the account file can also be saved to tape or another disk extent using PACCOUNT. The queue and data files are linked, meaning that whenever one file must be changed or extended, the other one is also reformatted. The ACCOUNT file can be reformatted independently. Formatting is triggered by FORMAT=Q|D for the queue and data files, FORMAT=A for the account file, and FORMAT=Q|D,A for all files.

Note: When two or more data file extents have to be formatted , system down time can be reduced by combining a cold start of the queue file and the first data extent with a subsequent extension of the data file during warm start.

Extending a Queue or Data File During Warm Start

Extending the Data File During Warm Start: Another type of extension exists for the VSE/POWER data file. It may be extended by one or more additional extents during a warm start (FORMAT=NO). The following description applies to the

general case of many extents. “Warm Start Extension” is triggered by appending the new EXTENT statements with ascending sequence numbers to the existing EXTENT statements of the data file. Use the PDISPLAY Q command to obtain the currently used specifications of the data file extent(s). Make sure that they match to the specifications of the existing // EXTENT statements, in order to avoid a 1QD3A warm start failure. The new statements must be appended because VSE/POWER accesses the DBLKs on the existing extents as a contiguous stream starting with DBLK number ’0’ and ending with DBLK number ’n’. DBLKs on already existing extents are referenced by their number, which cannot be changed. Therefore, data file extension addresses only the new extents for formatting. This step is postponed until the warm start has been completed. While the additional extents are being formatted, spooling is no longer disabled as during formatting of queue and data file at a cold start.

Leaving the already spooled data on the existing extents untouched and formatting the additional extents ″on the fly″ eliminates the need for saving all queue entries and reduces system downtime.

When specified, the additional extents are detected during VSE/POWER warm start, and the operator is informed by message

```
1QD7A  mm ADDITIONAL EXTENT(S) FOUND FOR EXTENSION OF EXISTING DATA FILE
       WITH nn EXTENT(S)
```

A list of the already existing data file extents is displayed to the operator by message(s)

```
1QD2I  EXISTING DATA FILE EXTENT NO. mm FOUND IN IJDFILE DLBL/EXTENT
      (// EXTENT SYSxxx,volid,1,nnn,start,length)
```

Then for each additional extent the operator is prompted for confirmation by message

```
1QD2D  DATA FILE EXTENT NO. mm - FOR FORMATTING REPLY 'YES' ELSE 'NO'
      (// EXTENT SYSxxx,volid,1,nnn,start,length)
```

When the operator responds with ’NO’ to one of the new extents, a VSE/POWER warm start ignores **all** appended extents and starts up with the existing data file from the previous session. The existing data file is protected against incorrect EXTENT statements by VSE/AF BAM OPEN which verifies the extent sequence order written to the VTOC(s) during the last cold start or a previous data file extension.

When the operator has confirmed all new data file extents by ’YES’, VSE/POWER checks that the logical unit numbers are in ascending order for **all** extents, that all disks are assigned uniquely, that all extents reside on the same device type, and that all EXTENT statements of existing extents match their previous location. The location of the new extents is checked for availability by opening one IJDTEST file comprising all planned new disk areas. All these steps ensure that the additional extents can be addressed by VSE/AF BAM OPEN without any obstacles. The OPEN process records the new extents (and also the old extents) in the VTOC (Volume Table of Contents) of their assigned disks. Formatting of the new extents starts after VSE/POWER startup has been completed as indicated by message

```
1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO. mm STARTED
```

The named new extent is now formatted while VSE/POWER continues spooling. When formatting has been completed for this extent, the operator is informed that the new DBLK groups are now available for spooling.

```
          1QD6I  FORMATTING OF NEW DATA FILE EXTENT No.mm COMPLETED,
                 nnnnn FREE DBLKGPS ADDED
```

This process is repeated until all extents have been formatted.

The following sequence of sample messages accompanies this ″dynamic extension of the data file″ on the console during a VSE/POWER warm start, and the subsequent PDISPLAY Q command returns the actually active data file extents:

```
F1 0001 1QD7A   3 ADDITIONAL EXTENT(S) FOUND FOR EXTENSION OF EXISTING DATA
        FILE WITH 1 EXTENT(S)
F1 0001 1QD2I  EXISTING DATA FILE EXTENT NO. 1 FOUND IN IJDFILE DLBL/EXTENT
        (// EXTENT SYS002,SYSWK1,1,000, 6690, 2025)
F1-0001 1QD2D  DATA FILE EXTENT NO. 2 - FOR FORMATTING REPLY 'YES' ELSE 'NO'
        (// EXTENT SYS003,SYSWK3,1,001, 15, 105)
1 yes
F1-0001 1QD2D  DATA FILE EXTENT NO. 3 - FOR FORMATTING REPLY 'YES' ELSE 'NO'
        (// EXTENT SYS004,SYSWK4,1,002, 15, 135)
1 yes
F1-0001 1QD2D  DATA FILE EXTENT NO. 4 - FOR FORMATTING REPLY 'YES' ELSE 'NO'
        (// EXTENT SYS004,SYSWK4,1,003, 2550, 450)
1 yes
F1 0001 1QD4I  VERIFYING LOCATION OF ADDITIONAL DATA FILE EXTENT(S) BY OPEN
        FOR 'IJDTEST'
F1 0001 4744D OVERLAP ON UNEXPRD FILE  IJDTEST  SYS003=153  SYSWK3
   YET.ANOTHER.FILE
F1-0001
1 delete
F1 0001 1QD5I  LOCATION OF ADDITIONAL DATA FILE EXTENT(S) VERIFIED
        SUCCESSFULLY
. . .
F1 0001 1Q12I  VSE/POWER 7.1.0 INITIATION COMPLETED
. . .
F1 0001 1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO. 2 STARTED
F1 0001 1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO. 2 COMPLETED, 66 FREE
        DBLKGPS ADDED
F1 0001 1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO. 3 STARTED
F1 0001 1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO. 3 COMPLETED, 84 FREE
        DBLKGPS ADDED
F1 0001 1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO. 4 STARTED
F1 0001 1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO. 4 COMPLETED, 281 FREE
        DBLKGPS ADDED
. . .
d q
F1 0001 1R49I  QUEUE FILE 003% FULL - 1364 FREE QUEUE RECORDS
F1 0001 1R49I  USED QUEUE RECORDS: 42, CRE-Q: 1, DEL-Q: 0
F1 0001 1R49I  RDR-Q: 22, LST-Q: 14, PUN-Q: 1, XMT-Q: 4
F1 0001 1R49I  QUEUE FILE EXTENT ON CKD-150, SYS001, 1005, 15
F1 0001 1R49I  DATA FILE 015% FULL - 1446 FREE DBLK GROUPS
F1 0001 1R49I  CURRENT DBLK SIZE=07548, DBLK GROUP SIZE=00008
F1 0001 1R49I  DATA FILE EXTENT 1 ON CKD-151, SYS002, 6690, 2025
F1 0001 1R49I  DATA FILE EXTENT 2 ON CKD-153, SYS003, 15, 105
F1 0001 1R49I  DATA FILE EXTENT 3 ON CKD-154, SYS004, 15, 135
F1 0001 1R49I  DATA FILE EXTENT 4 ON CKD-154, SYS004, 2550, 450
F1 0001 1R49I  ACCOUNT FILE 1 % FULL
F1 0001 1R49I  ACCOUNT FILE EXTENT ON CKD-151, SYS000, 8715, 90
```

The formatting period between message `1QD6I...STARTED` and `1QD6I...COMPLETED` may be lengthy if a large data file extent is added. During this time:

- a statistics display (see Figure 7 on page 46) presents the IJDFILE header line with `'n+m' EXTENTS`. In the console sample above, it would present ′1+3′ or ′2+2′ or ′3+1′, and show ′4′ as soon as formatting has completed.
- a PDISPLAY Q command (see sample under “Format 7: PDISPLAY Q” on page 268) appends `IN FORMAT` to the `1R49I DATA FILE EXTENT` message for each not yet formatted extent.

- A PEND command is accepted, but VSE/POWER's final termination does not take place until formatting of the current extent has completed. Formatting of the remaining unformatted extent(s) is postponed until the next warm start.
- A PEND IMM or PEND FORCE command, or VSE/POWER abnormal termination, interrupts formatting abruptly. The subsequent VSE/POWER warm start (with the already added data file extents) resumes formatting automatically.

Considerations for Shared Spooling: Data file extension during warm start can only be triggered on one sharing system when all other systems have been terminated normally. When the extending system has successfully checked the planned location and informed the operator by message

```
1QD5I  LOCATION OF ADDITIONAL DATA FILE EXTENT(S) VERIFIED SUCCESSFULLY
```

the other sharing systems can be started, but from this point in time they must use the same data file DLBL and EXTENT statements as the extending system. They will inform the operator by message

```
1QD6I  FORMATTING OF NEW DATA FILE EXTENT NO mm DETECTED ON SYSID sysid
```

that the extending system 'sysid' has not yet finished formatting. If the extending system terminates abnormally before formatting is complete, queue file recovery resumes formatting and is entered either

- automatically by another already started sharing system, or
- at the next warm start of the failing system. If this cannot be restarted, use the 'PRESET sysid' command from another sharing system to request recovery.

Note: The statistics display (see Figure 7 on page 46) names all 'ACTIVE SYSID'S FOUND:' together with the shared time intervals.

Re-Allocate Queue File During Warm Start: VSE/POWER offers to re-allocate the queue file during a warm start and thereby extend it without reformatting the queue and data file. Re-allocation places the queue file at a different location and is able to extend it at that moment. The queue file re-allocation does not affect already spooled queue entries because any additional queue records of the increased queue file are added as free queue records on top. The master record, which controls queue and data file access, and which is the last record of the queue file, is again placed at the end of the extended queue file. This leaves the linkage between existing queue entries and the data file as is, and therefore the data file need **not** be formatted.

Process Flow of Re-Allocation: The new function is triggered when VSE/POWER detects a **new** DLBL/EXTENT statements for IJQFILE addressing a disk area[7] assigned by **SYS001**, identified by a 'file-id' not yet listed in the appropriate VTOC. The previously used queue file must address its original disk area by DLBL/EXTENT statements for IJQFOLD assigned to **SYS034** with its original 'file-id'. Therefore you must use a different 'file-id' if you plan to re-allocate the VSE/POWER queue file to an area on the same disk, which already contains IJQFOLD.

Note: To enable VSE/POWER to use SYS034 for accessing IJQFOLD, the number of programmer logical units must at least be 35. This number is specified by the NPGR statement of $0JCL ASIPROC. Beginning with VSE/ESA 2.7 the default has been raised to 40.

7. Remember that the storage copy of the queue file is kept in partition GETVIS or in VIO. So when you plan to extend the queue file by re-allocation, increase partition GETVIS respectively VIO also.

After changing the DLBL/EXTENT/ASSGN for IJQFILE and IJQFOLD, the next warm start of VSE/POWER tries to open the not yet existing IJQFILE. VSE/AF handles the OPEN request and fails to locate the 'file-id' of the IJQFILE DLBL in the VTOC of the disk assigned to SYS001. Hence the operator is informed by message:

```
4601I NO FORMAT 1 LABEL FOUND  IJQFILE  SYS001=cuu  volid
```

and VSE/POWER assumes that re-allocation is intended and informs the operator by message:

```
1QE1I  RE-ALLOCATION PROCESS STARTED FOR VSE/POWER QUEUE FILE
```

Now the previously used queue file IJQFOLD assigned to SYS034 is opened. After a successful OPEN, VSE/POWER checks that:

1. the IJQFOLD EXTENT specifies the same location as obtained from the OPEN process
2. the IJQFILE DLBL/EXTENT is specified
3. the IJQFILE EXTENT does not overlap the IJQFOLD EXTENT
4. the IJQFILE and JQFOLD do not reside on different disks with the same VOLID

When the old and new queue files are accepted, the operator is informed about their extents and is asked to confirm re-allocation by messages:

```
1QE3I  IJQFOLD: // EXTENT SYS034,volid,1,n,start,length
1QE3I  IJQFILE: // EXTENT SYS001,volid,1,n,start,length
1QE3D  CONFIRM QUEUE FILE RE-ALLOCATION FROM IJQFOLD TO IJQFILE BY 'YES' ELSE 'NO
```

- For *NO*, VSE/POWER drops the re-allocation attempt but continues to warm start the existing queue file via IJQFOLD.
- For *YES*, VSE/POWER informs the operator by message 1QE4I that the extent of the new queue file is now verified for being unused. Therefore a temporary work file named IJQTEST is opened which uses the same extent definition as specified by the IJQFILE EXTENT statement. Then the number of queue record blocks is calculated to make sure that the new queue file can take all queue records of the old queue file.

If the new queue file is larger than the old one, new free queue records are defined and added on top of the already existing ones. So the existing queue records keep their original queue record number and preserve their linkage to the appropriate DBLK groups.

To ensure data integrity and unique recovery from failure, the time interval in which the new and the old queue file exist both on disk must be reduced to a minimum. Therefore VSE/POWER uses the temporary file name IJQTEST to build the new queue file:

1. first, all queue records and the master record are written to the new disk location,
2. next, VSE/POWER creates the VTOC entry for IJQFILE, thereby changing the temporary work file to the permanent new queue file.

At that moment successful re-allocation is confirmed by message:

```
1QE6A  RE-ALLOCATION FOR IJQFILE COMPLETED, nnnnn FREE QUEUE RECORDS ADDED
```

Finally VSE/POWER destroys the original IJQFOLD by overwriting it with a dummy work file (file-id: 'VSE.POWER.DUMMY.FILE').

Changes to Trigger Re-Allocation: The following example shows the changes needed for DLBL/EXTENT/ASSGN statements to accomplish queue file re-allocation. The following is an excerpt from STDLABEL.PROC and DTRPOWR.PROC:

```
// DLBL IJQFILE,'VSE.POWER.QUEUE.FILE',99/366,DA
// EXTENT SYS001,DOSRES,1,0,1005,15
- - -
// ASSGN SYS001,DISK,VOL=DOSRES,SHR      POWER QUEUE FILE
```

The following is an excerpt from STDLABEL.PROC and DTRPOWR.PROC **with** changes:

```
// DLBL IJQFILE,'VSE.POWER.QUEUE.FILE',99/366,DA
// EXTENT SYS001,SYSWK2,1,0,2400,450
// DLBL IJQFOLD,'VSE.POWER.QUEUE.FILE',99/366,DA
// EXTENT SYS034,DOSRES,1,0,1005,15
- - -
// ASSGN SYS001,DISK,VOL=SYSWK2,SHR      POWER QUEUE FILE NEW
// ASSGN SYS034,DISK,VOL=DOSRES,SHR      POWER QUEUE FILE OLD
- - -
```

Messages during Re-Allocation: The subsequent VSE/POWER warm start will show the following sequence of messages on the console for a successful queue file re-allocation:

```
F1 0001 // JOB POWSTART
        DATE 04/17/2005, CLOCK 10/17/51
F1 0001 4601I NO FORMAT 1 LABEL FOUND  IJQFILE  SYS001=152  SYSWK2
F1 0001 1QE1I  RE-ALLOCATION PROCESS STARTED FOR VSE/POWER QUEUE FILE
F1 0001 1QE3I  IJQFOLD: // EXTENT SYS034,DOSRES,1,000, 1005, 15
F1 0001 1QE3I  IJQFILE: // EXTENT SYS001,SYSWK2,1,000, 2400, 450
F1-0001 1QE3D  CONFIRM QUEUE FILE RE-ALLOCATION FROM IJQFOLD TO IJQFILE BY
        'YES' ELSE 'NO'
1 yes
F1 0001 1QE4I  VERIFYING LOCATION OF NEW QUEUE FILE IJQFILE BY OPEN FOR
        'IJQTEST'
F1 0001 4744D OVERLAP ON UNEXPRD FILE  IJQTEST  SYS001=152  SYSWK2
   VSE.PRIVATE.ANY.FILE
F1-0001
1 delete
F1 0001 1QE5I  LOCATION OF NEW QUEUE FILE IJQFILE VERIFIED SUCCESSFULLY
F1 0001 1QE6A  RE-ALLOCATION FOR IJQFILE COMPLETED, 41760 FREE QUEUE RECORDS
        ADDED
F1 0001 1RB4I  PLOAD NETWORK DEFINITION TABLE NDT271 LOADED
F1 0001 1Q20I  AUTOSTART IN PROGRESS
```

Follow-Up after Re-Allocation: After successful queue file re-allocation, use PDISPLAY Q to verify the new location of the queue file.

```
pdisplay q
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R49I  QUEUE FILE 000% FULL - 43139 FREE QUEUE RECORDS
F1 0001 1R49I  USED QUEUE RECORDS: 27, CRE-Q: 0, DEL-Q: 0
F1 0001 1R49I  RDR-Q: 19, LST-Q: 8, PUN-Q: 0, XMT-Q: 0
F1 0001 1R49I  QUEUE FILE EXTENT ON CKD-152, SYS001, 2400, 450
- - -
```

Then withdraw the IJQFOLD DLBL/EXTENT/ASSGN statements from STDLABEL.PROC and DTRPOWR.PROC.

Considerations for Shared Spooling: When VSE/POWER re-allocates the queue file, it requires exclusive access to the old and the new queue files. If VSE/POWER shares queue and data file with other systems (SHARED=Q), special considerations insure queue file integrity. VSE/AF LOCK requests for 'IJQFL.volid' are issued for the new queue file and, if JQFOLD resides on a separate disk, also for the old queue

file to ensure that only 1 VSE/POWER shared spooling system initializes at a time. The re-allocating system, having both files locked, then verifies, that no other sharing system is still working with the old queue file.

Fixable- and Virtual-Storage Requirements

The Size of the VSE/POWER Partition

The virtual storage provided by the ALLOC specification for the VSE/POWER partition comprises the following storage areas (identified by the corresponding message(s) of Figure 6 on page 45):

1. SETPFIX-LIMIT storage needed by VSE/POWER for fixing pages, also "fixable area"

   ```
   FIXABLE STORAGE ALLOCATED TO VSE/POWER          nnn K-Bytes (see MSG1)
   ```

2. Pageable storage, also "pageable area", to hold VSE/POWER executable code

   ```
   VIRTUAL STORAGE OCCUPIED BY VSE/POWER PHASES    nnn K-Bytes and
   UNUSED STORAGE REMAINING BELOW SIZE BOUNDARY    nnn K-Bytes
   ```

3. Partition Getvis storage, including working buffers of the VSE/POWER tasks and the copy of the queue file, if in a private partition

   ```
   TOTAL PART. GETVIS-24 STORAGE ALLOCATED         nnn K-Bytes (see MSG2)
   ```

Figure 6 on page 45 shows the storage layout of the VSE/POWER partition in two flavors:

- A (left-hand side): a large environment with ample partition Getvis so that the storage copy of the queue file resides beyond the 16 MB line (see MSG3).
- B (right-hand side): a standard environment with allocation below the 16 MB line and the queue file residing in Getvis-24 storage (see also MSG3).

Figure 6 on page 45 also illustrates how the partition framework is specified by the ALLOC, SIZE, and SETPFIX LIMIT values. In addition, the figure is annotated by messages from a statistics status report (see PDISPLAY STATUS command in Figure 7 on page 46, which reveal dynamic storage exploitation of a running VSE/POWER system.

Use the display of Figure 7 on page 46 together with the storage layout of Figure 6 on page 45 whenever you want to:

- question the statistics values collected by VSE/POWER versus the VSE specifications for ALLOC, SIZE, and SETPFIX LIMIT and versus the VSE recorded Getvis /PFIX consumption of the VSE/POWER partition Fn using the attention routine MAP Fn and GETVIS Fn command. From the MAP command display (or the ASIPROC startup values), look up the K-byte values for
  - ALLOC (V-SIZE + GETVIS in MAP output)
  - SETPFIX LIMIT of VSE/POWER partition ASIPROC (reflected in the MAP REAL display)
  - The SIZE command for the VSE/POWER partition or the SIZE operand on the // EXEC POWER statement
  - GETVIS (GETVIS in MAP output)
- tune the current storage layout by lowering the SIZE value, so that the "UNUSED STORAGE REMAINING BELOW SIZE BOUNDARY" does not exceed 40K.
- react to "fixable area shortage" (MSG 1Q59I, 1Q7BI, 1Q78I) by increasing the SETPFIX LIMIT value together with SIZE, or even ALLOC, so that the extension of a "lower" storage area is reflected in "higher" storage areas.

- react to "pageable area shortage" (MSG 1Q08I) by increasing the SIZE value, or even ALLOC, to avoid a reduction of the partition Getvis storage at the same time.
- react to "partition Getvis-24 shortage" (MSG 1Q85I, 1Q7AI) by increasing the ALLOC value.

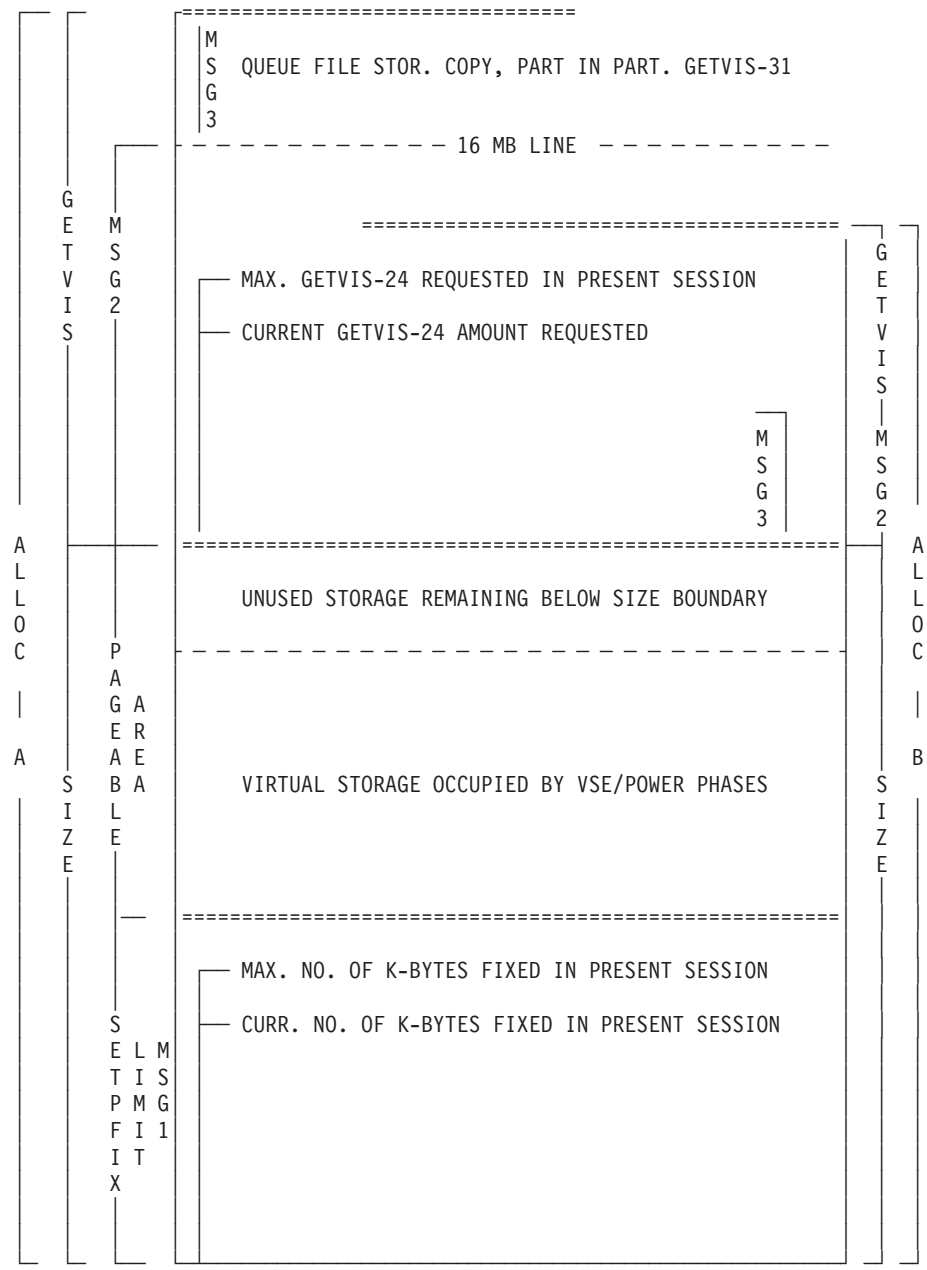


Figure 6. Storage Requirements for the VSE/POWER Partition

```
        1R46I VSE/POWER 7.1.0 STATUS FOR POWERSLB ON 04/15/2005 TIME 13/54/59
                  LAST QUEUE/DATA FILE COLD START ON 12/04/2004
1          PRESENT SESSION START (TURBO-DISP.-NP) ON 04/15/2005 TIME 13/51/03
2          APPLIED SERVICE LEVEL  >> DY-BASE <<   OF 11/27/2004
3       1R46I  NODE = POWER261 , SYSID = 2

        1R46I  QUEUE FILE     IJQFILE
           TOTAL NUMBER OF TRACKS                                    8 TRACKS
           TOTAL NUMBER OF QUEUE RECORDS                           734 RECORDS
4           FREE QUEUE RECORDS (INCL. 10 FOR CUSHION)              699 RECORDS
5           USED QUEUE RECORDS                                      35 RECORDS
6            QUEUE RECORDS IN DELETION                               2 RECORDS
7            QUEUE RECORDS LOST DUE TO I/O ERROR                     0 RECORDS
           MAX. NO. OF Q-REC'S USED IN PRESENT SESSION               35 RECORDS
           MAX. NO. OF Q-REC'S USED SINCE LAST COLDSTART             66 RECORDS
8          QUEUE FILE STOR. COPY IN PART. GETVIS (TOTAL)           276 K-BYTES
9          QUEUE FILE STOR. COPY, PART IN PART. GETVIS-31          276 K-BYTES

10      1R46I  DATA FILE      IJDFILE                                2 EXTENTS
           TOTAL NUMBER OF TRACKS                                14025 TRACKS
           TOTAL NUMBER OF DBLK-GROUPS                            1265 GROUPS
11          FREE DBLK-GROUPS (INCL. 20 FOR CUSHION)               1227 GROUPS
12          USED DBLK-GROUPS                                        38 GROUPS
13          DBLK-GROUPS IN DELETION                                  4 GROUPS
14          DBLK-GROUPS 'BEING FREED'                                2 GROUPS
15          DBLK-GROUPS LOST DUE TO I/O OR LOGIC ERROR               0 GROUPS
           MAX. NO. OF DBLK-GPS USED IN PRESENT SESSION             38 GROUPS
           MAX. NO. OF DBLK-GPS USED SINCE LAST COLDSTART           79 GROUPS
           DATA BLOCK GROUP SIZE                                    8 DBLKS
           DATA BLOCK SIZE                                       7548 BYTES
           SPOOL LIMIT PERCENTAGE                                  90 %
        DATA FILE DBLK GROUP TRACING                              ENABLED

16      1R46I  ACCOUNT FILE   IJAFILE
           TOTAL NUMBER OF TRACKS                                   90 TRACKS
           PERCENTAGE OF FILE THAT IS FILLED                         1 %

        1R46I  GENERAL STORAGE STATISTICS
           FIXABLE STORAGE ALLOCATED TO VSE/POWER                  152 K-BYTES
            MAX. NO. OF K-BYTES FIXED IN PRESENT SESSION            72 K-BYTES
            CURR. NO. OF K-BYTES FIXED IN PRESENT SESSION           72 K-BYTES
            NO. OF TIMES TASKS WAITING FOR PFIXED STORAGE            0 TIMES
           VIRTUAL STORAGE OCCUPIED BY VSE/POWER PHASES            659 K-BYTES
17         UNUSED STORAGE REMAINING BELOW SIZE BOUNDARY            213 K-BYTES
           TOTAL PART. GETVIS-24 STORAGE ALLOCATED               11265 K-BYTES
18          MAX. GETVIS-24 REQUESTED IN PRESENT SESSION             67 K-BYTES
19          CURRENT GETVIS-24 AMOUNT REQUESTED                      61 K-BYTES
            NO. OF TIMES TASKS WAITING FOR GETVIS-24 STOR            0 TIMES
           SYSTEM GETVIS STORAGE USED BY VSE/POWER                  32 K-BYTES

        1R46I  GENERAL TASK STATISTICS
           MAX. NO. OF TASKS ACTIVE AT ONE POINT IN TIME            24 TASKS
           CURRENT NUMBER OF ACTIVE TASKS                           17 TASKS
           MAXIMUM LIMIT OF SAS TASKS - MODIFY BY PVARY            250 TASKS
           CURRENT NUMBER OF ACTIVE SAS TASKS                       11 TASKS

20      1R46I  SPOOL ACCESS PROTECTION MODE                          INACT
21      1R46I  TIME INTERVALS FOR SHARED SPOOLING (SYSID=2, SHARED=Q,A):
           T1 =   5 SEC., T2 =   0 SEC., T3 =  60 SEC., T4 = 180 SEC.
           ACTIVE SYSID'S FOUND: 1,2
        1R46I  DYNAMIC PARTITION SCHEDULING STATISTICS
           SUCCESSFUL DYNAMIC PARTITION ALLOCATION                  90 TIMES
           UNSUCCESSFUL DYNAMIC PART. ALLOC. AS IN 1Q3FI            10 TIMES
            - DUE TO NO MORE ALLOCATION SPACE                        5 TIMES
            - DUE TO NO MORE DYN. PARTITION AT ALL                   3 TIMES
            - DUE TO NO MORE SYSTEM GETVIS SPACE                     2 TIMES
            - DUE TO NO MORE VSE/POWER SETPFIX STORAGE               0 TIMES
            - DUE TO NO MORE VSE/POWER GETVIS-24 STORAGE             0 TIMES
        1R46I  NOTIFY SUPPORT STATISTICS
           NUMBER OF NOTIFY MESSAGES LOST                           0 MSG(S)
22      1R46I  SUPPORT FOR RETRIEVAL OF JOB EVENT MESSAGES
           MESSAGE QUEUE SIZE                                      20 MSG(S)
           MAXIMUM MSG(S) LOST FOR ANY USER                         0 MSG(S)
```

Figure 7. Example of a Statistics Display, presented for PDISPLAY STATUS

Note: This statistics display applies to storage layout A (left-hand side) of Figure 6 on page 45, with a storage allocation beyond 16 MB.

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.

1 For information on a session started with the Turbo Dispatcher activated or not, refer to “Determining if the Multiprocessor Support is Active” on page 129.

2 The APAR number of the latest applied VSE/POWER level set PTF and its creation time. For a newly shipped release, the number reads 'DY-BASE'.

3 If nodename is not generated by a Network Definition Table, dashes ('-------') will be shown. If SYSID is not defined in the POWER macro or set by a SET SYSID=system-id autostart statement, a single dash will be shown.

4 The free Queue Record counts include 10 elements called the "queue record cushion", and the free DBLK-Group counts include 20 elements called the "DBLK group cushion". They are reserved for queue display requests creating a list queue entry at times when no more spool space is available. For details, see "Retrieving Messages" in *VSE/POWER Application Programming*.

5 Used queue records comprise those used by

- 'in-creation' queue entries
- entries in the RDR/LST/PUN/XMT queues
- 'in-deletion' queue entries (see extra display line)
- queue record block unable to read in (see extra display line)

6 All queue entries of the Deletion Queue (see PDISPLAY DEL).

7 Both the "MAX. NO. OF Q-RECS USED..." lines contain "NUMBER OF QUEUE RECORDS LOST DUE TO I/O ERROR". I/O error also means incorrect queue record identification as flagged during recovery by message 1QZ0I, RC=0011.

8 If the copy of the queue file resides in the VIO area, the message reads:

```
QUEUE FILE STORAGE COPY VIO SPACE                 276 K-BYTES
```

9 This message appears only if the copy of the queue file resides partly or totally in the partition Getvis area beyond the 16 MB line. See also “Copy of Queue File in Partition Getvis Area - VSE/POWER in Private Space” on page 27.

10 The data file header line shows the number of data file extents. When extending the data file during a warm start (see “Extending the Data File During Warm Start” on page 38) was requested and formatting of the additional extent is not yet complete, this is indicated by 'n + 1' .

11 For an explanation, see **4** .

12 Used DBLK-groups comprise those

- used by 'in-creation' queue entries
- used by entries in the RDR/LST/PUN/XMT queues
- used by 'in-deletion' queue entries (see extra display line)
- 'being freed' of a queue entry, that has just left the DELetion queue but has not yet completed this process (see extra display line)
- lost due to I/O or logic error (see extra line)

13 All data block groups of queue entries in the Deletion Queue (see PDISPLAY DEL).

14 Data block groups of a queue entry that has already left the DELetion queue, but whose DBLK group(s) are just being written 'free' on the data file (displayed only if number greater than zero).

15 For more information, see "Analyzing Queue File and Data File in Dumps" on page 523.

16 If VSE/POWER was generated with job accounting support deactivated, the account file statistics are replaced by the message

```
NO ACCOUNT SUPPORT
```

17 This part of the pageable area is not needed for loading of VSE/POWER phases. It may be reduced to 40K by decreasing the SIZE value correspondingly and thereby gain additional partition Getvis space.

18 These maximum/current Getvis-24 consumption values are only collected from the VSE/POWER native GETVIS requests. The GETVIS Fn attention routine command may well state larger values because of
- Getvis area control information
- Getvis storage fragmentation
- Getvis storage requirements by VSE services (e.g., Librarian) when using SLI support
- Getvis storage requirements by OEM products

Consider using the GETVIS Fn, ALL/DETAIL command for better tracking of discrepancies between VSE/POWER and VSE/AF collected high-water marks.

19 For an explanation, see **18** .

20 Spool access protection mode ('ACTIVE' or 'INACT') as defined by the `SEC=` operand of the IPL `SYS` command and the VSE/POWER `SET SECAC=` autostart statement.

21 The time intervals are printed only if VSE/POWER is operating in a shared (queue file, or queue file and account file) spooling environment. For active SYSID's, the following are reported:
- currently active sharing systems
- systems that were previously active and terminated abnormally, but did not yet re-IPL for recovery

22 The lost message count allows you to determine if the default or specified message queue size is suitable or not. The status report also indicates if the GCM support is not available due to insufficient storage. This is displayed as follows:

```
SUPPORT NOT AVAILABLE: NO REAL STORAGE
```

Virtual Storage (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 if VSE/POWER is started in a shared partition. This 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 33. You define the VIO area during system startup either in the IPL ASI procedure or in response to message 0I03D during system startup. For further details on system startup, see "The Supervisor Parameters Command" in *z/VSE System Control Statements*.

System Getvis Area

In order to load the nucleus and the static partition control blocks into the SVA, 32KB processor storage is retrieved from the System Getvis area. This storage is released when VSE/POWER is terminated.

Shared Spooling Requirements

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

- Specify the following operands when you generate VSE/POWER:
  SHARED=(Q)|(Q,A)
  SYSID=*number*
  TIME=(t1,t2,t3)

  For a description of the range of values and the defaults of these operands, refer to "POWER Generation Macro" on page 57.
- Add the disk devices to be used for the VSE/POWER spool files as sharable during VSE initialization. To do so, specify 'ADD cuu,SHR' in your IPL procedure.
- Define a VSE lock file during VSE initialization. This file must be on a shared volume. To do so, specify a DLF statement in your IPL procedure. 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 *z/VSE Guide to System Functions*.
- 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. In the shipped z/VSE system, the label information for the VSE/POWER spool files are contained in the STDLABEL procedure in the sublibrary IJSYSRS.SYSLIB.
- Make sure that the shared disk is assigned to the VSE/POWER spool files. In the shipped z/VSE system, the spool file assignments are contained in the DTRPOWR procedure in the sublibrary IJSYSRS.SYSLIB.

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.

Notes:

1. The SYSID= operand of the POWER macro and the SET SYSID= autostart statement are even accepted for a non-shared VSE/POWER system (SHARED=NO). However, shared spooling is not enabled then. For a display of the generated SYSID, see also the PDISPLAY STATUS command example in Figure 7 on page 46.
2. To verify that sharing systems actually access the same queue and data file, use the facilities described under "Command/Message Passing Between Sharing Systems" on page 520.

If several VSE systems are running as virtual machines under control of VM, the CP command SET CPUID must be used to provide a unique VM CPUID for every VSE system. Shared spooling by systems under control of VM requires specific mini-disk definitions so that VM can use the correct RESERVE/RELEASE commands. For details refer to *VM/ESA Running Guest Operating Systems* , SC24-5522 and to *Controlling Multiple VSE Systems under VM/ESA* GG24-3847.

VTAM Generation and Interface Requirements

If you plan to use the RJE,SNA or PNET,SNA support of VSE/POWER, then VTAM must be included in your system (in either shared or private address space).

Your generated version of VTAM must meet the following requirements:

- The corresponding VTAM sublibrary must be included in the phase search chain of the VSE/POWER partition.
- VSE/POWER must be defined to 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 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 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 [8] 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.

If VSE/POWER runs VTAM applications, RJE,SNA or PNET,SNA for example, consider the following: for buffering of transmission data, ACF/VTAM uses now data spaces instead of buffers in its partition GETVIS. VSE/POWER shares *one* data space for *all* its VTAM applications with ACF/VTAM. This data space is defined by VTAM during ACB OPEN using the DSPACE parameter in the EXEC statement of the VTAM application program. VSE/POWER recommends a minimum of 2MB, for example,

```
// EXEC IPWPOWER,DSPACE=2M
```

For more information on VTAM, consult the manual *Planning for NetView, NCP, and VTAM*.

8. USS = Unformatted system services

For more information on VSE/POWER startup, see "How to Start VSE/POWER Interactively (Warm and Cold Start)" on page 105.

TCP/IP for VSE/ESA Setup and Interface Requirements

If you plan to use the PNET TCP or the PNET SSL function of VSE/POWER, support for a local TCP/IP host partition must be included in your system. These PNET functions use EZASMI macro requests, which enter the Language Environment (LE-C) support, which eventually addresses the BSD/C interface of TCP/IP for ESA. In order to provide macro and phase support for these request layers, the VSE/POWER partition search chain for phases must include the following sublibraries:

- IJSYSRS.SYSLIB for EZAMI phases (and macro)
- PRD1.BASE for TCP/IP for VSE/ESA and the interface code of LE-C to TCP/IP.
- PRD2.SCEEBASE for the LE base addressed by the EZASMI runtime environment.
- the search chain sequence of PRD1.BASE followed by PRD2.SCEEBASE should be established

For the PNET TCP function, there are no extra generation requirements in the POWER macro. However, the Network Definition Table (NDT), generated with a series of PNODE macro calls, must identify

- for a local node, a port number (default 175) on which VSE/POWER will listen for connection requests from remote TCP/IP linked nodes
- for non-local (remote) NJE nodes, a port number (default also 175) with a dotted decimal or symbolic IP address of the foreign TCP/IP host, via which the remote NJE node should be contacted.

For the PNET SSL function, there are also no extra generation requirements in the POWER macro. However, the Network Definition Table (NDT), generated with a series of PNODE macro calls, must identify

- for a local node, a secure port number (default 2252) on which VSE/POWER will listen for connection requests from remote TCP/IP SSL linked nodes. For additional SSL related operands and their defaults see also PNODE macro "Format for Local Node" on page 82.
- for non-local (remote) NJE nodes, a secure port number (default also 2252) with a dotted decimal or symbolic IP address of the foreign TCP/IP host, via which the remote NJE node should be contacted. For additional SSL related operands and their defaults see also PNODE macro "Format 5: Defining a Directly Linked SSL Node" on page 85.

Contacting the Local TCP/IP Host Partition

TCP/IP for VSE/ESA executes in a "main partition" (ID=00). Additional copies can execute in a "secondary partition" (ID=nn), and they are identified in the TCP/IP EXEC statement by the ID=nn parameter. When the PNET TCP function is to connect to the TCP/IP main partition, no extra provisions have to be made; see also *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*, SC33-6601. When the PNET TCP function is to address a secondary partition for its EZASMI macro requests, the underlying LE-C code must be informed by the following Job Control statement placed into the startup of the VSE/POWER partition:

```
// OPTION SYSPARM='nn'
```

Storage requirements due to TCP/IP connections

For any started networking node, independent of the PNET function type, a Node Control Block (1.5 KB) is acquired from the SETPFIX LIMIT storage of the VSE/POWER partition.
For PNET TCP or PNET SSL the additional storage acquired from the VSE/POWER Partition Getvis area depends on the following facts:

- type of nodes you have defined (nodes using the SSL feature or not)
- number of TCP and SSL nodes
- specified value for BUFSIZE in the PNODE macro

VSE/POWER uses the following components for its TCP/IP connections:

- EZASMI HLASM macro support
- LE Runtime support
- TCP/IP for VSE/ESA

Due to these components (and their potential future enhancements) storage requirements are made upon

- the System Getvis area, where most (2.7 MB) of the LE-runtime support has been loaded for IBM distributed systems. Note, if this support is not loaded into the SVA, up to 3.5 MB may be required additionally from the VSE/POWER Partition Getvis area.
- the VSE/POWER Partition Getvis area with the following storage consumption:

```
 Component Support for TCP nodes only                        760 KB
 Component Support for SSL nodes only                        880 KB
(Component Support for TCP and SSL nodes                    1300 KB)

 Component Support for each started SSL node            70 - 120 KB
           70 KB - when BUFSIZE=4KB
          120 KB - when BUFSIZE=32KB

 Component Support for each started TCP node             16 - 70 KB
           16 KB - when BUFSIZE=4KB
           70 KB - when BUFSIZE=32KB

 PNET Buffer Space for each active TCP or SSL node      16 - 130 KB
           16 KB - when BUFSIZE=4KB
          130 KB - when BUFSIZE=32KB



 Notes:
       - The 'Component Support' storage values may be partly
         obtained from the VSE/POWER Partition Getvis-31 area.
         The BUFSIZE values are those specified in the PNODE macro.
       - The 'PNET Buffer Space' storage is only obtained from
         the VSE/POWER Partition Getvis-24 area.
         The values reflect 2 buffers per (default) receiver and
         transmitter. This will increase when more receivers or
         transmitters are started by the PACT command.
         The BUFSIZE values are those agreed upon during sign-on.
```

Figure 8. VSE/POWER Partition Getvis Storage Consumption for PNET TCP and SSL

VSE/POWER Classes

Classes are a means to group jobs or output that require the same set of processing resources. Classes tell VSE/POWER in which partition a job is to run or on what printer an output entry is to be printed. A class is a logical assignment of a job to a partition or output to a printer. For example, you can specify that all jobs with a class specification of 'A' run in partition BG or corresponding output print on device 00E.

Classes are one character definitions. A class can be a character from A - Z or from 0 - 9.

Assigning Classes to Jobs and Output Entries

In order to tag classes to elements that are to be processed, specify the class operand of the * $$ JECL statements that surround a job when submitted to VSE/POWER for processing. Specify:

*** $$ JOB ...,CLASS=A**
to tag, for example, execution class A to this job

*** $$ LST ...,CLASS=P**
to tag, for example, printing class P to the output produced by this job.

Of course, you may also omit explicit specifications and use default rules for class assignment. In all cases job and output entries will appear in the VSE/POWER queues ready to be selected for processing according to their tagged classes.

Assigning Classes to Processing Resources

Assigning Class(es) to Static Partitions

Typically, at system start-up, the PSTART command is used to relate a job class to a partition. For example:

```
PSTART F3,AB3
```

In the above example, all jobs with a class specification of A, B, or 3 will process in partition F3.

The order in which you specify the classes is the order in which VSE/POWER processes jobs in the partition. In the above example, jobs with class A, will process first, then those with class B; finally those with class 3.

Up to 4 different classes can be assigned to a partition.

The system has assigned the numbers 0 - 9 as default partition-related classes. If you omit the operand, VSE/POWER assigns processing classes as follows:

```
class 0 - for BG
class 1 - for F1
class 2 - for F2
. . . .
class 9 - for F9
```

Note that characters are **not** partition-related classes. That is, 'A' does not refer to partition 'FA', nor 'B' to 'FB'.

All jobs with, for example, a CLASS=1 specification, will process in partition F1. Likewise, a job with the class specification of 0 *cannot* run in partition F5 but only in BG.

Consider grouping jobs according to characteristics. For example, all large jobs may be given class 'A', because those jobs are assigned a partition which is able to execute jobs that require a large partition size. Likewise, you may want to give small jobs a class that requires less partition resources.

Providing Dynamic Partition Classes

The static partitions are permanently existing processing resources which are related to job classes by the PSTART partition command. Dynamic partitions, on the other hand, are processing resources created only upon demand- namely the event that a dispatchable job with class 'M' (for example) has been added to the reader queue. Then VSE/POWER checks the 'Dynamic Class Table' for an 'M' entry which provides various attributes of potential dynamic partitions Mx (x=1...n)

- Allocates and starts the dynamic partition M1, for example, for processing of the mentioned job
- Controls stopping and de-allocation of partition M1 at end-of-job.

Provided a Dynamic Class Table has been defined containing dynamic classes with attributes of its dynamic partitions, one can consider the VSE/POWER start-up

```
PLOAD DYNC
```

command as a method to load-activate and enable its defined dynamic class resources for processing of jobs with equal class.

Assigning Class(es) to a List Task When Started for Printing

You can use the PSTART LST command to relate output to a printer. For example:

```
PSTART LST,008,A
```

In the above example, all output with class A will be routed to a printer with address 008.

As output-related class, you can specify A to Z or 0 to 9. Up to 4 different classes can be assigned to an output device.

Consider grouping the output of jobs according to devices to which the output is to be written. For example, you may want to send dump output to one printer and output that has to do with bills or receipts to another printer. Output from different jobs but using the same form number can be kept together in the list queue by matching class and priority specifications. This helps to reduce the forms mount effort.

Assume a z/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

Chapter 3. Tailoring VSE/POWER

This chapter describes the VSE/POWER generation macros within VSE/POWER that you may use to best utilize the program. But remember: VSE/POWER is operational once you have z/VSE installed, and it may well serve your needs without any further modifications.

Generating VSE/POWER Tables

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

1. A phase named **POWER** − It provides support in accordance with the *default* values documented under "POWER Generation Macro" on page 57.
2. A phase named **IPWPOWER** − It provides the support as phase POWER. However, the default settings for phase IPWPOWER are as follows:
   ACCOUNT=YES
   SPOOL=YES
   SUBLIB=P
   MEMTYPE=P
   NTFYMSG=100

Both versions relate only to the POWER macro; other macros, if required, must be added.

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 startup 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 "VSE/POWER Startup" on page 97.

Generation Macro Overview

For the conventions of showing command and statement formats, see Chapter 1, "Understanding Syntax Diagrams," on page 1. They also apply to the presentation of macro operands.

To use these macros you must be familiar with the rules for coding programs or routines in Assembler language. For details, see *Guide to DOS/VSE Assembler*.

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 57.

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 Generation Macro" on page 72.

PRMT The macro defines the hardware characteristics of an RJE terminal. For a description of the macro and its operands, turn to "PRMT Generation Macro" on page 75.

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 Generation Macro for Networking Support" on page 81.

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 Generation Macro" on page 90.

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.

Figure 9 on page 57 helps you select the generation macros and operands related to the available RJE, PNET, and shared spooling support.

To Have Support for:					Code the Macro (or Operand of the Macro) as Shown
Remote Job Entry		Networking (PNET)		Shared Spooling	
RJE/BSC	RJE/SNA	BSC	CTC,SNA,TCP,SSL		
X	X	X	X	X	POWER
		X	X		PNET
		X	X		NETEXIT
		X	X		XMTEXIT
X	X				OUTEXIT
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

Note: X means that the macro or operand applies.

Figure 9. Generation Macro Selection Table

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.

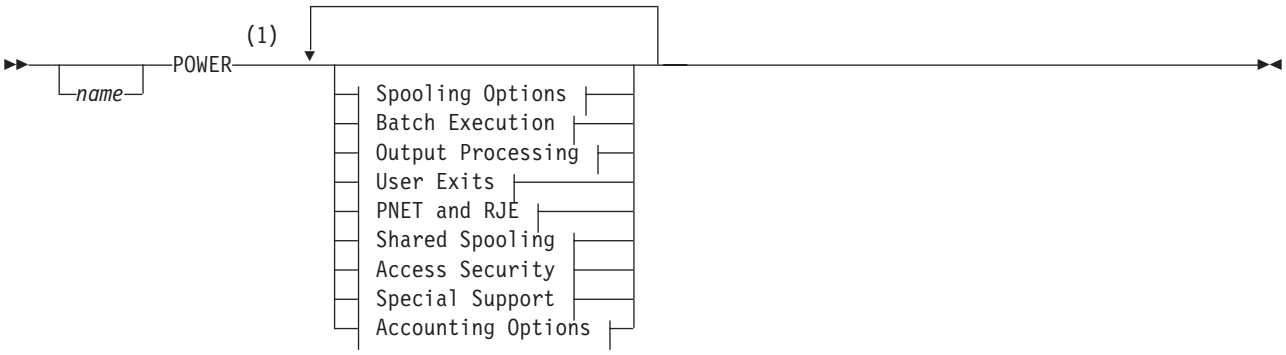
The functional support of a generated VSE/POWER phase can be modified and extended dynamically at system startup time. For details, see “SET: Setting VSE/POWER Startup Control Values” on page 473.

Skeleton **SKPWRGEN** provides a generation skeleton in VSE/ICCF library 59 for your convenience. It reflects the values which were used to generate the supplied VSE/POWER phase IPWPOWER. To change any of its values, copy the skeleton first into your VSE/ICCF primary library. Then edit the copied skeleton according to your needs. For values that differ in phase IPWPOWER in SKPWRGEN, see “Generating VSE/POWER Tables” on page 55.

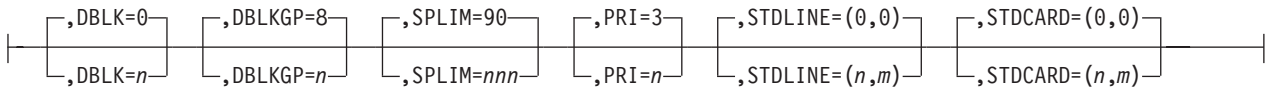
Format of the Generation Macro

The following format describes the generation macro. For ease-of-use, the operands are arranged according to functional groups. The description of the operands appears, however, in alphabetical order.

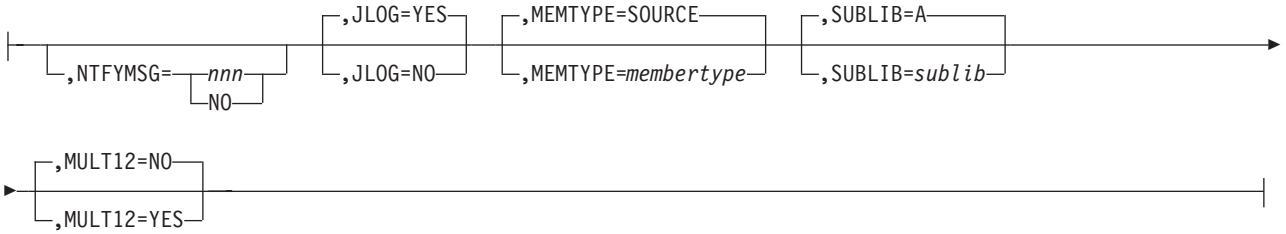
POWER Macro



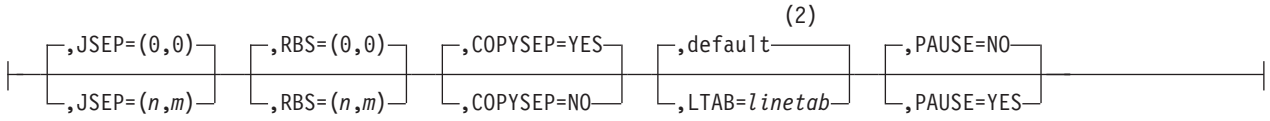
Spooling Options:



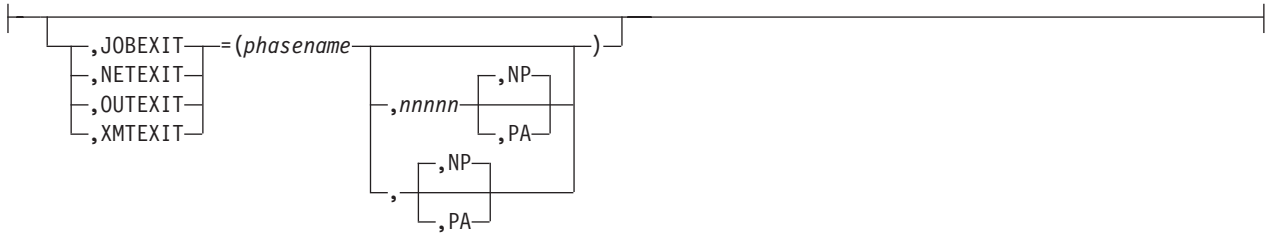
Batch Execution:



Output Processing:



User Exits:

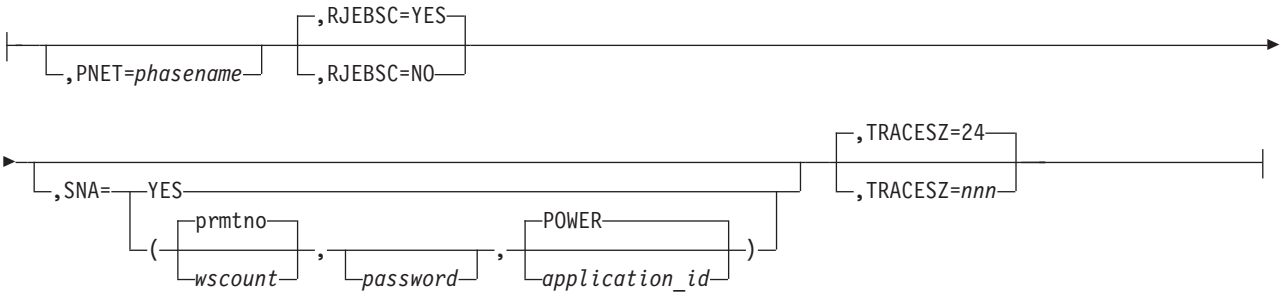


Notes:

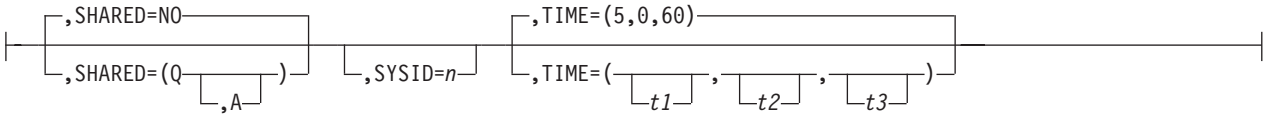
1 The macro name (POWER) must be separated from the *first* operand with a blank (not a comma!).

2 For the default values, see the operand description of LTAB.

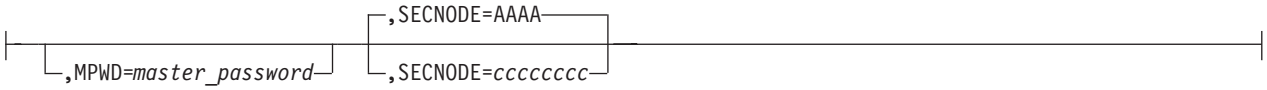
PNET and RJE:



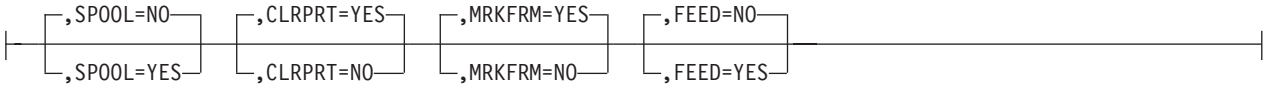
Shared Spooling:



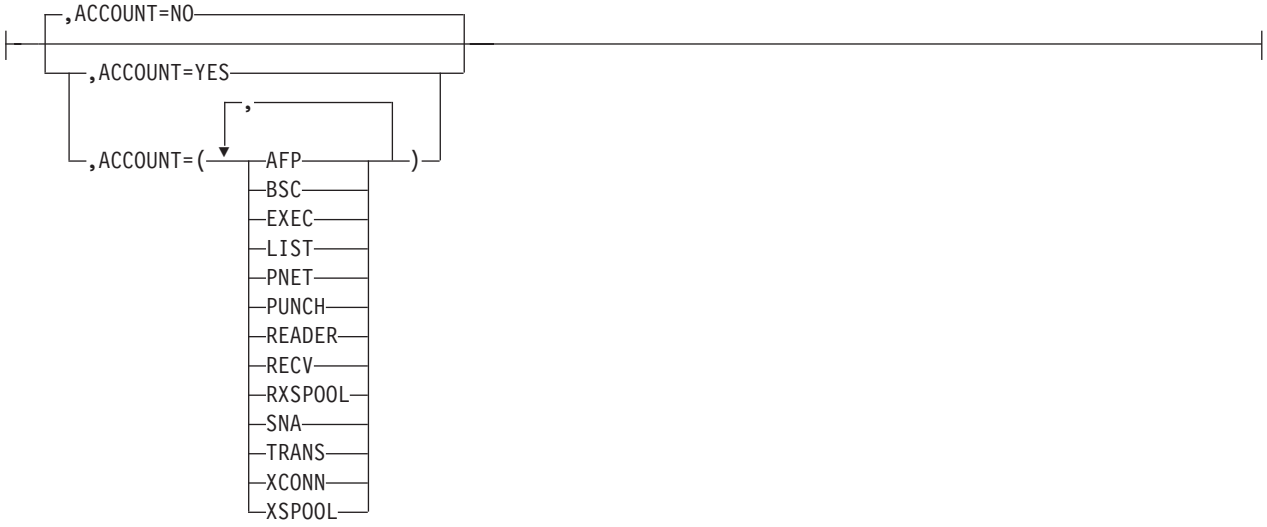
Access Security:



Special Support:



Accounting Options:



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.

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

You may limit the amount of accounting information collected by specifying an operand list. VSE/POWER writes only the account records specified in the list. This list may contain one or more of the following operands:

AFP
Advanced Function Printing account record

BSC
RJE,BSC account record

EXEC
Execution account record

LIST
List account record

PNET
Network account record

PUNCH
Punch account record

READER
Reader account record

RECV
Receiver account record

RXSPOOL
Reduced spool-access operation account record. The same account records are written as with the XSPOOL operand with the exception of account records coming from $SPLnnnn entries. These entries are temporarily built LST queue entries on request of a spool-access support CTL (PDISPLAY) request.

The operand overrides the XSPOOL operation.

SNA
RJE,SNA account record

TRANS
Transmitter account record

XCONN
Spool-access support connect account record

XSPOOL
Spool-access support operation account record.

The operand does not override the RXSPOOL specification.

Note: The ″system-up″ account record is produced unconditionally.

For more information on these ACCOUNT operands, see *VSE/POWER Application Programming*.

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 every 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

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 the JSEP option specification in the * $$ LST or * $$ PUN JECL statements.

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 every data buffer. Data buffers are used as I/O areas by the execution processors, reader and writer tasks, RJE and PNET tasks, as well as the Spool-access Support tasks. The DBLK size is the main parameter when you want to influence VSE/POWER's spooling performance.

The value for *n* must be a number from 1,000 to 65,024.

If you omit the operand or specify 0, VSE/POWER uses the default values as shown in Figure 5 on page 36.

- For the data file on a CKD disk, if you specify a value greater than the track capacity of the disk device, for example the maximum of 65,024, VSE/POWER warns you by message *1Q3BI* and continues using the maximum record size per track as DBLK value

- For the data file on an FBA disk, if you specify a value less than an interger multiple of an FBA block (512 bytes), some space of the last FBA block is wasted without a warning message

Whenever VSE/POWER performs a warm start, it uses the DBLK value that it finds already in the queue file. The operator is warned by message 1Q1CI if the existing and warm start DBLK values do not match.

For information on the DBLK size, refer to "The Size of the Data File" on page 36 and subsequent sections.

Note: When you increase the DBLK value (not the DBLKGP value), you should, at the same time, extend also the VSE/POWER:;

- SETPFIX-LIMIT area, because *one* pfixed I/O area of DBLK size is claimed in the SETPFIX area for *every* specified data file extent. See "Organization of the Data File" on page 28 for using more than one data file extent.
- Partition GETVIS area, because *every* task that does I/O to the data file claims an I/O area of DBLK size in the GETVIS area. Refer also to "Fixable- and Virtual-Storage Requirements" on page 44 for influencing VSE/POWER partition storage areas.

DBLKGP=8|n

Specifies the number of 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/output on the data file. When a job or output is bigger than the 'DBLK-value times DBLKGP size' number of bytes, then further DBLK groups are acquired and chained to the queue entry. The smallest DBLK group consists of two DBLKs, that is, the minimum value for *n* is 2. The maximum value that you can specify for *n* is 32,767. If you omit the operand, VSE/POWER sets the DBLKGP value to 8 DBLKs.

Whenever VSE/POWER performs a warm start, it uses the DBLKGP value that it finds already in the queue file. The operator is warned by message 1Q1FI if the existing and warm start DBLKGP values do not match.

For more information about DBLKGPs, refer to "The Size of the Data File" on page 36.

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 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.
- A VSE/POWER job with the LOG=NO specification in the * $$ JOB statement has completed processing.

The job-related data is:

Job name and number
Originating information

Time stamp

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

Specify JLOG=NO if no such logging is desired.

Independent of the JLOG specification, VSE/POWER records attributes of any currently running VSE/POWER job in the GETFLD FIELD=POWJOB area of the executing partition. For retrieval of this information refer to *VSE/POWER Application Programming*, chapter "Dynamic Access to VSE/POWER Job Attributes".

JOBEXIT=(phasename)|(phasename,nnnnn,NP|PA)
Specify this operand to load a user-written exit routine for processing local input into the pageable area of the VSE/POWER partition.

For *phasename* specify the name of the exit routine.

For *nnnnn* specify the number of bytes to be reserved as work area for the exit routine. Default is 'no work area'. The maximum is 65,535 bytes.

For *NP|PA* specify the type of workunit for which the exit should be given control:

NP as non-parallel workunit (default)

PA as parallel workunit. The PA option takes effect only when the multi-processor support of VSE/POWER has been enabled.

For details on processing workunits, refer to "Passing Control to VSE/POWER User Exits" on page 130.

For successful loading, the exit must not reside in the SVA.

For details on coding a JOBEXIT routine, see "Exit Routine for Local Input (Type JOBEXIT)" in *VSE/POWER Application Programming*.

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, both preceding and trailing a printed list queue entry.

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.

If you spool your output via the spool-access support and SPLONSEP contains X'40', VSE/POWER assigns the number of separator cards or pages defined in the JSEP operand. You can override this by specifying a number in SPLONSEP.

At printing time, 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, called 'last-one', than specified. Every separator page but the last-one contains information as described on "Separator Pages - Layout and Control" on page 181.

The last separator page contains only VSE/POWER jobname, jobnumber and sequence number.

If you do not want a different layout for the last separator page, use the SET ISEP autostart statement (page 482) or the ISEP/ISEPJ operand of the PSTART command (page 362 and 366).

If the page length defined for your output is less than 45 lines, VSE/POWER cannot print a job ID 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 = X'78' punches in all columns as start separator cards before the job's punched output:

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

- One card with the VSE/POWER job name punched such that it can be read when viewing the card from the back.
- Two blank cards as end separator cards.

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 terminal 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.

Every 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 specified d12 as 00, VSE/POWER uses the system-defined default page length (STDOPT LINES value) for d12. 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 regarding page size and channel positions. This ensures proper handling of output to this printer for which this default image is to be used. In general, for FCB controlled printers, use the FCB= operand of the * $$ LST statement to get the described page layout under all circumstances. See also the discussion of the operands FCB and LTAB under "* $$ LST: Defining List-Output Attributes" on page 426 and under "When Print Output is Misaligned" on page 116.

MEMTYPE=SOURCE|membertype

This operand applies to Source Code Inclusion (SLI) at job execution time. 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 uses A as default for sublib in the SLI statement (interpreted as membertype). 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.

MPWD=master-password

Specify this operand if VSE/POWER should use the specified password as the master password. The password of a spool-access support GET/CTL request is checked against this master password. If they match, unlimited access to the specified queue entry or for the specified command is allowed.

Specify 1 to 8 alphameric characters. If you do not specify this operand, access checking is done according to existing rules. For additional information, see *VSE/POWER Application Programming*, "Scope of GET/CTL Access to Queue Entries".

Note: After assembly of a POWER generation, ensure, for security reasons, that the MPWD value is deleted from the assembler source and listing.

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 user 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 every 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.

NETEXIT=(phasename) | (phasename,nnnnn,NP | PA)
Specify this operand to load a user-written exit routine for processing input from the network into the pageable area of the VSE/POWER partition.

For *phasename* specify the name of the exit routine.

For *nnnnn* specify the number of bytes VSE/POWER will reserve as work area for the exit routine. Default is 'no work area'. The maximum is 65,535 bytes.

For *NP | PA* specify the type of workunit for which the exit should be given control:

NP as non-parallel workunit (default)

PA as parallel workunit. The PA option takes effect only when the multi-processor support of VSE/POWER has been enabled.

For details on processing workunits, refer to "Passing Control to VSE/POWER User Exits" on page 130.

For successful loading, the exit must not reside in the SVA.

For details on coding a NETEXIT routine, see "PNET Receiver Exit" in *VSE/POWER Networking*.

NTFYMSG=nnn | NO
Specify *nnn* 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.

Specify NTFYMSG=NO, if you want any notify message for VSE/ICCF to be discarded without warning. This applies to both locally and remotely created messages.

If you omit this operand, the notify messages for VSE/ICCF will be discarded but a warning message to the central operator will be issued.

OUTEXIT=(phasename) | (phasename,nnnnn,NP | PA)
Specify this operand to load a user written exit routine for processing output into the pageable area of the VSE/POWER partition. This routine is given control for each record before it is printed.

For *phasename* specify the name of the exit routine.

For *nnnnn* specify the number of bytes VSE/POWER will reserve as work area for the exit routine. Default is 'no work area'. The maximum is 65,535 bytes.

For *NP | PA* specify the type of workunit for which the exit should be given control:

NP as non-parallel workunit (default)

PA as parallel workunit. The PA option takes effect only when the multi-processor support of VSE/POWER has been enabled.

For details on processing workunits, refer to "Passing Control to VSE/POWER User Exits" on page 130.

For successful loading, the exit must not reside in the SVA.

For details on coding an OUTEXIT routine, see "Exit Routine for Output (Type OUTEXIT)" in *VSE/POWER Application Programming*.

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/CTC/SNA/TCP/SSL) is to be supported. For phasename, insert

the name you assigned to the applicable set of PNODE macros, which means the phasename of the desired network definition table (NDT). One may even specify an NDT with a new local node name. For details see*VSE/POWER Networking*.

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 0 to 9, where 9 is the highest priority.

RBS=(0,0) | (n,m)
This operand applies only to output that is spooled to disk, that means output which is produced during execution of local batch jobs.

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 you can specify any number up to 999,999; for m you can specify any number up to 999,999. If you specify 0 or omit the operand, no record-count controlled segmentation takes place. Output segments may be identified by the same jobname and jobnumber but suffix numbers are 1 - 127.

If your job produces more than 127 segments, VSE/POWER assigns a new job number to each following group of 127 segments and starts with suffix 1 for each new job number.

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.

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

For this operand to take effect, submit at least one PLINE generation macro and one PRMT(BSC) generation macro.

SECNODE=cccccccc | AAAA
This operand specifies the VSE ″security zone″ for this system. A maximum value of 8 alphameric characters is allowed. The default value is ’AAAA’. The SECNODE operand is ignored if the z/VSE Access Control feature is not activated.

This operand identifies a CPU (or group of CPUs in the case that other systems have the same SECNODE value) on which an ″authenticated″ job may be executed with propagated security authorization from a parent job.

Such jobs will be recognized as ″authenticated″ even after:

- Transmission via a network to another system
- Submission via a shared DASD to another system
- Transport via a magnetic tape to another system

as long as the executing system shares the same SECNODE value.

Notes:

1. If the system is being warm-started, then care must be taken when specifying a different value than during the previous start-up. This will

overwrite the old SECNODE value and can cause already existing authenticated jobs to lose their VSE security authorization when they begin executing.

For further details, refer to section "Data Security Considerations" on page 11.

2. The user is responsible to insure that job security userids are uniquely specified in all DTSECTABs within the security zone, so that jobs do not unwittingly share secured resources without the proper authorization.

   For details on security and the concept of security zones, see *z/VSE Administration*.

SHARED=NO | (Q) | (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, application-id | 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 RJE 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: It sets this value to the total number of SNA RJE stations defined by PRMT macros or to 250, whichever is the smaller.
- If you specify a value smaller than the total number of RJE stations defined by PRMT generation macros, VSE/POWER uses the specified value. This can prevent a RJE station user from logging on to your system.
- If you specify a value greater than the number of RJE stations defined by PRMT generation macros, VSE/POWER sets the value equal to the number of defined RJE 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 VTAM ACB. This password must be identical to the one you specified in the PRTCT operand of the VTAM APPL statement. If you did not specify a password in the APPL statement, VSE/POWER ignores the password specified here.

application-id

For application-id, specify the name by which you defined VSE/POWER to VTAM as an application (when you set up the VTAM API). The default application ID for VSE/POWER is "POWER".

Do not specify the same name as specified for the APPLID operand in the PNODE macro used for the local node (LOCAL=YES).

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 takes 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 at output spooling time before VSE/POWER issues the warning message 1Q52I 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 999,999. STDCARD specifications may be overwritten dynamically by the RBM= operand of the * $$ PUN statement.

- A specification of 0 in the first entry and 0≤m≤999999 in the second entry means, that no warning message is given.
- A specification of 0<n≤999999 in the first entry but 0 in the second entry means that message 1Q52I is issued once when n is reached, but no further warning is created.
- 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 at output spooling time before VSE/POWER issues the warning message 1Q52I 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 999,999. STDLINE specifications may be overwritten dynamically by the RBM= operand of the * $$ LST statement.

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

SUBLIB=A | 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.

Note: The default SUBLIB=A may be overwritten by a one-character specification for the MEMTYPE operand.

SYSID=n

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

For n, specify the ID, a one-digit number from 1 to 9, of the system on which your VSE/POWER is to run. VSE/POWER uses this ID in all account records (within the 16 byte shared account-record prefix) and in its responses to PDISPLAY requests.

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

You can change the ID by SET SYSID=system-id 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 maximum active time in number of seconds. This is the maximum time your VSE/POWER is given update control of the spool files, provided many tasks need this mode. 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 it takes 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.

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:

```
TIME=(8,,45)
TIME=(2,3)
```

A further time interval "t4" is displayed in the statistics status report in Figure 7 on page 46. It defines how long the processor waits for completion of the VSE LOCK request before the operator gets message 1QB6I. "t4" can be specified, but there is no need to modify the default of 180 seconds.

TRACESZ=24 | nnn
The operand specifies the number of KB (up to 248) to be reserved for the telecommunication-trace function. This trace area is allocated in the Program Space of the VSE/POWER partition. The specified value is rounded up to the next 8 KB boundary. If you omit this operand but specify RJEBSC=YES or PNET=phasename, VSE/POWER reserves an area of the default size.

XMTEXIT=(phasename) | (phasename,nnnnn,NP | PA)
Specify this operand to load a user-written transmitter exit routine into the pageable area of the VSE/POWER partition. This routine is given control for each record (job or output) before it is transmitted. For *phasename* specify the name of the exit routine.

For *nnnnn* specify the number of bytes VSE/POWER will reserve as work area for every transmitter task that enters the transmitter exit. Default is 'no work area'. The maximum is 65,535 bytes.

For *NP | PA* specify the type of workunit for which the exit should be given control:

NP as non-parallel workunit (default)

PA as parallel workunit. The PA option takes effect only when the multi-processor support of VSE/POWER has been enabled.

For details on processing workunits, refer to "Passing Control to VSE/POWER User Exits" on page 130.

For successful loading, the exit must not reside in the SVA.

For details on coding an XMTEXIT routine, see "PNET Transmitter Exit" in *VSE/POWER Networking*.

PLINE Generation Macro

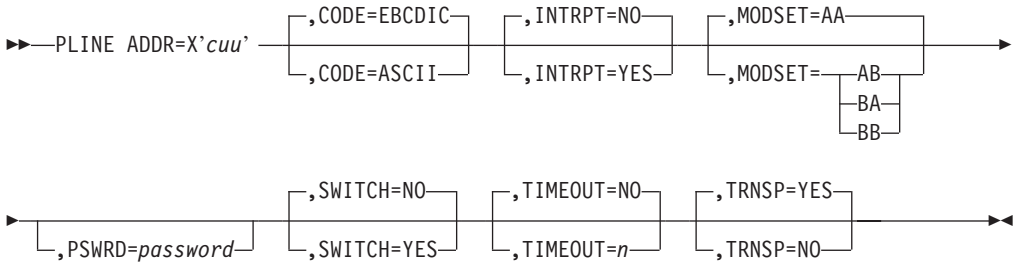
The macro and its operands define the hardware characteristics of a BSC line – a control unit such as an integrated communication adapter, a transmission control unit, or an IBM 37xx in emulation mode.

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 CTCA support under VM. One PLINE macro must be coded for every 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).

Skeleton **SKPWRBSC** provides a generation skeleton in VSE/ICCF library 59 for your convenience. To change any of its values, copy the skeleton first into your VSE/ICCF primary library. Then edit the copied skeleton according to your needs.

Format of the Macro



ADDR=X'cuu'
The operand specifies the line address (cuu) 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.

Note: PNET, BSC always uses the EBCDIC transmission code; ASCII specification does not take effect.

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 signon of a remote terminal 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 signon 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 terminal or line is allowed to remain idle before signoff is forced. The specification takes effect after signon is complete and when no job or output data is being transferred, which means VSE/POWER is in 'control mode'.

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

Notes:

1. After signon, when job or output data is being transferred, VSE/POWER forces signoff if the terminal contact has been lost (is timed-out) for 1,5 minutes.
2. Before signon, VSE/POWER disconnects if no signon 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 setup 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 signoff can save line charges on switched lines. It can save processor charges for a remote user because, without a forced signoff, the system processes timeouts every three seconds. Moreover, an idle terminal 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 timeout 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 signon is complete, VSE/POWER uses a timeout 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

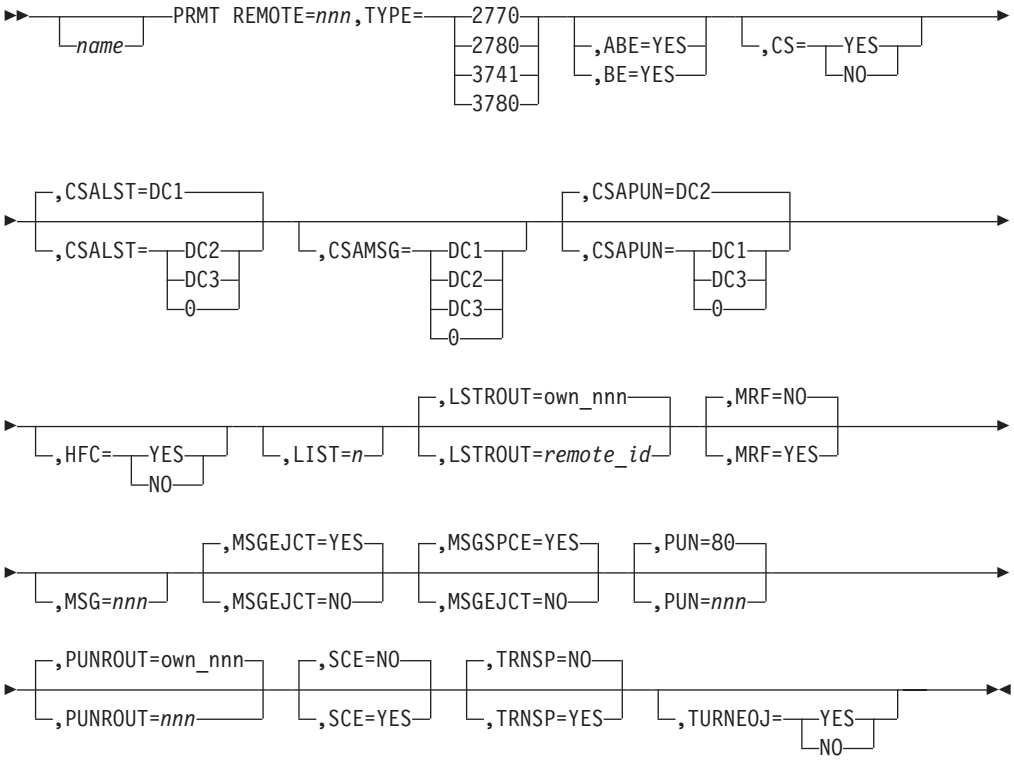
presented to the line (a way of transmitting data called transparency). The transmission of a language translator output, for example, requires transparency.

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.

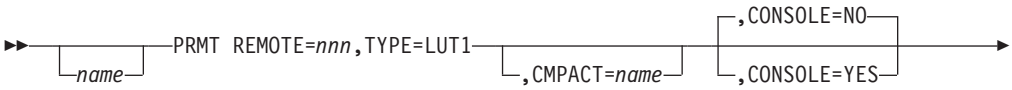
PRMT Generation Macro

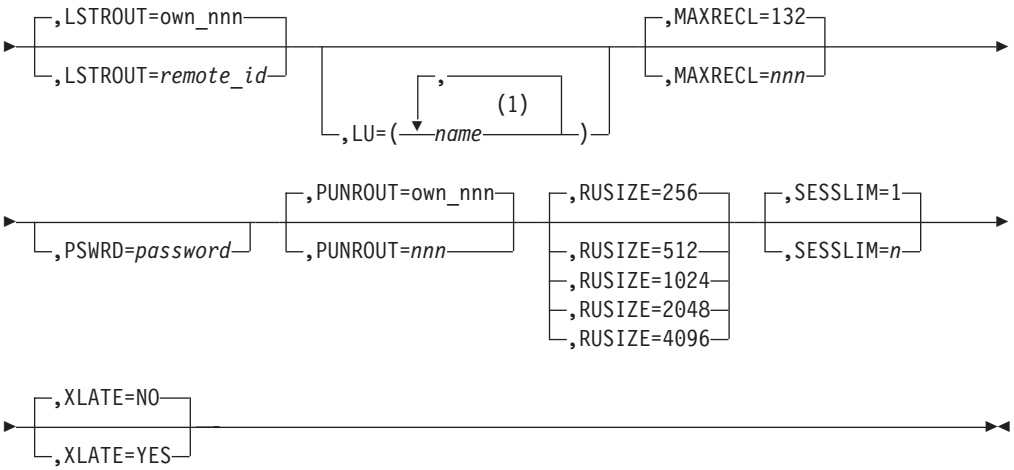
The macro defines the hardware characteristics of an RJE terminal. It identifies the user and describes where output is to be routed. At least one PRMT macro must be supplied for every remote terminal, and all PRMT macros must be submitted behind the PLINE macro(s), if present. The PRMT macros must be submitted in ascending order of the terminal IDs, and your definitions for RJE,BSC terminals **must** precede the ones for RJE,SNA terminals. Therefore, the lowest RJE,SNA ID must be higher than the highest RJE,BSC ID. An operand specification for a terminal without the specific hardware feature is ignored.

Format 1: Defining Hardware Characteristics for RJE,BSC Terminals



Format 2: Defining Hardware Characteristics for RJE,SNA Terminals



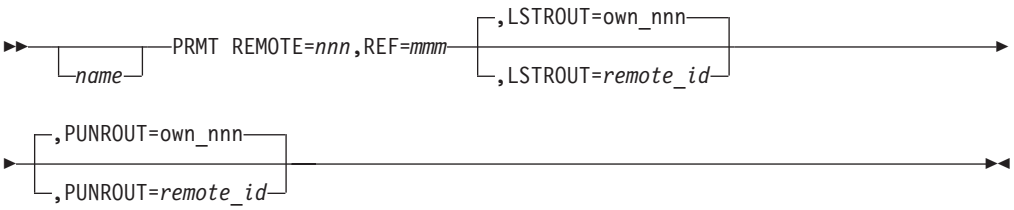


Notes:

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

Format 3: Short Form for RJE,BSC and RJE,SNA

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



REMOTE=nnn

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

Note: Consider the restrictions for terminal ID's described in the introduction to the PRMT macro.

REF=mmm

The operand, which applies only to the macro's short form, specifies the ID given in another PRMT macro for REMOTE=nnn. It indicates that this (short form) macro defines a terminal whose characteristics are the same as those defined by the referenced PRMT macro. The value for mmm 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 wrong.

TYPE=2770 | 2780 | 3741 | 3780
The operand specifies the type of terminal.

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

TYPE=LUT1
For an SNA terminal, 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 is overridden by a specification of ABE=YES.

CMPACT=name
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
Specify YES if the terminal has a line printer in addition to the normal terminal 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 terminal. A free session with the terminal 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 terminal does not have an additional line printer and you specify CONSOLE=YES, the printout at the terminal has interspersed operator messages.

CS=YES | NO
This operand applies to a BSC terminal 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 terminal 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 terminal.

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:
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 of 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 terminal. 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 terminal. For information about specifying a select character, refer to the discussion of the CS operand above.

HFC=YES|NO
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
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. The following table shows maximum numbers of print positions accepted and default numbers used by VSE/POWER:

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=remote-id | own-nnn
The operand tells VSE/POWER where to route by default the list output of jobs submitted at the terminal. If you omit this operand, VSE/POWER routes the output to this terminal.

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

If you use this operand, VSE/POWER rejects any LOGON request from the defined terminal 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 terminal may be larger than the number specified in the SESSLIM operand. This number is limited only by the VSE assembler restrictions.

MAXRECL=nnn | 132
This operand applies only to an SNA terminal and should only be specified if the terminal supports a print line length of more than 132 characters. It allows transmission of print records with a record length greater than 132 bytes.

For `nnn` specify the maximum length of the print records that are to be transmitted to this remote station. You can specify any value in the range of 132 to 512.

If you omit this operand, VSE/POWER uses the default of 132. Print records exceeding MAXRECL are truncated to the specified length. For details, see *VSE/POWER Remote Job Entry*.

MRF=YES | NO
This operand applies only to a BSC terminal 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
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
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 terminal.

MSGSPCE=YES | NO
MSGSPCE=NO suppresses the transmission of four contiguous Space-3 commands following the transmission of a message to the terminal.

PSWRD=password
The operand defines the password that has to be specified at the remote terminal 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 is reset to its default.

PUNROUT=remote-id | own-nnn
The operand tells VSE/POWER where to route by default the punch output of jobs submitted at the terminal. If you omit this operand, VSE/POWER routes the output to this terminal.

RUSIZE=256 | 512 | 1024 | 2048 | 4096
This operand applies only to an RJE,SNA terminal and should be specified only if the terminal supports a Request Unit (RU) size of more than 256 bytes. RUSIZE limits the Inbound and Outbound request unit sizes, which are negotiated between VSE/POWER and the remote workstation during its logon. The evaluated Inbound request unit size is the lesser of RUSIZE and the secondary request unit size in the Logmode table entry (BINSRUSZ) selected by the remote workstation. The evaluated outbound request unit size is the lesser of RUSIZE and the primary request unit size of the Logmode table entry (BINPRUSZ).

Note: VSE/POWER rejects, with 'BindReject', a Logmode table entry containing a primary and/or secondary request unit size other than those shown above.

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

TRNSP=YES | NO
This operand applies only to **punch output** destined for RJE,BSC terminals.

The operand specifies that the transparency feature is available at the terminal. 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 terminal. The specification results in all characters between X'00' and X'40' to be converted to:

```
Zeros  — If the receiving terminal is an IBM 3741.
Blanks — If the receiving terminal 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

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 terminal as shown:

Type of Terminal	A Default of
IBM 2770	YES
IBM 2780	YES
IBM 3741	NO
IBM 3780	YES

XLATE=YES | NO

The operand applies only to **print output**

Specify XLATE=YES to have characters X'00' through X'3F' converted to blanks. This may be required by certain printers attached to the terminal.

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

XLATE=NO specifies that no translation of characters takes place.

PNODE Generation Macro for Networking Support

Networking Support is initialized whenever the EXEC 'mypower' phase has been assembled with a POWER macro with PNET=ndt-name, which specifies a Network Definition Table (NDT). This must have been assembled under VSE/ESA 2.6 or later with a series of PNODE macro calls, one for each node of your environment.

The PNODE 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 *VSE/POWER Networking*.

Skeleton **SKPWRNDT** provides a generation skeleton in the system library (IJSYSRS.SYSLIB) for your convenience. To change any of its values, copy the skeleton first into your VSE/ICCF primary library. Then edit the copied skeleton according to your needs.

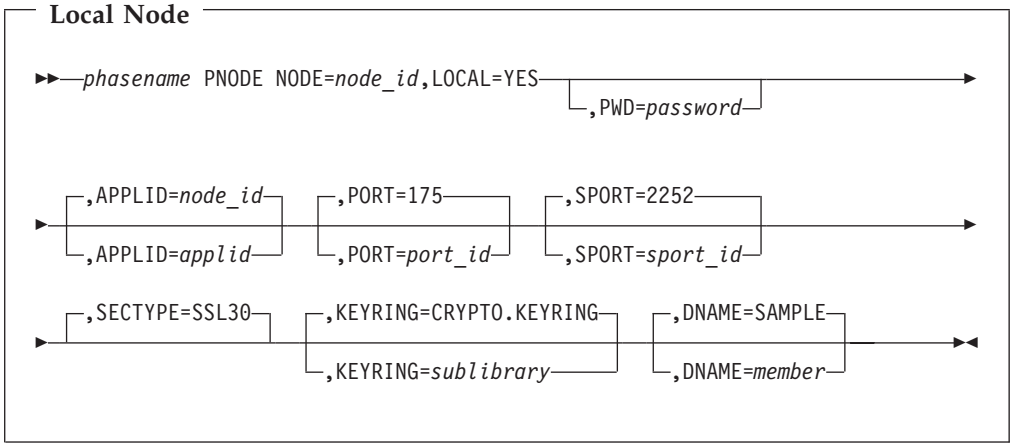
Format for Local Node

For the *local* node (see PNODE LOCAL=YES operand), other operands define what type of links to other (remote) nodes should be supported:

1. to other BSC nodes - no extra PNODE operand, but a 'cuu' address specified in the later PSTART other-node command, with 'cuu' added as 2703-type and defined additionally by a PLINE macro assembly
2. to other CTC nodes - no extra PNODE operand, but a 'cuu' address specified in the later PSTART other-node command, with 'cuu' added as CTCA-type
3. to other SNA nodes - with the optional PNODE APPLID= operand (default is 'node_id') to identify the own VSE/POWER application to VTAM
4. to other TCP/IP linked nodes - with the optional PNODE PORT= operand (default is 175) to define the local TCP/IP port number.
5. to other TCP/IP SSL linked nodes - with the optional PNODE SPORT= operand (default is 2252) to define the local secure TCP/IP port number. The optional SECTYPE=, KEYRING=, and DNAME= operands also apply to secure socket communications.

Note: The local host's IP address is acquired by an internal macro service. This address can be made visible using the PINQUIRE NODE=own-node command.

Format 1: Defining the Local Node



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 (all independent of LOCAL=YES or NO) 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 @.

Formats for Remote Nodes

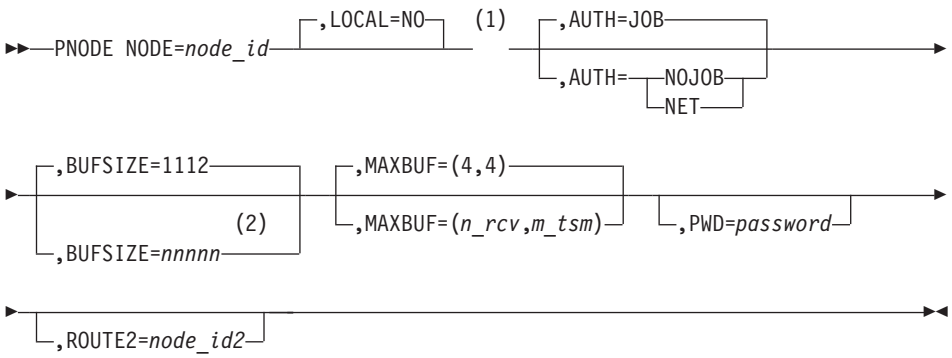
For the *remote* nodes (see PNODE LOCAL=NO operand), other operands define

- by what link type this node should be reached
- what attributes the communication should have (see PNODE AUTH=, PWD=, ROUTE2=)
- the size of the PNET buffer and how many should be used (see PNODE BUFSIZE=, MAXBUF=)

Format 2: Defining a Directly Linked BSC/CTC Node

BSC or CTC linked remote nodes need no extra PNODE operand for link identification, but the specification is done by the PSTART command.

BSC/CTC Node



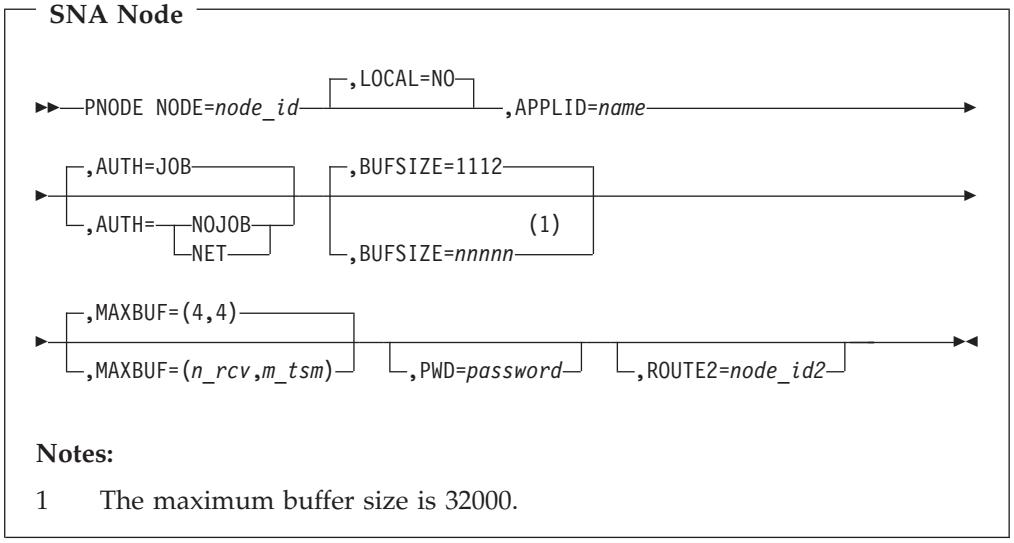
Notes:

1 The line address is specified with the PSTART PNET command. For a BSC node, this address refers to a PLINE macro entry assembled together with the POWER macro generation.

2 The maximum buffer size is 4000.

Format 3: Defining a Directly Linked SNA Node

SNA linked nodes need the APPLID= operand to identify the remote VTAM application.

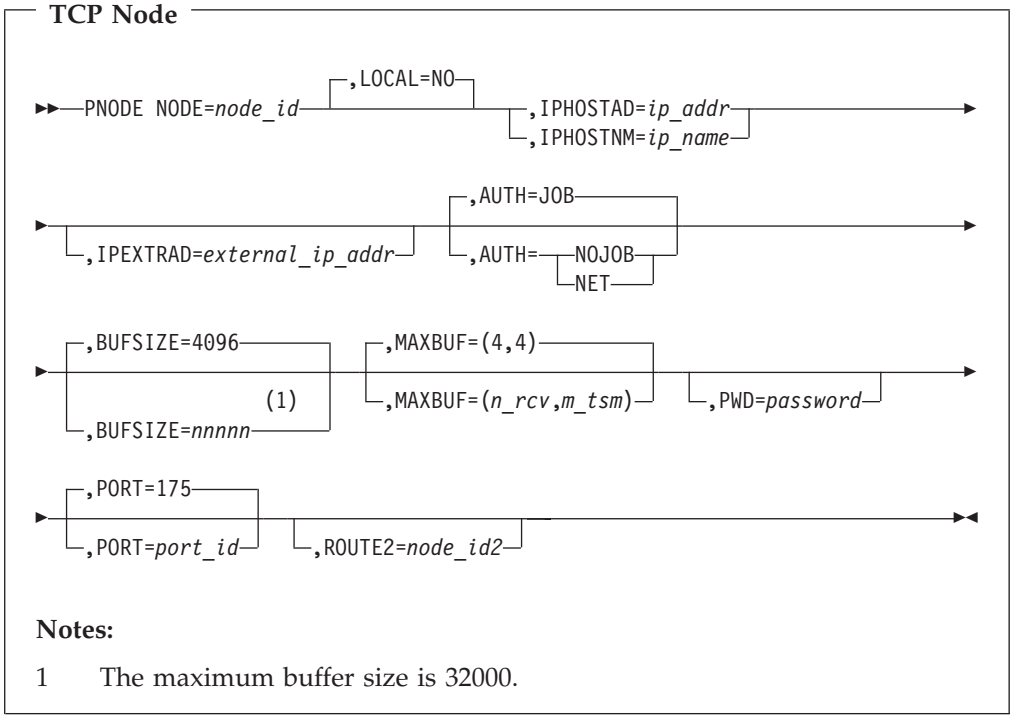


Notes:

1 The maximum buffer size is 32000.

Format 4: Defining a Directly Linked TCP Node

Nodes linked via TCP/IP need the IPHOSTAD= or IPHOSTNM= operand followed by the optional PORT= operand to identify the complete TCP/IP address of the counterpart VSE/POWER or RSCS node.



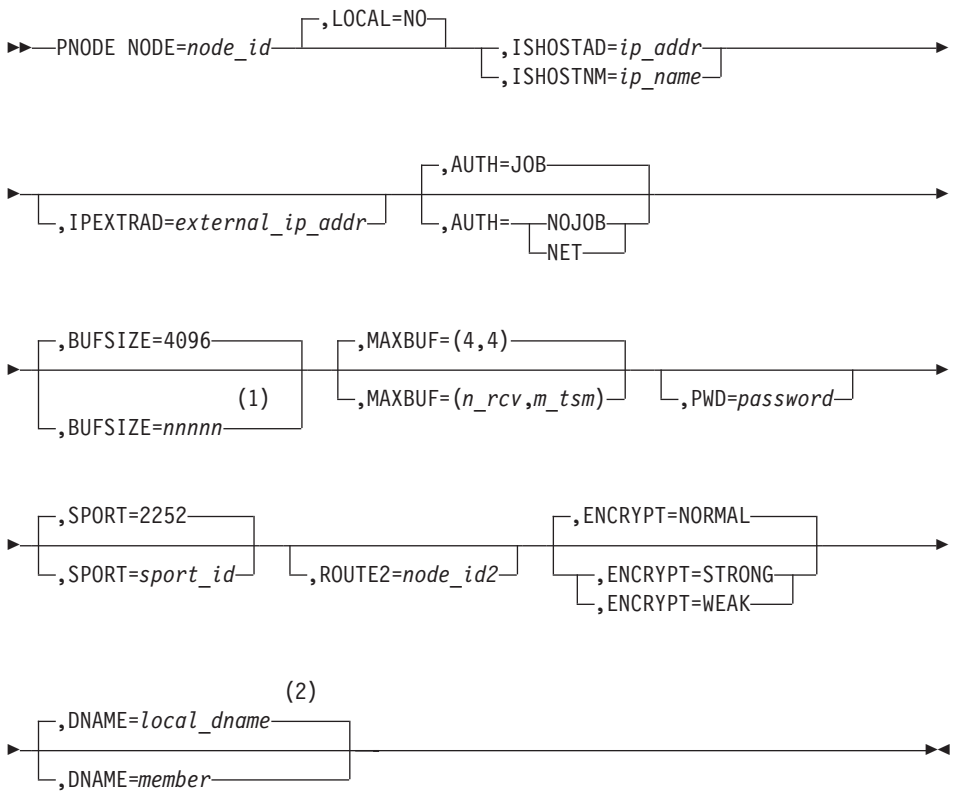
Notes:

1 The maximum buffer size is 32000.

Format 5: Defining a Directly Linked SSL Node

Nodes linked via TCP/IP SSL (Secure Sockets Layer) need the ISHOSTAD= or ISHOSTNM= operand followed by the optional SPORT= operand to identify the complete TCP/IP address of the counterpart VSE/POWER node. The optional ENCRYPT= and DNAME= operands apply also to secure socket communcations.

SSL Node



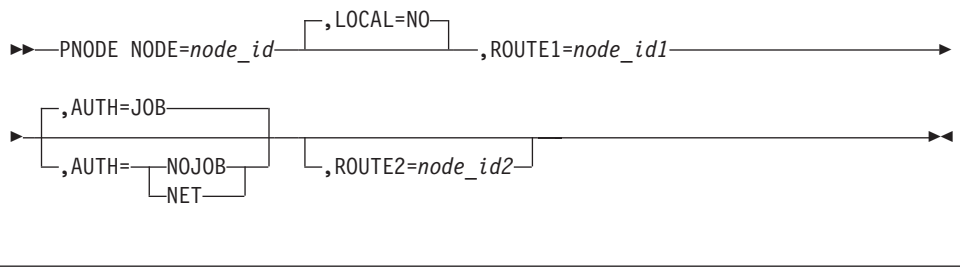
Notes:

1 The maximum buffer size is 32000.

2 The default value is the DNAME specified for the LOCAL=YES node.

Format 6: Defining an Indirectly Linked Node

Indirect Node



The following describes the operands selectable for the various formats:

NODE=node_id
For *node_id*, 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 (without /, . ,- as in assembler); 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 ID that is to be used to establish a 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 VTAM.

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

Notes:

1. If APPLID is not specified for the local node, VSE/POWER uses as application ID the name specified for node_id in the macro's NODE operand.
2. 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 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=node_id) 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 use general display functions.

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 *VSE/POWER Networking*.

BUFSIZE=nnnnn | 1112 (BSC/CTCA/SNA) | 4096 (TCP/SSL)
For *nnnnn*, specify (in number of bytes) the length of each of the VSE/POWER transmission buffers. You can specify a value from 300 up to:

4,000 bytes
For transmission via a BSC line or with CTCA support. These bytes are claimed from the SETPFIX-LIMIT (also called fixable) storage of the VSE/POWER partition.

32,000 bytes
For transmission via an SNA, TCP, or SSL link. The buffer bytes are claimed from the VSE/POWER partition GETVIS-24 storage.

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*.

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

Note: When increasing the BUFSIZE or MAXBUF values, you should extend at the same time the VSE/POWER SETPFIX or partition GETVIS-24 areas.

DNAME=member
If LOCAL=NO, Distinguished Name specifies up to 8 characters of a z/VSE sublibrary member which is used as an SSL key for this node. When the DNAME operand is omitted, the corresponding value specified or defaulted for the LOCAL=YES node is taken as the default for the SSL key.

If LOCAL=YES, Distinguished Name specifies up to 8 characters of a z/VSE sublibrary member which is used as an SSL key for this node. z/VSE distributes a member named "SAMPLE", which can be used for your demonstration purposes.

Specify your own member created according to *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*, SC33-6601.

ENCRYPT=NORMAL | STRONG | WEAK
Specifies the cipher information used for encryption. The operand selects the list of ciphers that are negotiated with the client to choose the SSL encryption technique, key size and message authentication code (MAC). Server and client agree upon the first match within the list. If using the TCP/IP for VSE/ESA host application distributed by IBM, the first suite value of the list is selected.

ENCRYPT=NORMAL
Causes use of exportable and domestic cipher information. Should be used when normal cryptographic functions are available, for example, within the U.S. and Canada. Following cipher suite value is used for negotiation:

```
09 for RSA1024_DESCBC_SHA
```

ENCRYPT=STRONG
Causes use of exportable and domestic cipher information. Should be used

when strong cryptographic functions are available, for example within the U.S. and Canada. Following cipher suite values are used for negotiation:

1. 0A for RSA1024_3DESCBC_SHA
2. 62 for RSA1024_EXPORT_DESCBC_SHA

ENCRYPT=WEAK
Causes use of only exportable cipher information. Should be used when only weak cryptographic functions are available, for example outside of the U.S. and Canada. Following cipher suite values are used for negotiation:

1. 08 for RSA512_DES40CBC_SHA
2. 01 for RSA512_NULL_MD5
3. 02 for RSA512_NULL_SHA

For details, see *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*, SC33-6601.

IPEXTRAD=external_ip_addr
Specifies the local host's dotted decimal IP address, by which the local node is known to the remote node.This operand is meaningful, if either:

- the local node is known to the remote NJE node via an external ip-address which is different from the local host's internal ip-address. The internal ip-address is the value specified in the TCP/IP statement SET IPADDR= and is to be used only within an internal network which is usually separated by a firewall from the external network, or
- within the local host an ip-address has been specified in the TCP/IP statement DEFINE LINK which is different from the ip-address specified in the TCP/IP statement SET IPADDR=

For further details on external_ip_address see also *VSE/POWER Networking*.

Note: for displaying the IPEXTRAD value in a Network Definition Table (NDT) refer to "Format 9: Displaying Network Definition Table Status (NDT)" on page 240

IPHOSTAD=ip_addr
Specifies the foreign host's dotted decimal IP address, by which the remote NJE node should be contacted. The dotted decimal format is, for example, 009.164.155.124 or 9.255.155.16, namely four groups (separated by '.') of 1 - 3 digit decimal addresses, each ranging from 0 to 255.

IPHOSTNM=ip_name
Specifies up to 255 characters of the fully qualified symbolic address of the foreign host, by which the remote NJE node should be contacted. This host name address will be resolved to an IP address using the 'TCP/IP for VSE/ESA, DEFINE NAME' support.

Note: If you specify a long symbolic address that exceeds the statement with the IPHOSTNM operand, use the existing Assembler continuation rules:

1. Place a continuation sign in column 72 (see "*" in the example).
2. Continue the operand value in column 16 of the next statement.

This coding is illustrated in the following sample generation for NDT254X, which contains a long symbolic address for the TCP/IP node POWER259:

```
// EXEC ASSEMBLY,SIZE=100K                                          24500010
NDT254X  PNODE NODE=POWER254,LOCAL=YES,PORT=175                     24500020
         PNODE NODE=POWER251,LOCAL=NO                               24500030
         PNODE NODE=POWER253,LOCAL=NO,PORT=175,IPHOSTAD=009.164.155.125 24500040
         PNODE NODE=POWER255,LOCAL=NO,APPLID=APPL255                24500050
         PNODE NODE=POWER256,ROUTE1=POWER255                        24500060
```

```
PNODE NODE=POWER259,LOCAL=NO,PORT=176,IPHOSTNM=ABCDEFGHIJKLMNO*24500070
      PQRSTUVWXYZ123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*24500080
      PQRSTUVWXYZ123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789*24500090
      PQRSTUVWXYZ123456789A,PWD=MYPWD                          24500100
END                                                            24500110
```

ISHOSTAD=ip_addr
Specifies the foreign host's dotted decimal IP address by which the remote NJE node should be contacted using the SSL feature. The dotted decimal format is, for example, 009.164.155.124 or 9.255.155.16, namely four groups (separated by '.') of 1 - 3 digit decimal addresses, each ranging from 0 to 255.

ISHOSTNM=ip_name
Specifies up to 255 characters of the fully qualified symbolic address of the foreign host by which the remote NJE node should be contacted using the SSL feature. This host name address will be resolved to an IP address using the 'TCP/IP for VSE/ESA, DEFINE NAME' support. See IPHOSTNM for coding rules.

KEYRING=sublibrary
Specifies up to 16 characters of the z/VSE sublibrary which is used as a key database. The members of this sublibrary serve as SSL keys. z/VSE distributes a sublibrary with the name "CRYPTO.KEYRING", which can be used for demonstration purposes. Specify your own library created according to *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*, SC33-6601.

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: If needed, refer to *VSE/POWER Networking* "Change Local Node Name During Warm Start".

MAXBUF=(n-rcv,m-tsm) | (4,4)
The operand specifies the number of buffers which may be used by every PNET-receiver and transmitter task.

For *n-rcv*, give the number of input buffers for each of the PNET-receiver tasks servicing the applicable other node. The minimum value is 3; the maximum value is 255 for BSC/CTCA/SNA remote nodes or 16 for TCP/SSL remote nodes.

For *m-tsm*, 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 BSC/CTCA/SNA remote nodes or 16 for TCP/SSL remote nodes..

Note: For TCP or SSL linked remote nodes, VSE/POWER acquires one additional buffer per started node.

For more information about these buffers, see *VSE/POWER Networking*.

PORT=175 | port_id

If LOCAL=NO, specify the 1-5 digit numeric port number of the remote NJE node on the foreign host to which PNET TCP should connect. The maximum accepted number is 32767. The default is the 'well known port' 175. port_id cannot be 2252 (see SPORT=).

If LOCAL=YES, specify the 1-5 digit numeric port number of the local TCP/IP host on which port VSE/POWER will listen for a connection request from remote hosts. The maximum accepted number is 32767. The default is the 'well known port' 175. port_id must be different from sport_id and cannot be 2252 (see SPORT=).

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). The password you specify can have up to eight alphameric characters. For more information about processing passwords, see *VSE/POWER Networking*.

ROUTE1=node-id1
The operand applies only to a node where no direct link exists to the node *node_id*. The operand specifies the name of the first node that is to be used for transmitting data or messages from your own node to your node *node_id*.

The node whose name you specify for *node-id1* must itself be defined by an entry in the NDT as a direct link (without the operand ROUTE1).

ROUTE2=node-id2
The operand specifies the name of a node to be used as a route alternative either to

- the direct link to the node *node_id* if the direct link cannot be established, or
- to the primary link specified in the ROUTE1 operand, 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 node-id2 must itself be defined by an entry in the NDT as a direct link (without the operand ROUTE1).

SECTYPE=type of security protocol
Specifies up to 8 characters for the security protocol used by the SSL feature. Currently, 'SSL30' for SSL Version 3.0 and 'TLS31' for TLS Version 3.1 are supported.

SPORT=2252 | sport_id
If LOCAL=NO, specifies the up to 5 digit numeric port number of the remote NJE node on the foreign host to which PNET SSL should connect using the SSL (Secure Sockets Layer) feature. The maximum accepted number is 32,767. The default is the 'registered port' 2252. sport_id cannot be 175.

If LOCAL=YES, specifies the up to 5 digit numeric port number of the local TCP/IP host on which VSE/POWER will listen for a connection request from remote hosts using the SSL (Secure Sockets Layer) feature. The maximum accepted number is 32,767. The default is the 'registered port' 2252. sport_id must be different from port_id and cannot be 175.

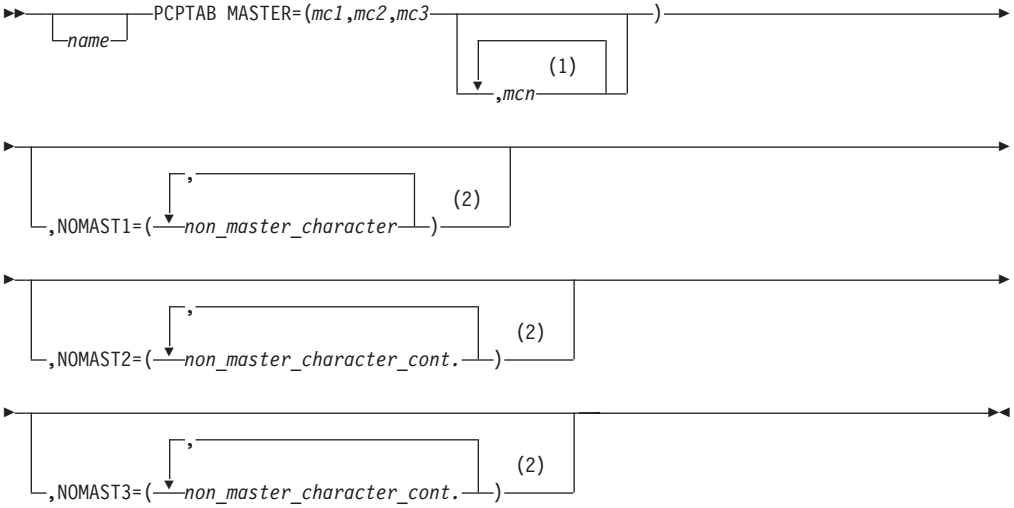
PCPTAB Generation Macro

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 every 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 C, “RJE, SNA Data Compaction,” on page 533.

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

Format of the Macro



Notes:

1 You can specify from 3 to 16 master characters in any sequence.

2 The number you specify depends on the number of master characters. For details refer to the description of the NOMAST1 operand.

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 is PCPT by default.

MASTER=(mc1,mc2,mc3) | (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, see Appendix C, "RJE, SNA Data Compaction," on page 533.

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 NOMASTn 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, see Appendix C, "RJE, SNA Data Compaction," on page 533.

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 be specified only 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 Table Generation

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

1. VSE/POWER only
2. VSE/POWER with both RJE,BSC and RJE,SNA support
3. VSE/POWER with RJE,SNA support only
4. VSE/POWER with PNET,BSC support only
5. VSE/POWER with PNET,BSC and SNA support
6. VSE/POWER with PNET,CTC and SNA, and TCP and SSL support

Note

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 terminals 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,BSC 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
        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
```

Example 5: VSE/POWER with PNET,BSC and SNA Support

```
NAME5   POWER  ACCOUNT=YES,      Accounting support
               DBLKGP=8,         DBLK group size
               SPLIM=80,         Spool limit value
               JSEP=(2,2),       Job separators
               PNET=NEWNDT5      PNET with NDT NEWNDT5
        PLINE  ADDR=X'cuu'       Defines BSC line to node C
        END

The macros below must be assembled and cataloged separately:

NEWNDT5 PNODE  NODE=NODEA,       The NDT entry for the own node
               LOCAL=YES,
               APPLID=INCA,      own VTAM application ID
               PWD=NODEAPW
        PNODE  NODE=NODEB,       The NDT entry for node B
               AUTH=JOB,         Command authorization
               BUFSIZE=800,      800-byte transmit buffers
               APPLID=TPNS       VTAM application ID of node B
        PNODE  NODE=NODEC,       The NDT entry for node C
               AUTH=NET,         Command authorization
               BUFSIZE=800,      800-byte transmit buffers
               PWD=NODECPW
        END
```

Example 6: VSE/POWER with PNET,CTC and SNA, and TCP and SSL Support

```
NAME6   POWER  ACCOUNT=YES,      Accounting support
               DBLKGP=8,         DBLK group size
               SPLIM=80,         Spool limit value
               JSEP=(2,2),       Job separators
               PNET=NDTBV01      PNET with NDT NDTBV01
        END

The macros below must be assembled and cataloged separately:

NDTBV01  PNODE NODE=NODEA,       NDT for the own node
               LOCAL=YES,
               APPLID=TPNS       own VTAM application ID
         PNODE NODE=BOEVS03,     NDT for SNA node BOEVS03
               LOCAL=NO,
               AUTH=NET,
               APPLID=ZURS,      VTAM application ID of node BOEVS03
               BUFSIZE=1500,
               PWD=DANYPW
         PNODE NODE=BOEVS04,     NDT for CTC node BOEVS04
               LOCAL=NO,
               AUTH=JOB,
               BUFSIZE=2500,
               ROUTE2=BOEVS03
         PNODE NODE=HURSL01,     NDT for indirect linked node HURSL01
               LOCAL=NO,
               AUTH=JOB,
               ROUTE1=BOEVS03    direct link BOEVS03, used for routing
         PNODE NODE=BOETP5,      NDT for TCP node BOETP5
               LOCAL=NO,
               AUTH=NET,
               IPHOSTAD=200.200.10.11,  TCP/IP addr. of foreign IP-host
               PORT=175          IP-port-# of BOETP5 at foreign IP-host
         PNODE NODE=BOETP6,      NDT for TCP node BOETP6
               LOCAL=NO,
               AUTH=NET,
               IPHOSTNM=BOE6.DE.IBM.COM,  symb. TCP/IP addr. of IP-host
               PORT=175          IP-port-# of BOETP6 at foreign IP-host
         PNODE NODE=BOETP7,      NDT for TCP node BOETP7
               LOCAL=NO,
               AUTH=NOJOB,
               BUFSIZE=8000,
               IPHOSTNM=BOETP7.NJECOMMUNICATIONS.DEUTSCHLAND.IBM.COM,
               PWD=BOET7PWD
         PNODE NODE=POWER262,    NDT for SSL node POWER262
               LOCAL=NO,
               AUTH=NET,
               BUFSIZE=32000,
               ISHOSTAD=009.164.155.137, TCP/IP SSL addr. foreign host
               SPORT=2252        secure IP-port# of POWER262 at IP-host
         END
```

Note: For a PDISPLAY of this assembled, cataloged, and loaded NDTBV01, refer to "Format 9: PDISPLAY PNET" on page 275.

Chapter 4. Operating with VSE/POWER

This chapter describes how to operate a VSE system with VSE/POWER from the console.

VSE/POWER Startup

You can request two types of startup for VSE/POWER: cold start (startup with spool file formatting) and warm start (startup without spool file formatting). When you should use each type of startup depends on why you need to start or restart VSE/POWER. Whether you start VSE/POWER warm or cold, you can start it choosing either of these methods:

1. Starting VSE/POWER automatically, that is 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 is the usual way.
2. Starting VSE/POWER interactively from the console.

This section describes both methods of starting VSE/POWER. For instructions on how to start VSE/POWER using the autostart procedure, see "How to Start VSE/POWER with Autostart (Cold and Warm Start)" on page 101. For instructions on how to start VSE/POWER interactively from the console, see "How to Start VSE/POWER Interactively (Warm and Cold Start)" on page 105.

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

Reference Material

To startup VSE/POWER you may use VSE commands, VSE/POWER commands, and autostart statements. For detailed information, reference the following manuals and chapters:

- For a detailed description of z/VSE system startup, see *z/VSE Operation*.
- For VSE commands, see *z/VSE System Control Statements*.
- For VSE/POWER central operator commands: see Chapter 5, "VSE/POWER Operator Commands," on page 191.
- For autostart statements, see Chapter 6, "JECL Statements," on page 397.

Startup Considerations

Consider the following when starting VSE/POWER:

Defining the Devices to be Spooled in Every Static 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 card 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 system without card device normally requires dummy device assignments.

Device Assignments for the VSE/POWER Partition

Reserve the logical units SYS000 through SYS033 for the VSE/POWER files by including assignments for them in your VSE/POWER startup procedure.

Shared Spooling: Startup 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 must synchronize their system clocks to the nearest minute for control purposes and to the same date format for creation of queue entries (see "Functions Not Expecting Date Format Changes" on page 128).

Starting of a sharing system may be delayed because:

- Another sharing system is 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 startup 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. For additional details, see "Example of a Spool File Recovery with Shared Spooling" on page 100.

To verify on a running system which shared spooling generation values have taken effect, use the following commands:

- PDISPLAY T, to see the own SYSID value
- PDISPLAY STATUS, to see SYSID(s) and interval TIME values

Starting VSE/POWER in Private or Shared Address Space

The Enterprise Systems Architecture/390 (ESA/390) allows any address space to access data in any other address space through access registers. VSE/POWER makes use of these access registers and can, therefore, be allocated in a private address space.

In order to move F1 with VSE/POWER into the shared address space, the z/VSE default allocations for F1 must be changed as follows:

1. Procedure ALLOC contains the effective ALLOC statements of F1. You can tailor this procedure using the skeletons SKALLOCn (n=A,B,C) in the VSE/ICCF library 59 and switch the statement ALLOC F1= to ALLOC S,F1=.
2. Extend the SYS statement of your effective IPL procedure by the SPSIZE= operand in order to reserve storage area for the shared partition F1.
3. Refer to "Organization of the Queue File" on page 26 and "Virtual Storage (VIO Space)" on page 48 and take into account that VSE/POWER starting in shared address space will attempt to place the queue file into VIO space instead of the partition Getivs area. Use the PDISPLAY STATUS command and find out the amount of KB required for the queue file in storage. When placed into VIO, the total ALLOC amount of the VSE/POWER partition may be reduced by the KB required for the queue file.

Restriction: If the VSE/POWER partition is allocated in a private address space, the following has to be considered:

- A user program that communicates with VSE/POWER using the SPOOL macro support (XECBTAB, XPOST and XWAIT macros) must run in a partition which is allocated in the same address space as the VSE/POWER partition. If this is not the case, error codes X'C0' is returned to the user program in register 15.

  For further details, please refer to *VSE/POWER Application Programming*.

Cold Start

Use a cold start when formatting of the queue, data, or account file is required. Usually, this is done when:

- VSE/POWER is started for the very first time
- The extents for the VSE/POWER spool files are changed between executions of VSE/POWER.

  Note: At warm start, the queue file extent can re-allocated and physically moved (see "Re-Allocate Queue File During Warm Start" on page 41) . Also at warm start, the data file size can be increased through the addition of another extent (see "Extending the Data File During Warm Start" on page 38), where the existing job queues are preserved.
- A new version or release of VSE/POWER is installed and your existing queue file(s) version is prior to version 6.7
- All types of warm starts are unsuccessful.

During a cold start, VSE/POWER removes all entries from the RDR, LST, PUN, or XMT queue. Thus, you must resubmit all jobs after a cold start. Because all jobs and output data in the system are lost when you perform a cold start, make use of the POFFLOAD command to save entries before a cold start and reload them after formatting has been done.

Shared Spooling Considerations: 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 startup is canceled; if the response is NO, the startup 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.

Warm Start

Use a warm start when total formatting of the queue, data, or account file is *not* required, that is when information existing in the VSE/POWER files can be used as is. Usually, this is done:

- After an orderly shutdown
- After VSE/POWER terminates abnormally due to an equipment or system failure.
- When a new version/release of VSE/POWER is installed and your existing queue/data file(s) are on version 6.7 or on later version/release.

Also use a warm start (FORMAT=NO) to extend the data file (see "Extending the Data File During Warm Start" on page 38) or to re-allocate the queue file (see"Re-Allocate Queue File During Warm Start" on page 41). In this case, job queues are preserved, and formatting takes place concurrently with spooling.

If you perform a warm start of VSE/POWER, the next jobnumber assigned to any new job or output entry is one higher than on termination. This prevents duplicate job numbers for accounting purposes.

Shared Spooling Considerations: Warm Start

Whenever a sharing system is 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.

When the addressed spool files have been or are used by a non-shared system, VSE/POWER prompts the operator by messages 1QB3A and 1QB3D. Then you are asked to specify if the startup means a deliberate switch from non-shared to shared operation with no other system operating, or if the startup is an inadvertent request with another non-shared system still currently accessing the spool files. If the latter is true, the startup is canceled for the sake of the integrity of the spool files.

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

To avoid inadvertent start of sharing systems with the same SYSID value, VSE/POWER records both CPU-ID and SYSID of operating sharing systems. That allows to distinguish between a correct shared restart and an incorrect startup of the same SYSID. The latter constellation is reported by message *1QAFI* and the operator is prompted for confirmation by message *1QAFD*.

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.

Example of a Spool File Recovery with Shared Spooling: The following example illustrates the spool file recovery concept. 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 startup 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 start over again after System B is started. Any jobs being printed or read in on System B are printed or read in from their beginning after System B is started.
3. Startup 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.

How to Start VSE/POWER with Autostart (Cold and Warm Start)

This section describes how to start VSE/POWER during startup 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 be cataloged in the system sublibrary. In the shipped z/VSE system, the standard JCL ASI procedure for all partitions is already contained in the IJSYSRS.SYSLIB.

The manual *z/VSE Guide to System Functions* describes how to write and catalog your own ASI JCL procedure set. The statements needed for a VSE/POWER startup procedure are discussed below.

Step 1: Allocating virtual storage for VSE/POWER in a private partition
Assume that VSE/POWER is to run in partition F1. Your allocation for the partition provided in the BG ASI procedure might look like this:

```
ALLOC F1=1600K
```

Step 2: Allocating fixable storage for VSE/POWER
Instead of defining an ALLOCR value, it is recommended to use the SETPFIX LIMIT= statement, which should be part of Step 5 ″Invoking VSE/POWER″.

Step 3: Starting the VSE/POWER partition
To start VSE/POWER in partition F1, include in the ASI JCL procedure for BG the following VSE commands:

```
START F1
STOP
```

This causes partition F1 to become active and partition BG to be stopped; it causes the ASI JCL procedure for partition F1 − the actual VSE/POWER startup statements − to be called.

Step 4: Making 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 startup is a warm start, the assignment causes VSE/POWER to provide a status report immediately after completion of the startup procedure. This report is similar to the one produced after a PEND command or after a PDISPLAY STATUS command. Figure 7 on page 46 gives an example of a PDISPLAY STATUS report.

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

Step 5: Defining VSE/POWER files and invoking VSE/POWER

Figure 10 shows an example of the VSE job control statements required for this purpose. The numbers in front of the statements refer to the explanations following the example.

For the standard z/VSE system shipped, the assignments for the spool files are contained in the DTRPOWR procedure in sublibrary IJSYSRS.SYSLIB. The label and extent information for the spool file are contained in the STDLABEL procedure in the same sublibrary.

```
1   // ASSGN SYS000,260
2   // DLBL IJAFILE,'VSE.POWER.ACCOUNT.FILE',99/365,DA
2   // EXTENT SYS000,POWER1,1,0,15,75
1   // ASSGN SYS001,260
2   // DLBL IJQFILE,'VSE.POWER.QUEUE.FILE',99/365,DA
2   // EXTENT SYS001,POWER1,1,0,75,30
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   // SETPFIX LIMIT=120K,PERM
5   // EXEC POWER,SIZE=960K
```

Figure 10. Job Stream Example for Starting IBM VSE/POWER

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 32 extents can be specified if the entire data file does not fit into one extent. Performance is slightly improved if you use additional extents.

For a definition of multiple data file extents, refer to Figure 1 on page 29.

4 120KB of the partition are reserved for VSE/POWER's real storage management that claims and frees fixable storage on demand for internal control blocks.

5 The example assumes that VSE/POWER runs in a 1600KB partition. The specification SIZE=960K causes 640KB of the partition to be reserved as Getvis space. For a precise determination of the SIZE value refer to "Fixable- and Virtual-Storage Requirements" on page 44.

If you address VTAM applications in your POWER startup (PNET,SNA or RJE,SNA), replace the sample EXEC statement by

```
// EXEC POWER,SIZE=960K,DSPACE=2M
```

Assume you will make use of the RJE,SNA support. Then, ACF/VTAM will define a data space with an initital size of 1MB to share with VSE/POWER after starting RJE,SNA via 'PSTART RJE,SNA'. This data space might be extended by ACF/VTAM to the maximum size of 2MB.

Starting PNET,SNA, then ACF/VTAM uses the same data space. For details on data space, refer to the "SYSDEF statement" in *z/VSE System Control Statements* and "Introducing Data Spaces" in *VSE/ESA Extended Addressability*.

If you omit a DSPACE value in the VSE/POWER startup, ACF/VTAM will define the data spaces with a maximum size of 1MB. This might be insufficient for transmission of data.

Step 6: Supplying the VSE/POWER autostart statements

These statements are described in Chapter 7, "VSE/POWER Autostart Statements," on page 471.

For the standard z/VSE system shipped, please refer to skeleton "SKPWSTRT" described in manual *z/VSE Administration*.

An example of a set of autostart statements is shown and discussed in Figure 11. The numbers in front of the statements refer to the explanations following the example.

```
1  SET SYSID=2
2  SET PNET=NETTBLE3
3  DEFINE L,PAGEDEF,1F,1,6,C
   DEFINE L,FORMDEF,1D,1,6,C
4  FORMAT=NO
5  PSTART BG,GST
   READER=00C
   PRINTERS=00E
   PUNCHES=00D
   PSTART F3,CEGF
   READER=00C
   PRINTERS=01E
   PUNCHES=00D
6  PSTART LST,00E
   PSTART LST,01E
   PSTART PUN,00D
   PSTART RJE,060
   PSTART RDR,00C
7  PSTART PNET,node-id,...
8  PLOAD DYNC
9  PSTART TASKTR,ENAB,24
   /+
```

Figure 11. Example of an IBM VSE/POWER Autostart Procedure

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, it is flagged by message *1Q13I*. You have to correct your procedure; the error cannot be corrected at the console. Use the PDISPLAY AUSTMT command to recollect all accepted and erroneous autostart statements processed at VSE/POWER start-up time.

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=Q 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.

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 static 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 startup without autostart.

In a running VSE/POWER system, use the PDISPLAY SPDEV command as described on page 245 to list the sequence and attributes of spool(ed) devices of active partitions.

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.

7 Your procedure may include PSTART PNET,node-id,... commands for both BSC and SNA nodes. It may include the required PACT commands; however, a PACT command may not precede its corresponding PSTART PNET command.

The first PSTART PNET command for an SNA node causes VSE/POWER to attempt linking to VTAM. If VTAM is not active, VSE/POWER queues the PSTART commands for SNA nodes until VTAM becomes active.

8 Your procedure may also include a PLOAD DYNC command to establish the active Dynamic Class Table DTR$DYNC.Z and to enable classes for scheduling of dynamic partitions.

9 This command reserves an in-core task trace area of 24 KB and causes the VSE/POWER task dispatcher to initiate recording of the task status whenever a task is given control. With such a serviceability provision you may aid in debugging of severe system failures.

How to Start VSE/POWER Interactively (Warm and Cold Start)

In rare cases, the operator at the system's console may have to startup 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, 01E, or 118
Disk storage device with device address 133, 134, or 260
Terminal device connected to a communication controller with device address 060

The disk devices are used for the VSE/POWER files; the terminal for remote job entry.

Step 1: Making 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: Setting 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 F1=1024K

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 186.

Step 3: Starting the VSE/POWER Partition

Start the VSE/POWER partition (F1 in the example) by issuing the command:

```
START F1
```

Step 4: Assigning 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: Preparing for a Status Report
Assign SYSLST if your installation requires a status report after a warm start. Figure 7 on page 46 shows an example of such a status report.

Step 6: Defining the VSE/POWER Files and Invoking 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 102.

If you did not assign SYSRDR, you have to enter the statements at the console, one after the other.

Step 7: Formatting VSE/POWER Queues
When VSE/POWER displays the message

```
1Q11D FORMAT QUEUES=
```

respond as instructed by your system administrator (see "FORMAT: Specifying File-Formatting Options" on page 492). Your response (behind the equal sign) can be one of the following:

Q Formats the queue file and the data file.

A Formats the account file.

Q,A Formats all VSE/POWER files (queue file, data file, account file)

NO No total formatting of the queue, data, or account file is required. Instead of entering NO, you may simply press END/ENTER. For VSE/POWER, this response is a "warm start" indication, which may also be used for extending the data file (see "Extending the Data File During Warm Start" on page 38). This approach preserves the existing job queues and formats only an additional data file extent. If you assigned SYSLST (in step 5), you get a status report.

When VSE/POWER startup is complete, you get the message

```
1Q12I VSE/POWER 7.1.0 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: Starting VSE/POWER Tasks
Use the PSTART command to place static partitions under control of VSE/POWER as required and to start VSE/POWER tasks as are needed.

Placing a Static Partition Under the Control of VSE/POWER

For example, the command:

```
PSTART F2,2A
```

places the partition F2 under control of VSE/POWER.

In response to the command, VSE/POWER prompts you to specify the devices that are to be spooled for the partition. The prompting messages for the above example are:

```
1R86I PLEASE SPECIFY DEVICES TO BE SPOOLED
1R50D F2 READER=
1R50D F2 PRINTERS=
1R50D F2 PUNCHES=
```

SYSLOG stops after every prompting message (1R50D). Thus you can specify a device address (in the form cuu) or enter NO, if one of the device types is not used, as shown in this example:

```
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 'writer-only' partition that is started. For the special handling of JECL statements in writer-only partitions, see also "JECL Influence on Writer-Only Partitions" on page 406.

VSE/POWER spools only the input if your response is READER=cuu, PRINTERS=NO and PUNCHES=NO and starts a 'reader-only' partition then.

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 System Control Commands with VSE/POWER

The following 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 has no effect on the FCB of the printer which is used for output from the LST task.

- The STOP command

  If STOP is used for a partition controlled by VSE/POWER, a following PSTOP or PEND command does not work until the partition has been restarted.

- The PRTY command

  The command can give to a VSE/POWER controlled partition only a priority which is below that of the VSE/POWER partition, unless the controlled partition has been started (PSTART) with the NPC parameter. There is, however, no restriction for giving a dynamic class any priority with respect to the priority of the VSE/POWER partition.

- The SYSBUFLD program

  The SYSBUFLD program should not be used 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.

Interaction with z/VSE Conditional Job Control Language

If the processing of VSE job steps contained in a VSE/POWER job is controlled by z/VSE conditional job control statements, the following rules apply at job processing time:

- While z/VSE Job Control is skipping statements in search for a GOTO label or when skipping the IF THEN statement, all intercepted VSE/POWER JECL statements contained in the area to be skipped are handled as follows:
  - * $$ LST/PUN/FLS/RDR/CTL/JOB/EOJ statements are ignored.

    For details on when an * $$ JOB/EOJ/CTL/RDR statement is processed at execution time, see "Handling Read-in JECL Statements at Execution Time" on page 403.
  - All other JECL statements take effect (that is, * $$ SLI including * $$ DATA).

  Note: You may request VSE/POWER to also suppress resolution of * $$ SLI (including * $$ DATA) statements by specifying the following Job Control option per VSE job

  ```
  // OPTION SLISKIP
  ```

- When an operator uses the PFLUSH or PCANCEL command to immediately terminate the processing of a VSE/POWER job, usually no further statements still contained in this job take effect. However, if an ON $CANCEL GOTO user-label routine has been established, VSE/POWER will pass all remaining statements of the VSE/POWER job to Job Control. Thus the $CANCEL routine statements may be located and interpreted.

  For such jobs, it remains the user's responsibility to determine what actions should be taken in the $CANCEL routine. If the whole VSE/POWER job should be terminated (in case more than one z/VSE job is included), the * $$ FLS JECL

statement may be specified in the $CANCEL routine. This results in unconditional termination of the whole VSE/POWER job.

Note: A $CANCEL routine is *not* recognized in the following cases:

- The GOTO user-label is the default label $EOJ.
- For VSE/POWER jobs, whose spooling of output fails due to data file I/O error resulting in loss or enforced segmentation of the output. In this case, conditional job control processing is informed by the X'01' setting of the POWFLG1 in the partition communication region of the serviced partition. This flag setting may be tested from within a $JOBEXIT user routine, to modify, for example, the corresponding label of a currently processed '// GOTO' job control statement.

- When due to a program cancellation or AR CANCEL command or PFLUSH/PCANCEL command, a $CANCEL routine has been reached and when then no * $$ FLS statement (JECL) is used but a PFLUSH/PCANCEL command is issued, cancellation takes place and all remaining statements of the VSE/POWER job are passed to Job Control because of the ON $CANCEL condition which remains in effect up to end–of–job. Suppression of the remaining statements can only be achieved when the $CANCEL routine establishes the ON $CANCEL GOTO $EOJ label before the PFLUSH/PCANCEL command can be issued.
- When processing of a VSE/POWER job is terminated by internal 'VSE/POWER triggered' flushing (e.g., in combination with messages 1R33D, 1QC0I-1QC4I, 1Q4FI, 1QC6I, 1Q4EI, or 1Q45I), no further statements still contained in this job take effect. Instead, after cancellation of the spooled partition, VSE/POWER simulates end-of-job and passes a '/&' statement to Job Control's read request and continues with the next job in the VSE/POWER reader queue.

However, with the SET INTFLUSH=OPER autostart statement, you can request VSE/POWER to handle internal flushing in the same way as operator flushing and to pass the remaining statements of the flushed job to Job Control, provided a $CANCEL routine has been established.

Using VSE/POWER Commands

The processing of jobs and the retrieval of output from these jobs is controlled by VSE/POWER commands in one of three ways:

1. At a console by an operator

   For a description of these commands, see Chapter 5, "VSE/POWER Operator Commands," on page 191.

2. At a remote terminal by the operator

   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 terminals, see *VSE/POWER Remote Job Entry*.

3. 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 in "CTL - Passing a Command" in *VSE/POWER Application Programming*.

The remainder of this section discusses operations at a console. It includes sections on the operation with miscellaneous devices and on the operation with tapes.

In using the VSE/POWER operator commands at a 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, every 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 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, if possible, also
- The number that VSE/POWER assigned to it. A number is required, for example, if several VSE/POWER jobs in a queue were logged under identical names.

If you are in doubt about the correct jobname and jobnumber of a VSE/POWER job, use the PDISPLAY command to have VSE/POWER display this. For details, see "PDISPLAY: Displaying VSE/POWER Status" on page 224.

This section discusses 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 a console.

Changing Partition Priorities

To place a partition under control of VSE/POWER, its processing priority normally must be lower than that of the VSE/POWER partition. This is especially true for a spooled batch partition. For cases where another partition may have a higher priority than the VSE/POWER partition, see "System Considerations" on page 24. To change priority, use the PRTY command described in *z/VSE 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.

Changing the Class Specification for a Static Partition

When placing a static 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 static partition.

Stopping and Restarting a Static 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 static 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 also are 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 114.
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 startup procedure on page 106.

Loading an FCB and a UCB under VSE/POWER

While spooling output, VSE/POWER knows about the current print position on the printer page using the information contained in the FCB image which is actually used for printing. To improve output spooling performance, VSE/POWER keeps up to 30 FCB images, each converted to a line table (LTAB) presentation in its own storage table. Each time an FCB is requested by an * $$ LST statement, the table is scanned for an entry with a matching FCB name. If the entry is found, its associated LTAB presentation is used when spooling the output entry. Otherwise, the relevant FCB image is loaded from the library, converted to a LTAB presentation and appended to the FCB table and used for spooling.

You can use the PDELETE FCB command to clear this FCB table. As a result, the FCB table is refreshed with the new FCB data.

For example, use the above command when you want to change the contents of a previously used FCB image and you want the changes to take effect by leaving your jobs unchanged without starting VSE/POWER again.

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. If no FCB image is specified, VSE/POWER loads the default FCB.

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 cuu address)

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
```

Changing Default FCBs and UCBs

For your printer's default FCBs and UCBs, see "Standard Buffer Image Phases" in *z/VSE System Control Statements*. During installation, you can, however, also create your own default FCBs and UCBs. Note that the name of your printer's default FCB can not be changed; only its values can be altered. For sample procedures to generate your values, see "Creating Print Buffers for a System Printer " in *z/VSE Installation*.

Starting a Reader Task

A reader task reads card-image records from the specified device or from a tape. It writes these records to 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 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,A9
```

starts a list task to print spooled output that has classes A and 9 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 to 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.

For more details, see "Format 1: Processing Disk-Spooled Output" on page 359.

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, see *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 also applies to stopping a *punch task*. Examples of PSTOP commands to stop a punch task are:

```
PSTOP 00D
PSTOP 00D,RESTART
PSTOP 00D,EOJ
```

If Output Records are Ignored

Under certain circumstances list or punch output records are not printed/punched. As much as all records of an output queue entry may be 'ignored' or 'lost', although the list/punch task has processed the entry according to disposition rules. This may happen when the real print/punch device addressed by the PSTART LST/PUN command differs from the device-type used at spooling time, namely when the output was created by job execution.

VSE/POWER considers a mismatch of spooled versus real output device in the following cases:

- **Local** processing: Message 1Q41I MISMATCHING PRINTER/PUNCH .. is issued to inform the operator about the mismatch of spooled and real device (unless this message is suppressed by SET 1Q41I=NO). When retrieving data records of the queue entry a check is made if any CCW operation-code found is compatible with the real printer/punch device.
- **Remote** (RJE) processing: No message is issued. When output is sent to a remote workstation no real printer/punch device is known to the sending VSE/POWER task.
  - For list output, it is only checked if the CCW op-code is a generally valid printer CCW op-code. No check is made if the CCW op-code is valid for the device type specified at spooling time.
  - For punch output, it is checked if the CCW op-code is a valid CCW op-code for the punch device specified at spooling time.

For both local and remote processing, a record is *ignored* when its CCW op-code can not be accepted.

To present more information about ignored output records, VSE/POWER offers the following aids:

Note: This description discusses only print output, but the same is also true for punch output

1. Message 1Q41I names the different device type codes of the devices used at spooling and at real printing time. For details on type codes read the explanation of message 1Q41I.
2. If records with CONTROL op-codes (not combined with printable data) are found which can not be handled by the local real printer or the RJE terminal printer, such records are *ignored without comment*. The same applies to op-codes (even combined with data) which are not WRITE-type (X'01') codes.
3. If records with unacceptable WRITE-type op-codes for printable user data are found, such records are **ignored and counted**, and message 1Q4LI (for local printing) or message 1Q4KI (for remote printing) is issued when processing of

the entry has finished. Both messages present the number of data records ignored per entry. The message explanation provides hints about op-code checking routines for IBM personnel.

4. To avoid that disposition D entries flagged by 1Q4LI or 1Q4KI are deleted when printed - with no more chance of record investigation - the

```
SET IGNREC=DISPY
```

   statement may be included in the VSE/POWER startup procedure. Then flagged entries are preserved with the non dispatchable disposition Y, which indicates that printing has failed somehow.

5. If you want to make the ignored CCW op-codes of a queue entry visible, start an appropriate list task with the option SHOWIGN. Then any ignored record - of both CONTROL or WRITE data type - is printed with its hexadecimal CCW op-code as data followed by up to 25 characters of record data. This provides additional help to locate ignored records in the original data stream as presented by an IPW$$DD dump. Note that this dump contains also internal VSE/POWER control records with op-code X'FF', which are generally bypassed for printing.

For further details on this topic refer to the SET IGNREC statement on page 473 and to the SHOWIGN option of the PSTART command on page 359 and page 364.

When Print Output is Misaligned

VSE/POWER can serve modern FCB controlled printers as well as carriage control tape printers, whose print page layout is described by the LTAB specification of the POWER generation macro or of the * $$ LST statement.

An FCB (or LTAB) is used by VSE/POWER at two different times:

1. At **spool-time** a job is running in a partition and produces output using a printer device which is a 'spooled device' (a device controlled by VSE/POWER defined as such when the partition is started). Then VSE/POWER intercepts the I/O started to the printer device and writes the output data to the VSE/POWER spool file. At this time the FCB is loaded and used to control the number of lines written to a page and to determine the size of a separator page.
2. At **print-time** the print output (created 'long ago') is read from the spool file and passed to a real printer started with the PSTART LST,cuu command. At this time the FCB is loaded into the printer buffer using the z/VSE LFCB macro.

Usually, the FCB used at spool time is the same one used at print time. If they are not the same, the layout of a page may not be as expected; for example, the number of lines per page will cause a splitting of pages at an unexpected position. Therefore, you should trigger the usage of the same FCB at spool and print time by explicitly specifying an FCB in a * $$ LST statement.

If, for non-3800 printers, you do *not* specify an FCB (or LTAB) in a * $$ LST statement, VSE/POWER uses an LTAB at spool-time, namely the LTAB specified in the POWER macro or its VSE/POWER default. At print-time (for FCB type printers) VSE/POWER uses the system default FCB which is printer-type dependent. If the LTAB does not reflect the default FCB's specifiations, it may lead to an incorrect layout of the printed pages.

To avoid incorrect layout, tell VSE/POWER not to refer to the LTAB but to use the printer's system default FCB already at spool time by placing the

```
SET FCB=DEFAULT
```

statement into the startup procedure. For details, see “SET: Setting VSE/POWER Startup Control Values” on page 473 and the following.

If the Account File is Full

When the account file is 80% full, message 1Q31I is issued every minute, provided a task requests more account file space. Thus, you are alerted in time to save the contents of the account file as described below. When the file is a 100% full, you are urged by message 1Q32A to take action. Any VSE/POWER task that must write another account record must wait until you issue a PACCOUNT command. 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.
- Delete all account records of the account file, provided they are really not important for your job accounting.

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.

Several different 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 in the RBS operand of either the POWER macro or in the * $$ LST or * $$ PUN statement for a given job.

Count-driven segmentation produces output segments with the same jobname and jobnumber, but suffix numbers are 1 - 127.

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.

In a PDISPLAY command, it is displayed as S=nnn.

The maximum suffix number is 127. If you have jobs producing more than 127 segments, VSE/POWER assigns a new jobnumber to each following group of 127 segments. For each new job number the suffix number starts with 1 again.

For special considerations related to RBS segmentation of CPDS queue entries, refer to “Page Counting for CPDS Queue Entries” on page 146.

Data-Driven Output Segmentation

You may submit several * $$ LST or * $$ PUN statements within the boundaries of one job. Each of these statements causes an individual list or punch queue entry to be created. Every list or punch queue entry has its unique job number. For details, see "Names and Numbers of Jobs and their Output" on page 404.

The // SETPRT VSE job control statement also can cause list output segmentation.

Multivolume Tape Segmentation

When end-of-volume occurs during unlabelled 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 the correct sequence.

Program-Driven Output Segmentation

In an application program the VSE/POWER IPWSEGM or SEGMENT macro may be used to separate the output. A SETPRT or FCB macro also causes output segmentation. For details, see "IPWSEGM" in *VSE/POWER Application Programming*.

Note: Segmentation due to a 3800 SETPRT request of a PDUMP macro may be suppressed using the SET PDUMP=NOSEGMNT autostart statement. See "SET: Setting VSE/POWER Startup Control Values" on page 473.

Command-Driven Output Segmentation

Whenever output of processing jobs is to be made available as a LST/PUN/XMT queue entry immediately, i.e., before the normal end-of-job, segmentation can be requested dynamically by the following commands:

1. PSEGMENT partition,cuu, which specifies an execution writer task for its spooled output. For details, see "PSEGMENT: Segmenting Output Spooled by Execution Writer Tasks" on page 355.
2. PALTER queue,jobname,jobnumber,CQNUM=qnum,SEGMENT, which specifies directly the spooled output entry (of an exection writer task). For details, see "Format 3: Altering to Segment Job Output In Creation" on page 209.

Both commands can trigger segmentation either on the next page boundary or immediately. They are thus useful for long-running jobs or when the data file is full (see "Options When Data File Is Full" on page 179). Command-driven segmentation produces output segments with attributes as described for count-driven output segmentation.

Spool-Access PUT-OUTPUT Segmentation

An application program interfacing to VSE/POWER through Spool-Access Support for spooling output into the list/punch queue may use the PUT-OUTPUT segmentation request. This results in production of output segments with attributes as described for count-driven segmentation. For details, see "PUT Service, Output" in *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. During warm-start recovery, VSE/POWER informs you issuing message *1QBCI*.

- 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, "VSE/POWER Disposition Codes," on page 497.

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 interfacing with the VSE/POWER External Device Support 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 ends abnormally 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*.)

Another way of creating such disposition Y entries is the following: output queue entries have been set to a disposition of Y when ignored records were found and SET IGNREC=DISPY was specified in the VSE/POWER autostart procedure.

Assignment of a disposition of Y makes the queue entry unavailable for normal processing. For further information on dispositions, see also Appendix A, "VSE/POWER Disposition Codes," on page 497.

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.

Unrecorded Processing

In support of automating more processes in a z/VSE environment, you can suppress SYSLOG data and avoid queueing of LST/PUN output.

Suppressing SYSLOG Data

You can define which of the jobs that have been submitted to the VSE/POWER reader queue should be logged totally or in parts on the console.

Specifying LOG=NO in an * $$ JOB statement causes the following behavior when the job is processed:

- The VSE/POWER start job logging message 1Q47I is appended by 'LOG=NO' to indicate the absence of typical console information.
- The Job Control '// JOB ...' and 'EOJ ...' logging messages do not appear on the console, but are recorded in the hardcopy file.
- The VSE/POWER waiting-for-work message 1Q34I or 1Q3EI is not issued at all if only 'LOG=NO' jobs have kept the static partition or the dynamic class busy since the last waiting-for-work message.
- The VSE/POWER end-of-job logging message 1QC7I is issued to aid in later reviewing of printlogs.

For details, see the LOG operand in the * $$ JOB statement on page 420.

To reduce console message traffic even further, you can combine the LOG operand of the * $$ JOB statement or the JLOG operand of the POWER macro with the z/VSE Job Control or Attention Routine NOLOG command.

Note: To have all SYSLOG data made available to your system, use the SET LOGEQNO=IGN autostart statement in your VSE/POWER startup procedure. For a description of the operand see page 483.

Suppressing Job Control Statements on SYSLST

List output can be decreased by indicating suppression of job control statements written to SYSLST using the LOG=NO option of the STDOPT statement or command.

Queueing of LST/PUN Output

You may avoid that unessential spooled output is added to the LST/PUN queue. This can be achieved by specifying PURGE=*nnnn* in a * $$ LST/PUN statement. It causes the output collected for the corresponding spooled device not to be made available in the desired queue if the following conditions are true:

1. The job has terminated without cancel condition *and*
2. The maximum user program return code, collected during all job steps of the VSE/POWER job, is less or equal to '*nnnn*' or is not provided at all.

If the PURGE condition is not met, a message alerts the operator. For example,

```
1Q4GI  00E OUTPUT NOT PURGED FOR SERVJOB 00384 IN PARTITION F8
```

indicates that the output spooled for device '00E' has been added to queue, so that the processed steps may be tracked for correction.

For details and restrictions in combination with other JECL operands, see the description of the PURGE operand on page 440 and page 454.

Suppression of VSE/POWER Individual Messages Generally

The PVARY MSG command (see "Format 4: Varying VSE/POWER Messages" on page 391) may also be used to disable nearly all VSE/POWER messages displayed on the console but are still recorded in the hardcopy file.

Job Scheduling Aids

The spool-access support CTL service offers valuable flag byte-coded information when scheduling of VSE/POWER reader jobs is monitored by a programmed interface. This information is offered in the reader queue display and is available to you when the FORMAT option is used; for details on the FORMAT option, see the "PWRSPL macro" in *VSE/POWER Application Programming*.

For example, the display byte PXFMFLG2 indicates for each job and its defined processing class the likelihood that the job can be scheduled for execution at the current operational state of the z/VSE system; for details on the display byte PXFMFLG2, see "PWRSPL DSECT" in *VSE/POWER Application Programming*.

The scheduling information included in this snapshot byte are:
- Whether the job processing class is defined for a static partition, in which case the following information is also included:
- Corresponding static partition is currently running
- Corresponding static partition is just waiting for work.
- Whether the job processing class is defined as a dynamic class, in which case the following information is also included:
- Corresponding dynamic class is currently suspended
- Corresponding dynamic class is enabled.

The following points should be noted when interpreting the above flags:
1. Scheduling information is only offered for reader jobs that are not in the execution preparation phase (PXFM2PRP indication) or not in execution state (PXFMDISP=*). For information on the execution preparation phase see page 140.
2. A processing class may be defined both as 'static' and 'dynamic' at the same time
3. No information is provided for undefined, invalid, or disabled dynamic classes.

Browsing of Queue Entries

Browsing means viewing the contents of a VSE/POWER queue entry without modifying its attributes, such as deleting it after viewing or changing its spooled data.

Parallel Browsing and Tracking of Multiple Access

The spool-access support GET service for browse views local queue entries without modifying them. For details, see "Starting the GET Service" in *VSE/POWER Application Programming*. Support for browsing even allows gaining browse mode access to entries that are being processed (DISP=*) by a task able to delete the entry after processing (called an "update task") and also being browsed by other tasks. Parallel browsing is limited per queue entry by

- Maximum 255 browsers in a non-shared VSE/POWER system
- Maximum 15 browsers per system-ID in a sharing VSE/POWER system environment.

The concurrently browsing users are tracked by the Multiple Access Count(s) which are presented with the 'MACC='value(s) of a PDISPLAY ...,FULL=YES command request. For a display example, see page 257.

In case of system failure and VSE breakdown, the recovery of VSE/POWER maintains the Multiple Access Count per non-shared or sharing system. If a failing sharing system can not be brought up again, then use the PRESET command to erase the browsing users of the failing system-ID.

Delaying the Deletion of Browsed Queue Entries

Update tasks, such as a local list task, select a queue entry for processing according to its dispatchable disposition (D or K) and after processing may change the entry's disposition from K to L or even delete the entry with all of its spooled data. This means that update tasks select a queue entry for private processing. They set it to 'active' as indicated by the disposition '*' column in any queue display. A special example of an update task is a VSE/POWER command processor such as for a PALTER command. The command processor does not set the queue entry to active but it does disable selection of the queue entry for other tasks until it has finished processing the specified command.

Browse tasks, on the other hand, when accessing a queue entry for viewing, are limited to a maximum number of parallel browsers per entry. Also, browse tasks do not set a selected entry to 'active'. Instead, the Browse column of any queue display indicates by an asterisk ('*') that a queue entry is being viewed by at least one browse task.

With these rules established for access to queue entries by different task types, browsers must be protected against deletion of the queue entry while it is being browsed. This protection is achieved by *delaying deletion of the queue entry until the last browser has ended*. In detail, the following takes place at deletion of a browsed entry:

1. The queue entry is removed from its class chain, thus making it no longer eligible for further selection.
2. The number of browsers is checked and, if not yet zero, deletion of the queue entry with its data is delayed.
3. Whenever a browser finishes processing an entry in *delayed deletion*, the number of browsers is decremented and checked again.
4. When the last browser ends, the queue entry is physically deleted along with its data.

Queue entries in delayed deletion are no longer queued in one of the class chains of the physical (RDR,LST,PUN,XMT) queues. They can be made visible only with the PDISPLAY DEL or PDISPLAY TOTAL commmand (see "Format 4: PDISPLAY DEL" on page 265 and "Format 5: Displaying Information for All Physical and Logical Queues" on page 236). These entries still claim data file space with their used DBLK groups until they are finally deleted when the last browser ends. The display of the Deletion Queue reveals in the DBGP column the spool space consumption per queue entry. In addition, the PDISPLAY STATUS report (see page 38) shows all queue records and all DBLK groups 'in deletion'.

31-Bit Addressing Considerations

Apart from access to the storage copy of the queue file in partition Getvis, VSE/POWER does not support any 31-bit addressing. VSE/POWER itself does not use any 31-bit addresses for its own code, control blocks or data areas, nor does it accept any 31-bit addresses for control blocks supplied by the Supervisor or any user.

Whenever the user passes data areas to VSE/POWER, these data areas must be within the 24-bit address range. This concerns:

- The I/O requests for a device spooled by VSE/POWER (SVC 0, SVC 3)
- The accounting requests for a device spooled by VSE/POWER (SVC 90, SVC 91)
- The segmentation requests by means of the IPWSEGM or SEGMENT macro
- The addresses passed between VSE/POWER and any user written exit routine (JOBEXIT, OUTEXIT, PNET receiver exit, and PNET transmitter exit)
- The requests for the cross-partition communication support (based on the SPOOL macro and XECB macro).

In most of the above cases, VSE/POWER does not know that the passed addresses should be used as 31-bit addresses. VSE/POWER just uses the addresses as 24-bit addresses, and the results thereby are unpredictable:

1. The request may be canceled due to invalid address
2. The request may complete without any error message, but instead using the data addressed via a 31-bit address VSE/POWER processes the data addressed via a 24-bit address.

In the following cases, VSE/POWER comes to know that 31-bit addresses should be used, and, therefore, issues an error message:

1. If the CCB for an I/O or accounting request indicates that Format-1 CCWs are to be used (which means 31-bit addresses should be used), VSE/POWER issues the message 1R30I with the reason code 12 and cancels the job.
2. If the CCW or the data area for an I/O or accounting request resides within the 31-bit shared area, VSE/POWER issues the message 1R30I with the reason code 3, respectively 4, and cancels the job.

User written exit routines (JOBEXIT, OUTEXIT, PNET receiver exit, or PNET transmitter exit) must *not* use the 31-bit addressing mode. When an exit routine loses control (for example due to a page fault), the status of the currently used addressing mode is not saved. When the exit routine is given control again, the previously used addressing mode is not restored. Therefore, if an exit routine uses the 31-bit addressing mode, the results are unpredictable.

Starting an RJE,BSC Line

You do this by starting a VSE/POWER RJE task. This task processes job streams from a terminal and builds a reader queue entry for every VSE/POWER job transmitted from that station. On request from the terminal operator, an RJE print or punch task makes output available for transmission to the terminal. 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 terminal operator can sign on at a terminal linked via this line. The operator's SIGNON command is accepted only if it specifies the password RJEOK09.

For details on password processing, see "Security" in *VSE/POWER Remote Job Entry*.

Starting an RJE,SNA Session

VSE/POWER's RJE,SNA support uses 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 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 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 a 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 terminal
All terminals
Another node
Terminal 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 ID, or all output directed to this remote ID.

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 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 terminal.

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 an RJE,SNA Session

To stop an individual user session, use a PSTOP command in this format:

```
PSTOP RJE,SNA,PRLL03,EOJ
```

In the command, PRLL03 is the logical unit name that was used to define the affected terminal to VTAM. EOJ in the command ensures that all input or output for the currently processed VSE/POWER job is 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.

Four-Digit Year (Year 2000) Support

This section summarizes all actions and provisions made in VSE/POWER 6.1.2 to enable a smooth transition from the 20th to the 21st century for all date-related processing of VSE/POWER 6.1.2 and all subsequent versions and releases.

The date-related processing is based upon the contents of the VSE/POWER partition Comreg field JOBDATWC, which is an 11 byte date field of the following format:

```
JOBDATWC  mm/dd/yy/cc   for example: 10/22/98/19
```

The field reflects either

- The total system date established by the 'SET DATE' IPL command, which sets the TOD [9] clock, or
- The VSE/POWER partition singular date established by the// DATE mm/dd/yyyy Job Control statement, which does not set the TOD clock.

The *mdy* or *dmy* format of this field is discussed in "Date Recording and Date Format" on page 127.

When simulating a century switch, for example, both methods may be used to change the current date. However, that should only be done on test systems, because all VSE/POWER queue entries remember their creation date as well as their TOD Store Clock time stamp, which determines the queueing sequence of a queue entry within its class chain at queue file recovery time.

Four Digit Year at Queue Entry Creation

Whenever job or output queue entries are created, even when received by PNET or loaded from tape, they are always branded with the current date including the cc-century (QRDYC field) and the yy-year (QRDY field). The only way to preserve an existing old creation date is to specify the NOJNO operand during POFFLOAD LOAD or SELECT.

You can display the creation date of queue entries with the FULL=YES option of the PDISPLAY command. See 'D=mm/dd/yyyy' in "Examples of the PDISPLAY Command" on page 248.

Four Digit Current Year in Messages

All time and status messages and even separator pages of list output present the date with cc-century and yy-year. For details see Figure 7 on page 46 and "Separator Pages - Layout and Control" on page 181.

Four Digit Year in Interface Control Records

For programmed interface communication by Spool-Access CTL or GCM Service requests, VSE/POWER also presents extended date qualification. For details see *VSE/POWER Application Programming*.

Unique Century Identification in Account Records

For a precise evaluation of accounting data, VSE/POWER identifies the 20th or 21st century by a flag in all types of account records. Thus, all account records keep their existing length and do not influence on running user accounting programs. For details see *VSE/POWER Application Programming*.

Transparency for an Old Two Byte Creation Year

VSE/POWER 6.5 code may still find queue entries which have been built with a two byte creation year, namely when

9. TOD = time-of-day clock, a binary counter of fractions of seconds elapsed since midnight, January 1, 1900.

- any pre 6.1.2 queue entry is loaded from tape while specifying the NOJNO operand of the POFFLOAD LOAD/SELECT command
- a PDISPLAY Tape command is started for any pre 6.1.2 offload or DISP=T tape.

For such queue entries all display or other date-related functions will temporarily expand the yy-creation year to a 4-byte year by applying the following **fix-88-window** rule

```
if yy >  88,  then assume 19yy
if yy <= 88,  then assume 20yy
```

Note: This rule has been always used for VSE/POWER's Time Event Scheduling support, when DUEDATE=mm/dd/yy is specified in the * $$ JOB statement.

So even then a 4 digit creation year is displayed and any date-based entry selection operates, as if a 4 digit creation year was present.

CRDATE Command Selection with a 2 or 4 Digit Year

The selection operand of the PDISPLAY, PALTER, PDELETE, PHOLD, and PRELEASE commands has allowed to specify a comparison operand and an mm/dd/yy limit date to be compared against the creation date of queue entries to be selected. Starting with 6.1.2, VSE/POWER

- Still accepts the specification of a 2 digit year and expands it during comparison according to the fix-88-window rule, and
- Also supports the specification of a 4 digit year.

For details, refer to the CRDATE= operand of the mentioned commands, for example PALTER on page 203.

Date Recording and Date Format

As described in "Four-Digit Year (Year 2000) Support" on page 125, all VSE/POWER date-related processing is based on the VSE/POWER partition Comreg field JOBDATWC, which may mean either:

- mm/dd/yy/cc, in the default sequence of month/day/..., or
- dd/mm/yy/cc, in the sequence of day/month/..., as specified by the// STDOPT DATE=DMY statement, which allows changing the date format.

Changing the date format in a running system may cause unexpected confusion for the VSE/POWER functions as well.

Functions Allowing Date Format Changes

The following VSE/POWER functions respect the date format as currently defined for the z/VSE system:

- the PDISPLAY T command returns the current time and date (in the current date format).
- the printed separator pages (see "Separator Pages - Layout and Control" on page 181) present the print processing date in the current format, with alphabetic month (JAN, FEB, etc.).
- the next processing day of the Time Event Scheduling Support can be specified in the * $$ JOB statement by
  - DUEDATE=aabbyy, using the current date format
  - DUEDATE=(aabbyy,D), using the DMY format

– DUEDATE=(aabbyy,M), using the MDY format

to record the scheduling date internally in a fixed format of YMD. Whenever the internal date is addressed

– for display (see "Display of the Next Due Date" on page 166), RUN= or EXP= presents a date format adjusted to the currently active format
– for scheduling of execution, the recorded YMD-date is compared correctly against the active current date.

For these functions, a change of the date format is transparent and will not cause confusion.

Functions Not Expecting Date Format Changes

The following VSE/POWER functions preserve and process a date without recording its original date format as defined for the z/VSE system:

- The 8-byte date header of all VSE/POWER account records represents an accounting date in the format active at processing time.
- When new job or output queue entries are built, their creation date is taken from Comreg JOBDATWC and passed to the first eight bytes of the queue record irrespective of the date format.
- The creation date presented by PDISPLAY...,FULL=YES (or in a fixed format display buffer of Spool Access Support) shows D=mm/dd/yyyy or D=dd/mm/yyyy in whatever date format was active when the queue entry was created.
- The creation date comparison operand (for example, CRDATE < limit-date) of the PDISPLAY, PALTER, PHOLD, PDELETE, or PRELEASE command interprets the limit-date according to the date format active at command entry time and compares it against the queue record creation date, also viewed under the command entry time format (but not creation time format).

This behavior shows that VSE/POWER does not expect the MDY or DMY date format to be changed after a warm start on an existing queue file. Otherwise, the display of creation date becomes misleading and the comparison against the creation date *can fail*.

Therefore, any change of the date format should be done in the following steps:

1. Run POFFLOAD BACKUP or PICKUP to preserve existing queue entries of interest.
2. Re-format the queue and data files by a cold start.
3. Change the date format using // STDOPT DATE=...
4. Re-load the saved queue entries from tape by POFFLOAD LOAD or SELECT, which provides a new creation date for the re-loaded queue entries.

Note: Receiving queue entries via PNET also provides a new creation date.

VSE/POWER Multiprocessor Support

This section describes the multiprocessor support within VSE/POWER. For an overview of the multiprocessor support within the z/VSE system, consult the manual *VSE/ESA Turbo Dispatcher Guide and Reference*. It describes at length how the Turbo Dispatcher works within z/VSE and gives a detailed definition of parallel work units (PA) and non-parallel work units (NP).

Following is just a brief description of these processing units:

Parallel work unit
A series of instructions, typically user applications, which do **not** modify z/VSE system control blocks and which can be given control at any time on any CPU of a multiprocessor system.

Non-parallel work unit
A series of instructions, typically VSE system services which, when given control on a CPU of a multiprocessor system, modify z/VSE system control blocks. That is why at the same time on other CPU's **no** other NP work unit must be processed while PA work units of other partitions can be given control.

Up to release 6.1.0, VSE/POWER executed as non-parallel work unit. Starting with VSE/POWER 6.1.1 it became possible to request parallel processing for VSE/POWER when running on a multiprocessor.

Activating Parallel Processing

When the autostart statement

```
SET WORKUNIT=PA
```

has been included in the VSE/POWER start-up procedure, VSE/POWER:

- Operates as parallel workunit whenever possible
- Switches *temporarily* to a non-parallel unit only to
  - Update the SYSCOM and COMREG's in lowcore
  - Update Supervisor control blocks
  - Issue selected exception SVC's
  - Enter VTAM, Librarian, and Idump or BAM Transient services.

With this concept, VSE/POWER minimizes the time to run as non-parallel work unit. This results in performance benefits for the VSE system when heavily used spooling functions are active, such as batch execution with output spooling, output printing, spool-access cross partition data exchange, and networking. Only for the RJE/SNA function and the seldom required VSE/Subtask services, VSE/POWER operates as non-parallel work unit only.

Determining if the Multiprocessor Support is Active

Use the PDISPLAY STATUS command to see if the multiprocessor support has been activated or not:

When the Turbo Dispatcher has **not** been activated at IPL time, VSE/POWER operates non-parallel only and ignores any request for switching of work units. In this environment, the second PDISPLAY STATUS line appears as

```
PRESENT SESSION START              ON mm/dd/yyyy   TIME hh/mm/ss
```

When the Turbo Dispatcher has been **activated** at IPL time, and VSE/POWER has been initialized **without** the WORKUNIT startup option, then VSE/POWER operates non-parallel only and ignores any request for switching of work units. This is the default VSE/POWER environment for the Turbo Dispatcher; the second PDISPLAY STATUS line appears as follows:

```
PRESENT SESSION START (TURBO-DISP.-NP) ON mm/dd/yyyy   TIME hh/mm/ss
```

When the Turbo Dispatcher has been activated at IPL time and VSE/POWER has been initialized **with** the SET WORKUNIT=PA startup option, then the

VSE/POWER multiprocessor support is enabled to process as parallel work unit whenever possible. That means, requests for switching to PA - and where necessary - to NP work units are passed to the Supervisor. Now, the second PDISPLAY STATUS line appears as

```
PRESENT SESSION START (TURBO-DISP.-PA) ON mm/dd/yyyy   TIME hh/mm/ss
```

Passing Control to VSE/POWER User Exits

In a multiprocessor environment, you should check your user exits or vendor exits. Output Exits (OUTEXIT) and Transmitter Exits (XMTEXIT) are given control for **every** record processed by the corresponding VSE/POWER function. Reader Exits (JOBEXIT) and network Receiver Exits (NETEXIT) gain control only for network control records and selected JECL/JCL statements, but **not** for ordinary data records. With multiprocessor support enabled, the calling VSE/POWER function always operates in parallel state and passes control to the user exit:

- By default as **non-parallel (NP) work unit**, because it is unknown which internal control blocks are updated by the exit code. Upon return, VSE/POWER switches back to parallel state.
- As **parallel (PA) work unit**, provided the exit has been defined as such for loading by either of the following:
  1. The POWER generation macro (see page 57), or
  2. The PLOAD command (see page 319).

  The following example of a JOBEXIT named MYJOBEX with a 200 bytes workarea shows the loading specifications:

  ```
  1. POWER macro operand: JOBEXIT=(MYJOBEX,200,PA)
  2. using command:       PLOAD JOBEXIT,MYJOBEX,200,PA
  ```

  For performance reasons it is strongly recommended to define an exit as parallel work unit in order to increase the percentage of PA code in your system and to avoid extra switches from PA to NP work unit and back again, whenever control is passed to an exit. But, before loading an exit with the 'PA' option, you should check and verify that the exit code does **not** update
  - The first page in lowcore consisting of SGLOWC, SYSCOM, and BG-COMREG
  - Any Supervisor owned control blocks located anywhere.

  See also "Activating Parallel Processing" on page 129 with a list of requests for which VSE/POWER must operate as a non-parallel work unit.

Identifying the Processing Mode of User Exits

The PDISPLAY EXIT command allows you to identify the type of work unit (WU) for which the exit should be given control. Assume, for example, an OUTEXIT named MYOUTEX which has been loaded with the 'PA' work unit specification. The exit display will then present:

```
1R4AI  EXITTYPE STATE    NAME     .....  .....  EXITSIZE WU
1R4AI  OUTEXIT  ENABLED  MYOUTEX  .....  .....  00910    PA
```

Note: The PA-option for exits takes effect only when the VSE/POWER multiprocessor support has been enabled.

For details on PDISPLAY EXIT, see page 245.

Avoiding Failures

You may wish to drop out from VSE/POWER's multiprocessor support and run the whole session only as non-parallel work unit for the following reasons:

- IBM or vendor products interacting with VSE/POWER no longer function correctly because of modified dispatching sequence of the VSE/POWER maintask versus its VSE/Subtasks.
- The small overhead for switching of VSE/POWER work units should be avoided when running the Turbo Dispatcher in a uniprocessor-only environment.
- You want to measure for your workload the performance improvement of the VSE/POWER multiprocessor support versus a non-parallel-only VSE/POWER.

In this case, remove the SET WORKUNIT=PA statement from the VSE/POWER start procedure for a subsequent default startup in the Turbo Dispatcher environment. Then, no request to switch work units is passed to the supervisor at all.

Failure Message 1Q2CI

When VSE/POWER fails and terminates abnormally, message 1Q2CI provides the following breakdown information:

A/T= with explanation on **A**ccess-register mode followed by **T**urbo Dispatcher work unit been active at the time of breakdown.

Access register mode may appear as **ON** or **OFF**. **T**urbo Dispatcher mode may appear as:

-- if the Turbo Dispatcher was not activated.

NP if failing task processed a non-parallel work unit and VSE/POWER parallel processing was activated with the `SET WORKUNIT=PA` autostart statement.

PA if failing task processed a parallel work unit and VSE/POWER parallel processing was activated with the `SET WORKUNIT=PA` autostart statement.

WN if failing task processed a non-parallel work unit and VSE/POWER parallel processing was not activated.

WP if failing task processed a parallel work unit and VSE/POWER parallel processing was not activated. This combination should not occur.

Utilizing the Multiprocessor Efficiently

The following hints explain how you can use the VSE/POWER multiprocessor support efficiently through VSE/POWER's generation options. In general, in a Turbo Dispatcher environment the aim is to **reduce** non-parallel work units, especially for often repeated actions in a spooling system. Such requests are:

- Read/Write requests on the data file
- Line Read/Write requests from PNET/BSC or PNET CTCA
- VTAM Send/Receive requests from PNET/SNA

Note: TCP/IP Send or Receive requests from PNET/TCP are driven as parallel work units due to the attributes of the TD Subtask interfacing to TCP/IP.

Data File I/O

Increase the DBLK operand of the POWER macro beyond the shipped default of about 7,500 bytes (see page 61). Also, reduce the DBLKGP value to reproduce your existing product of DBLK times DBLKGP (recommended product value: 60KB - 90KB). For details refer to "The Size of the DBLK Group" on page 37.

Note: An increased DBLK size results in a higher VSE/POWER partition Getvis-24 consumption.

PNET I/O

increase the BUFSIZE operand of the PNODE macro (see page 81) beyond the default of 1112 bytes. For the increased storage consumption, note that BSC and CTCA line buffers reside in the SETPFIX LIMIT area of the VSE/POWER partition, while buffers for VTAM connected, PNET TCP, or PNET SSL linked nodes are acquired from the VSE/POWER partition Getvis-24 area. For details see "The Size of the VSE/POWER Partition" on page 44.

For better multiprocessor exploitation you may also check the coding style of your frequently used application programs:

- If such programs use the SEGMENT macro, which ties up the Logical Transient Area for the duration of a simulated segment I/O request, you should replace the SEGMENT call by the modern IPWSEGM macro to increase the parallel share of your workload. For macro conversion refer to "Output Segmentation by SEGMENT Macro" in *VSE/POWER Application Programming*.

Dynamic Partition Support

The dynamic partition support extends the z/VSE job execution capability remarkably. Dynamic partition support allows jobs to allocate resources as they are needed and release them immediately after use. For an overview of the role of dynamic partitions within the z/VSE system, please refer to *z/VSE Planning*.

Dynamic partition processing is best understood when comparing it to the processing of static partitions.

Static Versus Dynamic Partition Support

A static partition is brought under VSE/POWER control by the 'PSTART partition' command. This partition must be allocated in an address space and must be in the stopped state. VSE/POWER asks for spooled devices and simulates the Attention Routine 'START partition' command to activate the static partition for z/VSE Job Control. Upon the first request to read the job data, VSE/POWER selects a dispatchable job entry with matching execution class. Processing of jobs starts and continues for further jobs until the static partition is removed from VSE/POWER control by the 'PSTOP partition' command or by the PEND termination.

Dynamic Partition Support, on the other hand, requires that the PLOAD DYNC command be entered to load the DTR$DYNC.Z member of the z/VSE IJSYSRS.SYSLIB library and to enable its dynamic classes for scheduling of jobs. Member DTR$DYNC.Z contains user specified characteristics of job classes, for which dynamic partitions may be allocated. Each partition processes one job of its corresponding VSE/POWER job input class. For further details about the definition of the DTR$DYNC.Z member refer to "Defining Dynamic Class Tables" in *z/VSE Administration*.

Note: Only up to 23 dynamic classes are accepted within a dynamic class table. Otherwise, the PLOAD command warns you by message

```
1Q6SA  TOO MANY CLASS ENTRIES FOUND - SURPLUS IGNORED
```

For environments sharing the z/VSE IJSYSRS.SYSLIB, each sharing system can address a different setup of dynamic classes. Specifying the ID=*n* operand of the

PLOAD command, a specific dynamic class table DTR$DYN*n*.Z may be accessed in the same SYSLIB. Non-shared systems may also use this ID operand to gain more flexibility in definition of dynamic classes.

After PLOAD DYNC has established the active Dynamic Class Table, VSE/POWER selects a dispatchable job corresponding to an enabled dynamic class, for example class Q. A dynamic partition 'Qm' is allocated in its own address space. Spooled devices are obtained from the active Dynamic Class Table and the dynamic partition is activated for z/VSE Job Control to process the profile procedure specified for the dynamic class Q. Upon the first request to read job data, VSE/POWER returns data of the selected job in class Q, and handles further read requests in the same way. After the VSE/POWER job has been processed, the dynamic partition is deallocated.

VSE/POWER continues to select jobs for execution in a dynamic partition as long as dispatchable reader queue entries exist for corresponding enabled dynamic classes.

Class(es) used for dynamic partitions can also be used for static partitions. Yet, if a job enters the queue for that class while a static partition is waiting for work and a dynamic class is enabled for allocation at the same time, a dynamic partition will be selected in most cases first to process the job. See also "Processing Attributes" below.

Attributes of Dynamic Partitions

Spooling Restrictions and Defaults

When a new dynamic class table is activated by the PLOAD DYNC command, VSE/POWER verification routines make sure that at least one valid spooled reader, printer and punch device is defined for each dynamic class. Hence, VSE/POWER does

- **not** support 'writer-only' dynamic partitions.

The VSE/POWER internal start of dynamic partitions yields the following attributes for all dynamic partitions:

- The default output class is A. If the SET DYNOUTCL=DYNCL autostart statement is active, the default output class is the execution class itself. The default may be overwritten by a * $$ LST/PUN statement.
- A higher z/VSE priority than that of the VSE/POWER partition is accepted. The PRTY command does not reject a dynamic class with a higher priority than that of the VSE/POWER partition itself. This is not, however, recommended for typical batch jobs, because it lowers the performance level of VSE/POWER's own functions, such as networking, printing, tape processing and remote job entry.
- MT-type dynamic partitions are not supported. For MT, see "Format 8: Putting a Static Partition Under VSE/POWER Control" on page 371.

Processing Attributes

Unless restricted explicitly, VSE/POWER offers the same functional support to the static as to the dynamic partitions. Consider the following for dynamic partitions:

- When a dynamic class is also defined as a processing class for a static partition, then
  - VSE/POWER allocates in most cases a dynamic partition for processing of a job instead of passing the job to a waiting static partition

  - but when high system paging is active before VSE/POWER job selection starts, then the job may be passed to a waiting static partition.

  If, on the other hand, you want to pass the job first to a waiting static partition, include the autostart statement SET DYNAL=LOW in the VSE/POWER autostart deck. In this case, the existing static partitions are serviced first before a new dynamic partition is allocated.
- The z/VSE processing priority, defined by the PRTY command for dynamic classes, is reflected in the VSE/POWER job selection algorithm for enabled classes. For example, changing class Q from a lower to a higher class priority has an immediate effect on VSE/POWER. Hence, jobs of processing class Q are selected first for allocation of a dynamic partition before other processing classes of lower priority are served.

  A priority balanced group of, for example, C=D=E is, however, not reflected. Instead, VSE/POWER interprets this specification as an ascending sequence.
- Only **one** VSE/POWER job is processed in its allocated partition before the partition is stopped and its address space is deallocated. Since this happens frequently, the message 1Q33I (partition stopped) is not issued for dynamic partitions.
- Whenever the last active dynamic partition of any dynamic class cannot find a dispatchable job of its execution class in the queue, the message

  ```
  1Q3EI  DYNAMIC CLASS 'x' WAITING FOR WORK
  ```

  is issued.
- When the VSE/POWER PEND termination period is entered, all jobs running in static and dynamic partitions are allowed to operate until the end of the job, at which time the partition is stopped. No new job is selected for execution after the PEND command is entered.
- If VSE/POWER terminates abnormally while jobs are active in dynamic partitions and is restarted with the SET NORUN=YES autostart option, the dynamic partitions will not enter the z/VSE '// PAUSE mode'. For more details, see also the message 1Q36I.
- If VSE/POWER terminates abnormally while jobs are active in dynamic partitions and is restarted with SET NORUN=YES,DYN1=PAUSE all dynamic classes with 'maximum number of partitions within class equal to 1', which are started by PLOAD DYNC during autostart, and which find a job eligible to run, are placed into Job Control // PAUSE mode.
- SLI inclusion from a VSE sublibrary will replace the leading '$$' of the membername by 'c$', in case the job executes in a dynamic partition. 'c' is the dynamic class character, which means the first character of the dynamic partition-id.

Processing Sequence of Jobs

The dispatchable jobs for **one** specific execution class in the reader queue are processed as described in the following cases:

1. A static partition supporting the class processes the jobs one by one according to their queueing sequence.
2. A corresponding dynamic class (allowing only *one* dynamic partition to be activate) processes jobs one by one the same way as for static partitions.
3. But, a corresponding dynamic class (allowing more than one dynamic partition to be activated) processes the jobs in parallel with no predefined sequence.

Note that although both cases 1 and 2 yield the same processing sequence, the jobs may behave differently whenever *one* z/VSE job spans more than one

VSE/POWER job. This is because for each job a dynamic partition is allocated and z/VSE Job Control is newly initialized. (Also see "Job Streams with JECL Delimiters" on page 401).

Dynamic Partition Support - Operation

Activate Dynamic Partition Support

Loading a Dynamic Class Table for Verification: The PLOAD DYNC,VERIFY command helps you 'debug' a future Dynamic Class Table before having it activated. It loads the DTR$DYNC.Z member of the z/VSE library into the VSE/POWER area and checks all specifications of the class characteristics. No active Dynamic Class Table is produced. Instead, possible error messages and a status display of the verified Dynamic Class Table are returned.

Example: The following example shows the display of the verified Dynamic Class Table in response to a PLOAD DYNC,VERIFY command for an arbitrary member DTR$DYNC.Z:

```
1Q6AI  ******  VERIFIED DYNAMIC CLASS TABLE  DTR$DYNC.Z  ******
1Q6AI  CLS STATE   ACT/MAX ALLOC    SIZE  SP-GETV  PROFILE LUBS
1Q6AI   G  *INV*     0 32     4M    300K    128K  *ASIPR-G  110
1Q6AI   -* *INV*     0 32     2M    300K   2000K*  ASIPRC   100
1Q6AI   Q  VALID     0  5     1M    200K    128K   ASICQ     50
1Q6AI   R  VALID     0  5     1M    200K    128K   ASICQ     50
1Q6AI   X  *INV*     0 42*    1M    100K    ---K*  ASIX      40
1Q6AI   P  *INV*     0  8     1M   1800K*   128K   ASIPROCP 440*
1Q6AI   Y  *INV*     0  8     1M    ---K*   168K   ASIPROCY  40
1Q6AI   B* *INV*     0 42*    1M    100K    128K   ASIB      40
1Q6AI   M  INV-SP    0 10 16000M*   800K    168K   ASIM      30
1Q6AI   N  INV-SP    0 10     4M    600K    164K   ASIN      30
1Q6AI  CLS M - UNDEF, DUP, OR NO HEX RDR: 0H2
1Q6AI  CLS M - UNDEF, DUP, OR NO HEX PUN: ---
1Q6AI  CLS N - UNDEF, DUP, OR NO HEX PRT: 02E,03E,04H
1Q6AI  CLS N - INVALID PRT: 06D
1Q6AI  CLS N - INVALID PUN: 04E,05E,06E,07E,08E
1Q6BI  DYNAMIC CLASS TABLE VERIFIED
```

The verification results are reflected in the STATE column and may appear as

VALID
: when all class specifications are correct.

INV when incorrect class specifications have been detected. These are marked by '*'. For details on flagged ALLOC, SIZE, or SP-GETV, refer to "Storage Management" in *z/VSE Guide to System Functions*. A display of '-' indicates either that a required field (CLASS, ALLOC, or SPOOLED DEVICES) is not specified at all, or that an optional numeric field contains non-numeric specifications.

INV-SP
: when incorrect class specifications, including the definition of spooled devices, have been detected. Additional messages identify the device number and failure. All missing spooled device types are signalled by '---'.

Loading a Dynamic Class Table Conditionally: The PLOAD DYNC,COND command helps you load a Dynamic Class Table and enable its dynamic classes, provided that **all** specifications are correct. If not, an error summary is produced. Any possibly active dynamic partition scheduling is, however, not interrupted.

The PLOAD DYNC,COND command loads and checks member DTR$DYNC.Z as done by the VERIFY option. If all classes are specified correctly, an active Dynamic Class Table is created with all classes **enabled** for dynamic scheduling of jobs, unless a certain class specifies 'initially disabled.' Such a class is set to 'DISABled' state and may be enabled later using the PVARY command. A status report of the Dynamic Class Table is then displayed.

If one or more classes are specified incorrectly, the command continues as if PLOAD DYNC,VERIFY had been entered. The PLOAD DYNC attempt is terminated by message

```
1Q6BI  DYNAMIC CLASS TABLE NOT LOADED
```

Example: The following example shows the display of the active Dynamic Class Table in response to a successful PLOAD DYNC,COND command for an arbitrary member DTR$DYNC.Z:

```
1Q6AI  ******   ACTIVE DYNAMIC CLASS TABLE DTR$DYNC.Z    ******
1Q6AI  CLS STATE   ACT/MAX ALLOC    SIZE  SP-GETV  PROFILE LUBS
1Q6AI   G  ENAB      0 32     4M    400K    128K   ASIPRCG  110
1Q6AI   X  ENAB      0 15     1M    100K    128K   ASIX      40
1Q6AI   P  ENAB      0  8     2M    200K    128K   ASIPROCP  40
1Q6AI   M  DISAB     0  8     1M    200K    168K   ASIPROCM  15
1Q6BI   DYNAMIC CLASS TABLE LOADED SUCCESSFULLY
```

Note: When an active dynamic class table containing classes with currently active partitions should be replaced by a new table, all these classes must be contained in the table to be loaded. If the load request finds even one dynamic class with active partition(s) not contained in the new Dynamic Class Table, the PLOAD request is rejected. The verified table is displayed, and the command is terminated by the following message:

```
1Q6BI  DYNAMIC CLASS TABLE NOT LOADED - ACTIVE CLASS(ES) MISSING
```

Loading a Dynamic Class Table Unconditionally: The PLOAD DYNC,FORCE command helps you load a Dynamic Class Table even if a class has incorrect specifications. This command operates as the PLOAD DYNC,COND version, provided that all classes are specified correctly. When a class is invalid, an active Dynamic Class Table is created anyhow. Invalid classes, however, will enter the '*INV*' or 'INV-SP' state, and they are disabled by definition.

Example: The following example shows the display of the active Dynamic Class Table in response to a PLOAD DYNC,FORCE command for an arbitrary member DTR$DYNC.Z:

```
1Q6AI  ******   ACTIVE DYNAMIC CLASS TABLE DTR$DYNC.Z    ******
1Q6AI  CLS STATE   ACT/MAX ALLOC    SIZE  SP-GETV  PROFILE LUBS
1Q6AI   G  *INV*     0 32     4M    300K    200K  *ASIPR-G  110
1Q6AI   M  INV-SP    0 10     2M    200K    168K   ASIM      30
1Q6AI   N  ENAB      0 32     1M    150K    128K   ASIPRCN   15
1Q6AI   X  ENAB      0 15     1M    150K    128K   ASIX      40
1Q6AI   P  ENAB      0  8     1M    100K    128K   ASIPROCP  40
1Q6AI   R  DISAB     0  8     1M    100K    128K   ASIPROCR  15
1Q6BI   DYNAMIC CLASS TABLE LOADED - WITH INVALID CLASSES
```

No detailed messages about incorrect definitions of spooled devices are appended to this display. To obtain further message information, enter the additional command PLOAD DYNC,VERIFY.

Replacing an Active Dynamic Class Table: You may use the COND (by default) and FORCE option to replace an active class table by a new one. Any dynamic partition started before the replacement is allowed to run until the end of the VSE/POWER job. Yet the new table is accepted only if classes of running partitions are also contained in the new Dynamic Class Table, even with changed features or with invalid specifications. For more details, please refer to the PLOAD DYNC,COND command on page 321.

PLOAD Authorization and Validity: The central operator and zero password authorized spool-access CTL users are authorized to use the PLOAD DYNC command. The command is accepted during VSE/POWER autostart time. It is rejected after the PEND command has been entered.

Display Features of Dynamic Classes

Displaying the Active Dynamic Class Table: The PDISPLAY DYNC command may be used to get information about predefined features and actual state of classes contained in the active Dynamic Class Table. This command is not limited to any type of operators. It functions as soon as a Dynamic Class Table has been activated by the PLOAD DYNC command.

Example: The following example shows the display of the active Dynamic Class Table DTR$DYNC.Z (any arbitrary library member) in response to a PDISPLAY DYNC,ALL command. It reflects the user-specified attributes of a dynamic class, the current class 'state', and the currently 'active' number of partitions within the specified maximum.

```
1Q6AI  ******   ACTIVE DYNAMIC CLASS TABLE DTR$DYNC.Z    ******
1Q6AI  CLS STATE   ACT/MAX ALLOC    SIZE  SP-GETV  PROFILE LUBS
1Q6AI   G  ENAB      6 32     1M    200K    168K   ASIPRCG   15
1Q6AI   X  ENAB      0 15     2M    300K    200K   ASIX      15
1Q6AI   P  ENAB      4  8     1M    150K    128K   ASIPROCP  15
1Q6AI   Q  DISAB     0  5     1M    100K    128K   ASICQ     12
1Q6AI   M  SUSPEND   8  8     1M    100K    128K   ASIPROCM  15
```

Notes:

1. Class M, which is actually enabled, is currently SUSPENDed for allocation of more dynamic partitions, because its maximum number of partitions per class has been reached. As soon as an active partition of class M terminates, class M will be resumed automatically for further allocation. This means that class M will re-enter the ENABled state.
2. The PDISPLAY DYNC command does not present the spooled devices. If this information is required, you can display the corresponding member DTR$DYN*n*.Z using z/VSE Librarian services or the Interactive Interface services.

For specification of dynamic class features, as shown in a PDISPLAY DYNC output, refer to the *z/VSE Administration* manual. For an additional sample display with description, see "Format 12: PDISPLAY DYNC,ALL" on page 280.

Halt and Resume Dynamic Partition Allocation

Disabling a Dynamic Class: Any dynamic class of the active Dynamic Class Table may be disabled for further dynamic scheduling of jobs by means of the PVARY DYNC,DISAB command. Any active dynamic partition within the disabled class is allowed to process until the end of the running VSE/POWER job.

Enabling a Dynamic Class: Whenever a dynamic class of the active Dynamic Class Table has been disabled or has been loaded with the 'disable initially' option, it can be enabled for dynamic scheduling of jobs by means of the PVARY DYNC,ENAB command. The PVARY command may be used to influence the system load or to tune an existing Dynamic Class Table until an approved predefined table has been established.

PVARY Authorization and Validity: The PVARY DYNC command is offered to the operator and to master password authorized spool-access CTL or CTLSPOOL users. The command is only accepted if an active Dynamic Class Table has been established by the PLOAD DYNC command. During the PEND period, the PVARY DYNC command is accepted but does not influence dynamic job scheduling anymore.

Partition Related Commands

Dynamic partition limitation: The influence of partition related commands on dynamic partitions must be seen under the following rules of dynamic partition support:

- The partitions are started without operator intervention.
- The job execution class and the dynamic class match.
- The partitions are stopped after one VSE/POWER job without operator influence.

PALTER partition,class: Providing a new input class is not supported for dynamic partitions, any attempt is rejected with the message 1R52I.

PSTART partition: This command starts a static partition. PSTART cannot be used to start a dynamic partition. Any attempt to do so is rejected with the message 1R90I.

PSTOP partition: The partition stop initiated by the operator is meaningless for dynamic partitions. They are always stopped after completion of the VSE/POWER job.

PFLUSH partition: This command will also be supported for 'partition' = partition-id of a dynamic partition. Also, the 'PCANCEL jobname' command may be used for jobs running in a dynamic partition. However, neither command is effective during profile processing of Job Control at initialization of a dynamic partition.

PGO partition: This command allows the set up of continuation conditions when prompted by message 1Q57A or 1Q58A for tape mounting while spooling to tape. It will also support the partition-id of a dynamic partition.

PDISPLAY A: This command shows active tasks associated with both active static and dynamic partitions. For selection criteria adapted to dynamic partition support, refer to page 239.

Dynamic Partition Support - Supervision

Providing VSE/POWER Storage Resources

The section "Fixable- and Virtual-Storage Requirements" on page 44 provides detailed estimates about SETPFIX and partition Getvis storage consumption in the VSE/POWER partition. When planning to provide support for n dynamic partitions within VSE/POWER, use the following rule of thumb for storage requirements per dynamic partition (SLI support excluded):

	SETPFIX LIMIT storage	Partition GETVIS storage
per spooled reader (1 per partition)	0.8KB	1.0KB + DBLK value
per spooled writer- according to * $$ LST/PUN statement or actual I/O request (up to 28 per partition)	0.8KB	1.0KB + DBLK value

Figure 12. Storage Requirements per Supported Dynamic Partition within IBM VSE/POWER Partition

Limitations During Allocation

External Limitations: Certain external specifications limit the number of dynamic partitions allocated simultaneously by VSE/POWER. These are:

- Total number of partitions which can be generated (NPARTS)
- Amount of address space available (VSIZE)
- Amount of System Getvis Space available (IPL SVA command)
- Maximum number of active partitions per class as defined in DTR$DYN*n*.Z

When one of these limits is reached during an attempt to allocate a dynamic partition, VSE/POWER will suspend allocation. It will resume the allocation attempt as soon as a job running in a dynamic partition terminates whereupon space or partition resources are made available again.

However, when allocation of dynamic partitions fails due to lack of address space and static partitions are unbatched and deallocated by operator intervention, then dynamic allocation is not resumed. In this case, use the PLOAD DYNC command once more.

VSE/POWER Limitations: The following factors influence the number of active dynamic partitions:

- Number of jobs existing in the reader queue which are planned for execution in dynamic partitions.
- Number of dynamic classes enabled for dynamic partition scheduling. You may disable a dynamic class temporarily by using the PVARY DYNC command.
- The size of Getvis storage assigned to the VSE/POWER partition.
- The size of SETPFIX fixable storage assigned to the VSE/POWER partition.

When the Getvis or SETPFIX limits are reached during the process of starting an allocated dynamic partition internally, VSE/POWER suspends allocation for up to 10 seconds or until the next job running in a dynamic partition terminates.

Failure of Allocation

Correctable Failure of Internal Services: The messages 1Q6DA or 1Q6FA during the period of dynamic partition allocation that is, VSE/POWER starting and z/VSE activation indicate a correctable allocation failure of internal services. In this case, VSE/POWER *disables* the dynamic class in question internally to avoid recursive failure. After removal of the cause, the operator can re-enable the class.

Non-Correctable Failure of Internal Services: Whenever internal supervisor services fail to load a dynamic class table, to enable/disable a dynamic class, or to allocate/deallocate a dynamic partition, the failure is indicated by message 1QB5I and 1QZ0I with RC=0022. VSE/POWER writes a dump to the dump sublibrary, ignores the failing request, and continues processing.

Tracking of Allocation Events

The VSE/POWER statistics status report shows the resource consumption for the dynamic partition allocation. You may use the PDISPLAY STATUS command or the 'PEND' command to obtain the items listed under the

```
DYNAMIC PARTITION SCHEDULING STATISTICS
```

section, as demonstrated in the example of Figure 7 on page 46.

All reasons for allocation failure listed under this section are described by the corresponding 1Q3FI message, which is issued when the failure occurs. Please study the explanation, system action and programmer response of message 1Q3FI in advance.

Hiding the Execution Preparation Phase

When a VSE/POWER job is selected for execution in a static partition, its disposition changes from D or K to '*' and remains in execution until the job is finally deleted from the reader queue or is requeued with disposition L.

When a job has been selected by a dynamic class, various preparation steps have to be done until the actual job execution phase starts in a dynamic partition; (see "Static Versus Dynamic Partition Support" on page 132). It is possible that these preparation steps fail occasionally (see "Limitations During Allocation" on page 139 and "Failure of Allocation"), and the already selected job must be requeued with its original disposition D or K.

To make internal differences in job selection for static versus dynamic partitions transparent to operators and programmed interfaces, which follow up job scheduling using the PDISPLAY RDR command, the 'execution preparation' phase for a dynamic partition is not made visible in any display of the reader queue. At the same time, the PDISPLAY A command identifies a VSE/POWER execution reader task which processes the selected job in the dynamic partition. When the dynamic partition is started successfully, its selected VSE/POWER job will display disposition '*'. The 'execution preparation' phase is offered only in a fixed format queue display of a spool-access support CTL request. See also "Job Scheduling Aids" on page 121 for more details.

Specific Devices

This section discusses the use of the IBM 4248 printer, of card 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 ID 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 ID with that of the mounted band. If the IDs match, processing continues. If they do not match, VSE/POWER

  1. Issues a 'band-needed' message to the operator.
  2. Causes the ID 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 mounted, VSE/POWER issues a forms-mount message. In addition, the required form is displayed on the printer, provided the MSG operand has been specified for the PSTART command.

  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 "Support of the IBM 4248 Printer" in *VSE/POWER Application Programming*.

Note: The described 4248 functional support applies also to 6262 printers added as 4248 type (ADD cuu,4248,EML).

Using Card Devices

A multipurpose card device 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 * $$ RDR statement, then SYSIN records can be read only by an application program that reads from a card reader specified at partition startup as spool device. However, if you specify the DSKTIN operand in the * $$ JOB statement, your program can use DTFDU [10] 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 z/VSE or VSE/POWER job or it may contain several such jobs. Jobs with no class specification in their * $$ JOB statement or jobs for which no * $$ CTL statement is in effect, are put into class B.

10. DTFDU = define the file for diskette unit in an assembler program; a diskette I/O file declaration/definition for a compiler.

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 * $$ JOB statement. The * $$ 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 to the spool file exactly as read. The records can be read only by a user program reading from a diskette I/O unit defined in an * $$ RDR statement. DTFDU or DTFDI [11] can be used to access these files.

Several files per diskette volume may be specified by several * $$ RDR statements in the input stream.

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.

11. DTFDI = define the file as device-independent in an assembler program; a device-independent file declaration/definition for a compiler.

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 //&rbl;EXTENT statements is not required if the logical unit was assembled into the DTFDU. One EXTENT statement is required for every 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 ID 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 460.

CPDS Record Spooling and Page Counting

The Print Support Facility (PSF/VSE) program, for example, supports Advanced Function Printing (AFP) on laser printers and makes use of special Composed Page Data Stream (CPDS) records (also called AFP records) to carry information on setup and layout of pages to be printed. CPDS records are identified by a X'5A' Channel Command Word (CCW) operation code, and their data part is called a Structured Field. According to *PSF Data Stream Reference* (SH35-0073), the 3 byte coded Structured Field Identifiers, for example BEGIN PAGE (BPG), are used to interpret the information of CPDS records correctly for printing of image data.

Spooling of CPDS Data

VSE/POWER CPDS list queue entries may be of unique record type or they may appear as a mixture of record types, namely:

- CPDS - X'5A'-coded Structured Fields only
- MCC (machine command code line data) mixed with CPDS
- ASA (ASA operation coded line data) mixed with CPDS

As soon as a list queue entry houses one or more CPDS records, the output of the PDISPLAY LST,...,FULL=YES command indicates CPDS record format (RF=CPDS). VSE/POWER's own local list task (PSTART LST,cuu) does *not* select CPDS-queue-entries for printing, but they can be passed for printing via Spool-Access Support (to PSF/VSE for example).

VSE/POWER supports the *creation* of CPDS queue entries through:

- The Spool-Access PUT-Output interface for unique or intermixed MCC, ASA, and CPDS records
- The standard execution of jobs for spooling of output by either
  - physical IOCS (you build your own CCW chain) to any spooled printer accepted by VSE/POWER (see "PRINTERS: Defining Devices for List Output" on page 495) with a maximum record length of 32K-1, or
  - logical IOCS (you use the BAM DTF interface to a spooled PRT1 printer (using DTFPR with CTLCHR=YES and RECFORM=UNDEF) with CCW op-code x'5A' and a maximum record length of 32K-1.

  This process is actually intended for spooling of MCC-coded line data to be printed on pages, whose setup is described by a Forms Control Buffer (FCB or

LTAB). Therefore, during spooling, VSE/POWER informs the executing program by 'posting', whenever Channel 9 or 12 position within the FCB has been reached.

However, when spooling CPDS only or MCC with CPDS records, then for every CPDS record
- no FCB or LTAB page controlling or Channel posting is done;
- instead, the 'line position on page' is reset to 'start of page' for potential MCC-type records to follow.

VSE/POWER supports the *retrieval* of CPDS queue entries through the Spool-Access GET-Output interface as implemented by PSF/VSE. When a queue entry is being retrieved (Disp='*'), then the PDISPLAY LST,... presents the 'remaining' number of pages/lines 'still to be processed' for CPDS queue entries as well.

VSE/POWER supports the *transportation* of CPDS queue entries via
- PNET transmission and receiving, and via
- POFFLOAD SAVE and LOAD.

Page Counting for CPDS Queue Entries

Pre-6.3 versions of VSE/POWER used to increment the line count per spooled CPDS record, but the total page count of a CPDS queue entry was set to '1', because the 'page information' of a CPDS record was not interpreted at all. Therefore some applications have inserted an MCC Skip-to-Channel-1 record before a CPDS Begin Page record in order to force VSE/POWER to increment the page count.

Starting with VSE/POWER 6.3, a common routine derives the page count for CPDS records from the following Structured Field Identifiers:
- BPG - Begin Page, actually start new page
- IDM - Invoke Data Map, to provide page-layout information
- IMM - Invoke Media Map, also to provide page-layout information.

Their sequence and mixture with non-CPDS records increments the page count, so that it becomes an **acceptable approximation** of the actual number of pages printed later on for this queue entry by the Print Support Facility (PSF/VSE). An 'approximation' only, because detailed and updated IDM/IMM-information may not be available before real print time. Note that even existing output streams with inserted Skip-to-Channel-1 records will yield an acceptable page count.

Since page counting for CPDS entries has become meaningful, the RBS segmentation and RBC checkpointing operands of the * $$ LST statement can also be used, because these functions operate on page boundaries. However, it is not recommended to use RBS segmentation for queue entries that contain CPDS records (PDISPLAY ,..., FULL=YES with RF=CPDS). When printed by PSF, insufficient information on page arrangement may be available when starting a 2nd, 3rd, etc. segment, leading to misaligned printout. To avoid problems, VSE/POWER segments - when the RBS page has been reached - only at the subsequent SKIP-to-CH1 (X'8B') request. That may result in a segment page count equal or possibly *higher* than the specified RBS value.

CPDS Page Counting on Related Systems

The page count evaluated for a CPDS list queue entry on a VSE/POWER 6.3 or follow-on release may be changed when:

- Migrating downward with POFFLOAD BACKUPxx/SAVExx to a pre-6.3 VSE/POWER system.
- Transmitting the entry via PNET to a pre-6.3 VSE/POWER node.
- Transmitting the entry via PNET to VM-RSCS, OS/390-JESx, or AS/400 nodes, which use their own CPDS page counting routines.

Important Specifications for Output Spooling

In the following, the relationship between certain specifications for VSE/POWER Output Spooling are explained.

For simplicity reasons, this is done only for LST queue output; but the same rules, considerations and specifications are also valid for PUN queue output.

Specifications for OUTPUT Spooling

During execution of a job in a 'VSE/POWER controlled partition', output is produced using the following rules:

1. In your program you specify the logical unit to which the output will be written. For example, by:

   ```
   DTFPR DEVADDR=SYSLST
   ```

2. With the // ASSGN statement of your job, you direct the output from the logical unit to a physical unit (also printer) of your choice. For example:

   ```
   // ASSGN SYSLST,F9E
   ```

   Note: For spooled printers, you should *not* use generic assignments such as

   ```
   // ASSGN SYSnnn,PRINTER
   // ASSGN SYSnnn,PRT1
   ```

   that specify a device class or type, because it may result in multiple assignments to the same spooled device address and hence produce unpredictable spooling confusion.

3. If the output should be spooled by VSE/POWER, this physical unit must be specified in the PRINTERS=cuu,cuu.... autostart statement for the corresponding partition. For example:

   ```
   PSTART BG,...
   READER=...
   PRINTERS=...,...,F9E,...
   PUNCHES=...
   ```

   Note: For a display of the spooled printers established for a certain partition, use the PDISPLAY SPDEV, ... command described in "Format 15: Displaying Information About Spooled Devices" on page 245.

4. With the * $$ LST statement, you can specify several output attributes for your output entry (like JNM, DISP, PRI, etc.). With the LST=cuu operand of the * $$ LST statement, you tell VSE/POWER that this * $$ LST statement should be taken for the output spooled to the physical unit 'cuu', which you specified in the LST=cuu operand. For example:

   ```
   * $$ LST JNM=...,...,LST=F9E,...
   ```

   Remember the following default rules and support conditions for the * $$ LST statement:

   a) * $$ LST statement without LST=cuu operand

If the LST=cuu operand is omitted in the * $$ LST statement, the attributes specified in this statement are taken for the output spooled to that printer device which is specified **first** in the sequence of PRINTERS=**cuu**,cuu,... .

b) no * $$ LST statement at all

If you are spooling output to a printer device, for which no * $$ LST statement is provided, then this output entry gets default output attributes as described for the DISP=, CLASS=, and PRI= operands of the * $$ LST statement (usually DISP=D, CLASS=A, PRI=3).

c) track assignment of default attributes

Whenever default output attributes are assigned to any output stream spooled by your job, you may request VSE/POWER to warn you by console message

```
1Q8CI DEFAULT OUTPUT VALUES USED FOR jobname jobnumber, SPOOLED DEVICE cuu
```

To request such notification of the central operator, just enter the NTFY=(*,R000) operand in the * $$ JOB statement of the job in question.

Whenever the * $$ LST statement in question does not yield the expected output attributes, place the following extensions before this LST statement:

a. LISTIO ASSGN, to log all current assignments
b. EXEC DTRIATTN, to request logging of spooled devices

The extended job would appear as follows:

```
* $$ JOB JNM=SPOOLTST,...,NTFY=(*,R000)
// JOB SPOOLTST
 .
 .
LISTIO ASSGN
// EXEC DTRIATTN,PARM='PDISPLAY SPDEV'   OR SUBSELECT BY ',PART,XX'
* $$ LST DISP=K,LST=00E
 .
 .
/*
/&
* $$ EOJ
```

d) * $$ LST statement skipped conditionally

A * $$ LST statement may be ignored by VSE/POWER, i.e., specified output attributes do not become effective, when conditional job control processing is searching for a GOTO label. For details, refer to “Interaction with z/VSE Conditional Job Control Language” on page 108.

How to Link the OUTPUT Specifications

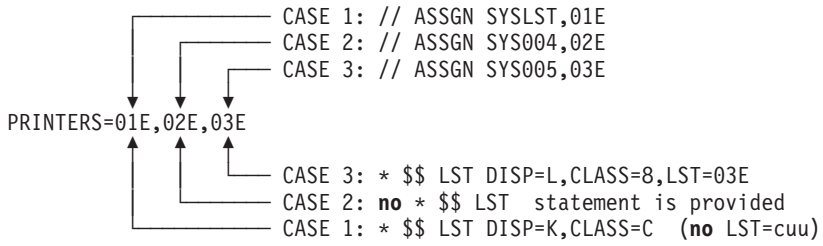
Respect the following rule:

Your output obtains the desired * $$ LST attributes only if the spooled printer device ...

```
- selected by the ASSGN statement and
- addressed by your * $$ LST statement
```

... is the same!

In the following there are three cases explained:



Case 1:
- Your program writes output to SYSLST
- SYSLST is assigned to 01E
- 01E is specified as the *first* printer in PRINTERS=01E,02E,03E chain.
- The LST=cuu operand is omitted in the * $$ LST statement, therefore this statement applies to the *first* of PRINTERS=01E,02E,03E, namely to 01E.
- **Result:** The output entry spooled to 01E gets the specified output attributes DISP=K,CLASS=C.

Case 2:
- Your program writes output to SYS004
- SYS004 is assigned to 02E
- 02E is specified in PRINTERS=01E,02E,03E
- **no** * $$ LST statement is provided at all
- **Result:** The output entry spooled to 02E gets the default output attributes as described for the DISP=, CLASS=, and PRI= operands of the * $$ LST statement (usually DISP=D, CLASS=A, PRI=3).

Case 3:
- Your program writes output to SYS005
- SYS005 is assigned to 03E
- 03E is specified in the PRINTERS=01E,02E,03E .
- By the LST=03E operand, the * $$ LST statement applies to the output entry spooled to printer 03E.
- **Result:** The output entry spooled to 03E gets the specified output attributes DISP=L,CLASS=8.

Specification Errors that Lead to the 'Ignored * $$ LST Statement' Symptom

For the following cases we assume that your program produces output for SYS004:

1. **Case 1:** The LST=cuu operand is omitted on the * $$ LST statement.

   Specifications:

   ```
   - PRINTERS=FEE,F9E,F8E
   - * $$ LST JNM=...,DISP=K,PRI=1
   - // ASSGN SYS004,F9E
   ```

   Result: The output entry spooled to F9E obtains only default output attributes. The provided * $$ LST statement is valid for device FEE because of the missing LST=F9E specification.

2. **Case 2:** The output is directed to another print device than specified in the LST=cuu operand.

   Specifications:

   ```
   - PRINTERS=FEE,F9E,F8E
   - * $$ LST JNM=...,DISP=K,CLASS=X,LST=F8E
   - // ASSGN SYS004,F9E
   ```

Result: The output entry spooled to SYS004 obtains only default output attributes because the * $$ LST statement is valid for F8E (LST=F8E), but output is spooled to F9E.

Tape Processing with VSE/POWER

The following VSE/POWER tape handling functions are available:

Table 6. Tape Processing with VSE/POWER

User Task	Command or JECL Statement	See Page
Spooling LST or PUN output to tape	* $$ LST DISP=T * $$ PUN DISP=T	159
Offloading spool entries via backup, pickup, or save	POFFLOAD BACKUP POFFLOAD PICKUP POFFLOAD SAVE	155
Restoring from offload or spool output tapes	POFFLOAD LOAD POFFLOAD SELECT	157
Printing or punching offload or spool output tapes	PSTART LST,uraddr,X'tapeaddr' PSTART PUN,uraddr,X'tapeaddr'	160
Displaying contents of offload or spool output tapes	PDISPLAY entry,TAPE=tapeaddr	161
Reading jobs from SYSIN tape for input spooling	PSTART RDR,tapeaddr	161
Saving VSE/POWER account file to tape	PACCOUNT tapeaddr	163

Tapes are processed in three stages:

1. The mount and unmounting of tape(s) by OPEN (mount single tape) and CLOSE (unmount single tape) and EOV (End-Of-Volume multiple tape processing - unmount last tape and mount a new tape). This is performed either by:
   - VSE/SAM (Sequential Access Method), or
   - Native VSE/POWER tape support.
2. Labeled or unlabeled tape processing – the writing, reading or checking for the presence of tape labels and tape marks during OPEN/CLOSE/EOV processing.

   Labeled or unlabeled tape processing is possible, supported by either VSE/SAM (Sequential Access Method) or native VSE/POWER tape support, depending on the tape function.
   - VSE/POWER **native** [12] support – always displays a tape label (when present) on the console and requests the operator to verify what action is to take place. There is no dependency on z/VSE Access Control protection of labeled tapes. No labels are created and/or written to tape.
   - VSE/SAM – performs label checking based on the function (read or write) and the `// TLBL` defined by the user; the operator is prompted only if an error occurs. If z/VSE Access Control protection of labeled tapes is occurring, the operator may not reply IGNORE to error conditions.

   For information on the content of tape labels, see *z/VSE System Macros Reference*, "Standard and Non-Standard Labels". For information on the positioning of tape labels, see *z/VSE System Macros User's Guide*, "Data Management Concepts".
3. Reading/writing of spool or accounting data from/to tape. This is done entirely by VSE/POWER tape support.

Labeled Tape Support

Depending on the tape function, VSE/POWER can:

- **Write** and **read** labeled tapes through VSE/SAM tape support, or
- **Read** labeled tapes through **native** VSE/POWER tape support

Labeled Tape VSE/SAM Processing

The VSE/POWER tape support via VSE/SAM provides:

- Increased tape spooling capacity since the spool output may now exceed the capacity of a single tape (multi-volume support for a single-file queue entry)
- Improved system security and management since labeled tapes permit control of system and user data.
- VSE/SAM interfaces and exits which might be exploited by a Tape Management System (TMS).

See "Layout of VSE/POWER Tapes" on page 527.

Invocation: The VSE/POWER tape support via VSE/SAM is initiated for commands or JECL statements through the operands LTAPE=YES and/or TLBL=, or for the `PACCOUNT tapeaddr,,tlblname` command. The VSE/SAM interface uses a DTF macro for OPEN/CLOSE/EOV processing (DTFPH for PACCOUNT, DTFMT otherwise).

VSE/POWER Access of Tape Label Information: VSE/SAM requires a tape label to be defined in the VSE/POWER partiton labeled area or standard label area when required. See the // TLBL statement in the manual *z/VSE System Control Statements*.

Whenever the VSE/POWER operand TLBL=tfilename has been specified, either in a VSE/POWER command or JECL statement, a `// TLBL tfilename...` statement must be available to the VSE/POWER partition either in the VSE/POWER startup job JCL, or via the standard label area by executing in BG for example the following statements:

```
// OPTION STDLABEL=ADD
// TLBL tfilename,....
/&
```

Note: It is not sufficient to specify the `// TLBL` statement as a user label within the jobstream which contains the `* $$ LST/PUN DISP=T` statement.

Single File Support only: VSE/POWER labeled tape support via VSE/SAM is single file, multi-volume. This is because:

- Multi-file labeled tapes (multiple data areas on tape each beginning with a HDR1 label and ending with an EOF1 label) are not allowed by z/VSE Access Control
- Furthermore, since the VSE/POWER data area between the tape labels is non-standard (tape marks used to separate the spool entries), Tape Management Systems are prevented from properly locating additional tape labeled files.

This means that the no-rewind option (NOREW or REW=NO) for tape commands is **not** allowed, nor is multiple job output spooling to tape performed (which is

12. Native support is executed automatically if none of the VSE/SAM operands TLBL= or LTAPE= are specified in the given VSE/POWER command or JECL statement.

allowed for unlabeled tape spooling). Processing always begins at the tape load point (followed by an initial rewind if necessary).

Labeled Spool Output Tapes are Unloaded: Labeled spool output tapes produced at job execution time (DISP=T) tapes will be unloaded to avoid producing multi-file labeled tapes when any of the following occur:

1. The end of the VSE/POWER job.
2. A `* $$ LST/PUN` input reader statement is read for the same printer or punch device
3. A IPWSEGM KEEP=NO macro occurs for the same output unit
4. A SEGMENT macro occurs for the same output unit.

Therefore, normally labeled job execution spool tape(s) contain only a single output entry. However, the labeled spool output may be separated into related entries when:

- The macro IPWSEGM KEEP=YES is used to segment output
- Either of the macros LFCB or SETPRT is called during output spooling.

In such cases, a trailer queue record followed by a tape mark followed by a new header queue record are written to identify the separate spool entries, but no tape unload occurs and further spooling is written to the same tape until either the tape becomes full (then another is caused to be mounted) or any of the previous conditions occur to cause the tape to be unloaded. A PDISPLAY of the tape(s) will indicate the presence of individual spool entry(s) corresponding to the output spooled prior the calling of the macro. As usual for these macros, the output following each macro has its own jobnumber.

Message Text Meaning Of 'VOLUME=nnn' Or 'VOL=nnn': When in a message the text 'VOLUME=nnn' or 'VOL=nnn' appears, this refers to the labeled tape volume number of the spool entry being processed. Although a labeled tape function may produce several tapes, the given message text will refer only to the volume(s) concerned with the indicated spool entry. This volume number can always be displayed by using the PDISPLAY command.

Displaying Of Queue Entry Volume Number In Messages - Restriction: VSE/POWER can only display a maximum of 126 volumes individually in messages, for example "VOLUME=nnn" or "VOL=nnn". If the number exceeds 126 then the number displayed will be "***", for example "VOL=***" - this means that the actual volume number is 127 or greater. When the number of tape volumes exceeds 126, then the operator must take steps to properly identify the proper tape sequencing.

Labeled Tape Sequence Checking: When a set of tape volumes are produced with a given tape label, the operator will have some freedom in choosing the sequence order of mounting the tapes when the tapes are being read.

If the tape label specified for reading the tape indicates to do sequence checking, then VSE/SAM will enforce a sequential checking of the tapes, which is generally recommended (// TLBL filename,fileid,,,1). However, in some cases this may have disadvantages:

- When displaying the contents of a single labeled spool tape and one is not interested in the prior sequenced tapes belonging to the same label
- When performing a POFFLOAD SELECT to locate a single spool entry, one is not interested in the prior sequenced tapes

- When processing spool output on labeled tape and the operator does not want to begin with the first tape of entries.
- When processing spool output on labeled tape and the operator wants to perform PRESTART to have printing/punching restart at a spool record on a previous tape.

For the above cases, it is recommended that the tape label not specify to perform labeled tape sequence checking. VSE/POWER will in any case automatically check for the proper tape mounting sequence only when necessary, when a spool entry spans more than one tape volume.

When processing spool output on labeled tape, and a tape is mounted which contains an incomplete spool entry at the beginning (because the operator did not want to begin with the previous tape volume(s)) then VSE/POWER will skip the incomplete entry as indicated by the message:

```
1Q7DI TAPE BEGINS WITH INCOMPLETE SPOOL ENTRY. SKIPPING TO NEXT ENTRY
      ON device FOR cuu|task
```

When processing has reached the end of tape, it continues until the end-of-file label EOF1 has been read, as would be the case normally should the operator had mounted the first volume of the label.

Aiding Operator to Identify Multi-volume Labeled Tapes: In the case that a spool entry is being written to tape, and the data cross a tape boundary (multi-volume support) VSE/POWER displays the message:

```
1QA2I VSE/POWER MULTI-VOLUME TAPE COMPLETE FOR jobname jobno jobsuffix queue
      VOLUME=yyy ON dev FOR task,cuu
```

The message text either displays VOLUME=nnn, meaning a tape has become full for the indicated spool entry and is being rewound and unmounted. or it displays VOLUME=nnn(LAST), meaning a spool entry spanning two or more multi-volume tapes has been completely written to tape.

The message is to aid the operator in manually labeling tape volumes if desired. Since VSE/POWER volume number sequencing refers to the individual spool entry (and not to the VSE/SAM tape label) then manually labeling the tape can aid the operator in locating tapes to be mounted during later processing (see message 1QG0A).

The information can also be obtained from the PDISPLAY TAPE command, but since the information is located at the end of the spool entry on tape, the command reply may be delayed for a somewhat long time till the end of the entry is located, therefore such a message may save the operator time in locating a given tape later on.

Labeled Tape Native VSE/POWER Processing

Native support is executed automatically if none of the VSE/SAM operands TLBL= or LTAPE= are specified in the given VSE/POWER command or JECL statement.

VSE/POWER native labeled tape processing can only read labeled tapes but not produce them. It is used for processing SYSIN tapes or displaying labeled spool tapes.

Native labeled tape support does not use the tape labels `// TLBL tfilename...`. Instead it displays tape label information with the message 1QB9A and requests the operator to indicate which system action is appropriate..

Unlabeled Tape Support

VSE/POWER processes and produces unlabeled tapes either via VSE/SAM or VSE/POWER native tape support.

Unlike labeled tape processing, with unlabeled tape the ″no rewind″ operand may be indicated in the commands (NOREW or REW=NO). This permits more efficient usage of individual tape capacity.

However, when performing POFFLOAD BACKUP|PICKUP|SAVE, any individual spool entries which exceed the capacity of a single tape cannot be processed and is flagged with message 1Q7EA.

Should the end of tape capacity occur during job execution tape spooling then the output is segmented and a new tape is caused to be mounted.

See “Layout of VSE/POWER Tapes” on page 527.

Unlabeled Tape VSE/SAM Processing – Invocation

The VSE/POWER VSE/SAM tape support is initiated for commands or JECL statements through the operand LTAPE=NO, or for the `PACCOUNT tapeaddr` command. The VSE/SAM interface uses a DTF macro for OPEN/CLOSE/EOV processing (DTFPH for PACCOUNT, DTFMT otherwise).

Unlabeled Tape Native VSE/POWER Processing – Invocation

The VSE/POWER native tape support is initiated for commands or JECL statements whenever tape support is invoked without the VSE/SAM operands.

Unlabeled Spool Output Tapes Are Not Unloaded

During spooling of job output to tape using the `* $$ LST/PUN DISP=T` statement, the tape is not normally rewound after a job unless the tape becomes full. This permits additional job spool entries to be stored to tape.

Processing for the Offload-Queue Function

Labeled or unlabeled tape processing is possible. VSE/SAM supports both labeled and unlabeled tape. Native VSE/POWER tape support is available for unlabeled tape processing only.

A journal listing the spooling entries written to tape is created by default and placed in the LST queue with the name $OFJ*nnnn* where 'nnnn' is the jobnumber of the journal LST entry. For details, see 337.

POFFLOAD BACKUP|PICKUP|SAVE Function

POFFLOAD BACKUP: This function 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 to the tape mounted on the drive at the address 280. Queue entries saved by a BACKUP operation are retained in their queue unchanged.

POFFLOAD SAVE: The SAVE function is performed in parallel with normal processing of queue entries and output spooling.

For example, the command

```
POFFLOAD SAVE,LST,280,,C
```

causes dispatchable list queue entries of class C to be copied to the tape mounted on the drive at the address 280. The 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.

POFFLOAD PICKUP: The PICKUP function is generally performed in parallel with normal processing of queue entries and output spooling (as for SAVE).

For example, the command

```
POFFLOAD PICKUP,LST,280,,C
```

causes all (as for BACKUP) list queue entries of class C, regardless of their assigned disposition, to be copied to the tape mounted on the drive at address 280. Queue entries saved by a PICKUP operation are retained in their queue unchanged.

Note: An exception to the paralled processing by PICKUP occurs when an entry, either being processed by other tasks (DISP=*) or resides in the 'wait for run' subqueue, is copied to tape: only for this period is the entire queue blocked from access by other tasks (as for BACKUP).

Unlabeled Tape Processing for BACKUP/PICKUP/SAVE: If the mounted tape volume is at its load point, the tape is checked for the presence of standard labels. If standard labels exist the operator is prompted whether to:

- Reject the volume since the tape is labeled, or
- Continue processing with the mounted tape and ignore the tape label.

If the label is to be ignored, it is overwritten.

If processing is done by native VSE/POWER, the operator is prompted by the message:

```
1QB9A  tapeaddr, HEADER: filelabelcontents ...
```

If processing is done by VSE/SAM (Sequential Access Method), the operator is prompted by the message:

```
4125D VOL1 LABEL FOUND ...
```

If the output tape is not at its load point, it is rewound unless the no-rewind option is specified. 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, VSE/POWER discards any partially completed queue entry at the end of the tape, closes the tape and informs the operator by a message that a new tape must be mounted to process the remaining queue entries,

including the one 'in process' at end-of-volume. When all queue entries of the specified classes have been written to tape, the task stops, and VSE/POWER issues task-complete messages.

Warning: If saving of a big entry starts at the begin-of-tape, but the entry does not fit on this tape, no segmenting takes place. Instead, VSE/POWER skips the entry with message 1Q7EA. It might be helpful to stop the saving function and retry it with a higher tape density, or to use labeled tape processing instead.

Labeled Tape Processing for BACKUP|PICKUP|SAVE: VSE/SAM checks the tape for the presence of standard labels, beginning by checking for the presence of a volume VOL1 label, and if none found will create one if indicated by the operator. Then VSE/SAM will perform the indicated volume serial number and label checking as specified by the tape label `// TLBL` indicated in the TLBL= operand of the command. If VSE/SAM finds the labels to be in order, or the operator indicates to ignore an error, it updates the labels as required (date, volume number, etc). VSE/POWER then position the tape following the labels and proceeds to write the spool output to tape.

If end-of-volume occurs, (reflective marker was encountered) any partially completed queue entry at the end of the tape is retained and closed, the operator is prompted to mount the next tape (multi-volume support), and processing continues with the remaining data of any partially written queue entry being continued on the next tape volume until no further entries remain to be processed. VSE/POWER executes a VSE/SAM FEOV macro to cause a fresh tape to be mounted. VSE/SAM will issue the message:

```
4140A NO ALTERNATE DRIVE ASSIGNED ...
```

to which the operator must mount a new volume and issue the response 'NEWTAP'. At the end of the previous tape a trailing queue record will be written with the volume number for the queue entry being continued on the next tape. The next tape will begin also with a header queue record following the tape labels, but with the entry volume number incremented by one. These volume numbers are independent of those used by VSE/SAM in its standard labels and concern only the continued queue entry being spooled.

The labels of the next tape are checked as previously, and once finished, VSE/POWER continues tape spooling.

When the spooling entries have been written to tape, VSE/POWER releases the tape assignment with the tape being rewound and unloaded.

POFFLOAD LOAD|SELECT Function

POFFLOAD LOAD: Restores (load) queue entries written to tape by using the POFFLOAD BACKUP/PICKUP 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, where those queue entries are restored with DISP=D. .

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.

VSE/POWER jobs that are restored to queues regain their original disposition, except when you specify HOLD=YES in your POFFLOAD LOAD/SELECT command. Then you may suppress uncontrolled processing of reloaded jobs.

If the queue ID on the tape does not match the ID given in the POFFLOAD LOAD command, then VSE/POWER rejects the tape and stops the task. You are informed of this by a message.

POFFLOAD SELECT: Reloads 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.

Unlabeled Tape Processing for LOAD|SELECT: If you specify NOREW in the POFFLOAD LOAD command, ensure that the tape is correctly positioned (in front of the queue record of the next queue entry you want to address--see "Layout of Spool Tapes" on page 530 and following the preceeding tape mark(s) for the queue entry) 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 by message:

```
1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON cuu
```

If multiple volumes are used, an extra POFFLOAD LOAD command must be given for every volume, although the message OFFLOAD SUCCESSFULLY COMPLETED appears already after the first volume has been reloaded.

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.

Labeled Tape Processing for LOAD|SELECT: If multiple volumes are used, the operator will be prompted by the message:

```
4140A NO ALTERNATE DRIVE ASSIGNED ...
```

to which the operator must mount a new volume and issue the response 'NEWTAP'.

The load operation ends if end-of-file EOF1 label is recognized as indicated by the message:

```
1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON cuu
```

Note: When label checking of the labeled tape to be loaded should be bypassed, specify a 'default' tape label as described in "Displaying Labeled Spool Tape via VSE/SAM" on page 161.

Stopping a POFFLOAD Operation

A POFFLOAD operation can be stopped immediately (with a possible loss of data) or only after the currently processed queue entry has been completely saved or restored. You do this by submitting a PSTOP, or PSTOP EOJ respectively, 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, BACKUP, or PICKUP 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.

Spooling Output to Tape

You can direct VSE/POWER to route printer or punch output to tape by specifying DISP=T and/or TADDR= in the * $$ LST/PUN statement.

Labeled or unlabeled tape processing is possible. VSE/SAM supports both labeled and unlabeled tape. Native VSE/POWER tape support is available for unlabeled tape processing only.

Unlabeled Tape Processing

If the mounted tape volume is at its load point, the tape is checked for the presence of standard labels. If standard labels exist the operator is prompted whether to:

- Reject the volume since the tape is labeled, or
- Continue processing with the mounted tape and ignore the tape label.

If the label is to be ignored, it is overwritten.

If processing is done by native VSE/POWER, the operator is prompted by the message:

```
1QB9A  tapeaddr, HEADER: filelabelcontents ...
```

If processing is done by VSE/SAM (Sequential Access Method), the operator is prompted by the message:

```
4125D VOL1 LABEL FOUND ...
```

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. The operator is prompted to mount a new tape. If the new tape is a labeled one, handling proceeds 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.

Labeled Tape Processing

If the mounted tape volume is at its load point, VSE/SAM checks the tape for the presence of standard labels, beginning by checking for the presence of a volume VOL1 label, and if none found will create one if indicated by the operator. Then VSE/SAM will perform the indicated volume serial number and label checking as specified by the tape label `// TLBL` indicated in the TLBL= operand of the `* $$ LST/PUN DISP=T` statement. If VSE/SAM finds the labels to be in order, or the operator indicates to ignore an error, it updates the labels as required (date, volume number, etc). VSE/POWER then position the tape following the labels and proceeds to write the spool output to tape. Otherwise the operator may cancel the tape operation, and VSE/POWER will instead spool the output to disk (as indicated by the message 1Q5KI).

Tape spooling always proceeds from the tape beginning following the tape labels.

If end-of-volume occurs (reflective marker was encountered), VSE/POWER will execute a VSE/SAM FEOV macro to cause a fresh tape to be mounted. VSE/SAM will issue the message:

```
4140A NO ALTERNATE DRIVE ASSIGNED ...
```

to which the operator must mount a new volume and issue the response 'NEWTAP'. At the end of the previous tape a trailing queue record will be written with the volume number for the queue entry being continued on the next tape. The next tape will begin also with a header queue record following the labels, but with the entry volume number incremented by one. These volume numbers are independent of those used by VSE/SAM in its standard labels and concern only the continued queue entry being spooled.

The labels of the next tape are checked as previously, and once finished, VSE/POWER continues tape spooling.

When spooling to tape for a job is complete, VSE/POWER releases the tape assignment with the tape being rewound and unloaded.

Printing or Punching Output from Tape

To have VSE/POWER print or punch spool output contained on tape you must have issued a PSTART LST/PUN,cuu,tapeaddr command.

Labeled or unlabeled tape processing is possible. VSE/SAM supports both labeled and unlabeled tape. Native VSE/POWER tape support is available for unlabeled tape processing only.

When VSE/POWER is reading list or punch output from tape it checks that the mounted tape was created either by spooling output to tape (DISP=T) or by a POFFLOAD BACKUP/SAVE function. This involves a tape rewind operation.

The operator gets a message if a wrong tape was mounted.

Output cannot be printed from the IBM 9346 or 3592 tape — any attempt is rejected by message 1Q7FI

Unlabeled Tape Processing

When end-of-file occurs (double tape marks), the writer task stops. It is the operator's responsibility to ensure that subsequent tapes (if any) are processed, that means reissuing the PSTART command.

Labeled Tape Processing

If an end-of-volume EOV1 label is encountered, VSE/POWER will execute a VSE/SAM FEOV macro to cause a fresh tape to be mounted. VSE/SAM will issue the message:

```
4140A NO ALTERNATE DRIVE ASSIGNED ...
```

to which the operator must mount a new volume and issue the response 'NEWTAP'. Processing continues until the end-of-file EOF1 label is read.

Processing for the PDISPLAY Tape Function

To have VSE/POWER display a tape containing spool entries you must have issued a PDISPLAY TAPE=cuu command. The possibilities for selection and limitation of the entries displayed are the same as for entries on disk. Only individual tapes will be displayed.

Labeled or unlabeled tape processing is possible. VSE/SAM supports both labeled and unlabeled tape. Native VSE/POWER tape support is available for both labeled and unlabeled tape processing.

Displaying Labeled Spool Tape via VSE/SAM

The VSE/POWER PDISPLAY command will in any case display only one tape even though there are more than one tape for the given label. It is recommended to specify a default tape label (without file-id, etc.) and load it into the label area, for example

```
// TLBL DUMMY
```

when entering the command, for example

```
PDISPLAY queue,TAPE=cuu,...,TLBL=DUMMY
```

which enables a display of any tape since no label checking is performed by VSE/SAM.

The tape display will also display the HDR1 tape label, for example:

```
F1 0058 4151I HDR1 LABEL INFORMATION  TLBL1    SYS004=282
   HDR1 MY LABEL FILEID VOLXXX00010001       96232 9723100000000IBMDOSVS
```

The display of spool entry information on a POFFLOAD tape will differ from that of a job execution spool output tape when the entry is continued on two or more multi-volume tapes, that is when VOL=nnn is displayed for the entry:

- The information for each volume of the entry for a POFFLOAD tape will be the same. The total spooling data are displayed, for example page or record count.
- For a job execution spool tape, the data displayed are the running totals of accumulated spooled output when the given tape became full. Only when the last tape (VOL=nnn(LAST)) is displayed are the total spooling information shown.

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.

Labeled or unlabeled tape processing is possible. VSE/SAM and native VSE/POWER supports both labeled and unlabeled tape.

The mounted tape volume is rewound to check for the presence of standard-labels.

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.

Unlabeled Tape Processing

The unlabeled tape SYSIN function supports single and multi-file, single volume tape (tape processing ends when the end-of-tape double tape mark is read).

Note: Unlabeled multi-volume files are not supported.

If processing is done by VSE/SAM (Sequential Access Method), the tape is positioned at its load point, and processing of the 80 byte records proceeds. If standard labels exist, VSE/POWER creates an AUTONAME job (DISP=H) out of the label records and proceeds to the terminating tape marker for the standard labels where it processes the SYSIN data, and upon reaching the ending tape labels will also create an AUTONAME job (DISP=H) of the label records. Processing of the SYSIN records continues until the end-of-tape double tape mark is found, when the task stops.

If processing is done by native VSE/POWER, and standard labels exist, then the operator is prompted by the message 1QB9A, and if indicated by the operator, then labeled tape processing is initiated. Processing of the SYSIN records continues until the end-of-tape double tape mark is found, when the task stops.

Labeled Tape Processing

The processing depends on the access method chosen. The VSE/SAM interface supports a single-file multi-volume SYSIN tape (processing stops when an EOF1 label is read - multiple file tapes are not supported for security reasons). VSE/POWER native code supports single and multi-file, single and multi-volume SYSIN tape.

If processing is done by VSE/SAM (Sequential Access Method), the tape is positioned at its load point, and checking proceeds for the presence of standard labels, beginning by checking for the presence of a volume VOL1 label. Then VSE/SAM will perform the indicated volume serial number and label checking as specified by the tape label `// TLBL` indicated in the TLBL= operand of the PSTART command. If VSE/SAM finds the labels to be in order, or the operator indicates to ignore an error, processing continues. If an unlabeled tape is being processed using labeled tape support the operator will be prompted by the message:

```
4111D NO VOL1 LABEL FOUND ...
```

If processing continues (because the operator indicated to ignore the label error), then the task will end with an error message (due to missing end-of-file labels) and the last processed job may be deleted. If end-of-volume EOV1 label is encountered, VSE/POWER will execute a VSE/SAM FEOV macro to cause a fresh tape to be mounted. VSE/SAM will issue the message:

```
4140A NO ALTERNATE DRIVE ASSIGNED ...
```

to which the operator must mount a new volume and issue the response 'NEWTAP'. Processing ends when the EOF1 label is read.

Note: The user may have multi-file labeled tapes which he processes (or processed earlier) with the native VSE/POWER SYSIN function. The VSE/SAM support is only able to read the first file on the tape(s), but none of the following files.

If processing is done by native VSE/POWER tape support, the tape is positioned at its load point, and checking proceeds of the standard labels. VSE/POWER displays the HDR1 label information in the message 1QB9A and waits for the operator to indicate whether to proceed. If indicated by the operator, processing proceeds until either an end-of-volume EOV1 label is encountered, in which case the operator will be instructed to mount a new volume, or the end-of-file EOF1 label is encountered and the job being read is closed and stored on disk. VSE/POWER then proceeds to check for further tape files, and if none found will terminate the task.

Processing for the PACCOUNT Tape Function

To have VSE/POWER archive the contents of the VSE/POWER account file to tape you must issue a `PACCOUNT tapeaddr` command.

Labeled or unlabeled tape processing is possible via VSE/SAM support.

Tape Library Dataserver 3494

To access a 3494 robot tape volume, the operator need only issue the command:

```
LIBSERV MOUNT,VOL=,UNIT=,PART=
```

indicating the VSE/POWER partition, prior to issuing a VSE/POWER tape command addressing the same tape unit.

Time Event Scheduling

By using time event scheduling operands in the VSE/POWER * $$ JOB statement, you can determine the day and time a job is to be scheduled for processing. The time event scheduling occurs only for jobs in the reader queue that are going to be processed in a partition. The transmission of a job to another node or the printing/punching of an output is not affected by time event scheduling.

However, if a job is transmitted to another system, the due date operands are saved and sent together with the job (within the VSE/POWER section of the network job header record) to the other system.

Thus processing of the job at the destination node can take place according to the due date operands (if the destination node runs at least VSE/POWER 4.1.1). To date, the due date operands are not meaningful for systems other than VSE/POWER.

Time event scheduling operands were initially introduced for one-time processing of a certain job

- on a certain date (1 day), or
- on pre-selected days

With extended support by the DUEFRQ operand, a frequency of processing, i.e., n-time processing of a certain job on a scheduled day has been introduced (see "Scheduling a Job More Than Once Per Day" on page 171). For better understanding, however, first read the rules of one-time processing per day.

Scheduling Operands

If a job is to be scheduled *once*, you specify the time and the date. For example, if a job is to be scheduled on February 10th, 2005, at 1:30 p.m., you specify within the * $$ JOB statement:

DUETIME=1330,DUEDATE=021005

If a job is to be scheduled *repetitively*, i.e., on pre-scheduled days, you can specify that the job is to be scheduled

1. Daily
2. Every weekday (for example, every Monday)
3. Every specific day of every month (for example, every first day)
4. Every specific day of certain months (for example, every first of January, June and December)

Following are some examples of how the operands can be used. If a job is to be scheduled daily at 14 o'clock, you specify within the * $$ JOB statement:

DUETIME=1400,DUEDAY=DAILY

If a job is to be scheduled every Monday at 9 o'clock, you specify within the

* $$ JOB statement:

DUETIME=0900,DUEDAY=MON

If a job is to be scheduled every 10th at 0:15 o'clock, you specify within the

* $$ JOB statement:

DUETIME=0015,DUEDAY=10

For details see Chapter 6, "JECL Statements," on page 397.

Scheduling Rules

When a job gets scheduled for processing, VSE/POWER queues it according to its class and priority into the dispatchable class chain using the rules of today (FIFO). That is, the job that gets dispatchable first, gets processed first (if same priority).

The following scheduling rules apply, if a job 'enters the system in order to be processed':

Rule 1 If just a time, but no day or date has been specified, the job gets scheduled once:
1. On the same day, if the due time is later on that day,
2. Or on the next day, if the due time has already passed.

Rule 2 If a job is supposed to be scheduled just once and enters the system after the specified date, the job is scheduled at once for processing. For evaluation of the processing date, see "Functions Allowing Date Format Changes" on page 127.

For example: if a job is to be scheduled at 10 a.m. on January 11, 2005, and is read in on January 12, 2005, the job is scheduled at once.

Rule 3 If a job is supposed to run every Monday, the job is scheduled at the next occurrence of the due date.

For example: if a job is to be scheduled every Monday at 11 o'clock and enters the system on Monday at 12 o'clock, the job does not get scheduled before the next Monday.

The following examples clarify the expression 'a job enters the system in order to be processed':

- A dispatchable job is read in by the local card reader.
- A dispatchable job is read in by an RJE workstation.
- A dispatchable job is read in from a tape.
- A dispatchable job is submitted from programmed interfaces.
- A dispatchable job is entered from tape via POFFLOAD.
- A dispatchable job is received via network.
- The disposition of a recursive job gets changed from non-dispatchable to dispatchable.
- The disposition of a job for which just time is specified gets changed from non-dispatchable to dispatchable, after the job has initially been read into the non-dispatchable chain.
- The destination of a dispatchable job gets changed in such a way that the job is removed from the transmit queue and added to the local reader queue (for example, by means of the PALTER command).
- A job which should be scheduled more than once according to the due date operands has finished processing and gets queued for the next scheduling.

Now some examples follow where the job does **not** 'enter the system in order to be processed'. Therefore, the scheduling rules do not apply:

- A non-dispatchable job is read in by the local card reader, or any of the above sources.
- The class or priority of a job gets changed.
- The disposition of a job gets changed from non-dispatchable H to non-dispatchable L , or vice versa.
- The disposition of a job gets changed from dispatchable D to dispatchable K, or vice versa.
- The disposition of a job gets changed after recovery from X to dispatchable (for example D or K).
- The destination of a dispatchable job gets changed in such a way that the job remains in the transmit queue, or that the job is moved from the local reader queue to the transmit queue.

Changing the Jobnumber

If a job is to be processed more than once (for example, every Monday), the job gets assigned a new jobnumber every time the due date expires (except for the first time).

Thus the list or punch output produced during the last run always has the same jobnumber as the job that produced the output. (However, this is not the case if the job includes more than one * $$ LST statement or * $$ PUN statement.)

Disposition Handling

When the due date operands specify that the job is supposed to be processed **more than once** (for example, every Monday), the following rules apply:

1. If in the * $$ JOB statement no disposition has been specified, the disposition gets assigned the value K as default in order to reflect the repetitive processing of the job.
2. When the job has been processed and the original disposition was K, the new disposition becomes K and not L. This is to achieve that the job starts processing without any operator intervention at the next appropriate due date specified in the * $$ JOB statement.
3. All other rules of disposition processing remain unchanged, which means for example when the job has been processed and the original disposition was D, the job gets deleted in spite of the repetitive specification for the job.

When the time event scheduling operands specify the job is supposed to run **just once** at a specific time, the default value for the disposition is D, if no disposition has been specified in the * $$ JOB statement. After processing of the job, normal disposition handling occurs:
1. If the original disposition was D, the job gets deleted.
2. If the original disposition was K, the new disposition is L.

Display of the Next Due Date

The next due date, which is time and day the job gets processed the next time, is displayed in the first line of the PDISPLAY RDR status report - after the number of cards and before the FROM operands. It is displayed only for jobs with disposition D or K.

If the due date has not yet expired, it is displayed together with the constant RUN=. For the format of the displayed date, see "Functions Allowing Date Format Changes" on page 127.

The following example shows jobs that were read in on Sunday, January 5th, 2003. The PDISPLAY command was issued on the same day.

```
D RDR
=>
       1C39I COMMAND PASSED TO POWER

       1R46I  READER QUEUE   P D C S  CARDS B
       1R46I  HUS01    00001 7 D A        4   RUN=12:30,01/09
       1R46I  HUS02    00002 7 K A        3   RUN=16:30,01/09 FROM=(HUS)
       1R46I  HUS03    00003 7 K A        3   RUN=17:30,01/10 FROM=BOE(HUS)
```

Figure 13. Example 1: Status Layout for PDISPLAY of Unexpired Jobs

If the due date has expired and the job has not yet been processed, the due date is displayed together with the constant EXP=.

The following example shows the same jobs as the previous example, but the PDISPLAY command has been issued on January 11th and so far it has not been possible to process the jobs.

```
D RDR
=>
       1C39I COMMAND PASSED TO POWER

       1R46I  READER QUEUE   P D C S  CARDS B
       1R46I  HUS01    00001 7 D A        4   EXP=12:30,01/09
       1R46I  HUS02    00002 7 K A        3   EXP=16:30,01/09 FROM=(HUS)
       1R46I  HUS03    00003 7 K A        3   EXP=17:30,01/10 FROM=BOE(HUS)
```

Figure 14. Example 2: Status Layout for PDISPLAY of Expired Jobs

For jobs with disposition H or L a due date is not meaningful and therefore just two dashes '--' are displayed.

The next example shows the above jobs after their disposition has been changed.

```
D RDR
=>
       1C39I COMMAND PASSED TO POWER

       1R46I  READER QUEUE   P D C S  CARDS B
       1R46I  HUS01    00001 7 H A        4   RUN=--
       1R46I  HUS02    00002 7 L A        3   RUN=-- FROM=(HUS)
       1R46I  HUS03    00003 7 L A        3   RUN=-- FROM=BOE(HUS)
```

Figure 15. Example 3: Status Layout for PDISPLAY of Non-dispatchable Jobs

For jobs queued for transmission in the XMT queue, neither a next due date nor a constant is displayed, because the due date operands are meaningful only for the execution of a job and not for its transmission.

Display of the Time Event Scheduling Operands

In case you cannot recall the time event scheduling operands once specified you may use the PDISPLAY command with the option FULL=YES. This displays all the due date operands which have been specified in the * $$ JOB statement.

If the DUEMONTH operand has not been specified, the default value DUEMONTH=(1-12) is displayed. If the RERUN operand has not been specified, the default value RERUN=YES is displayed.

All the due date operands are displayed separately in a third line. If all specified values do not fit in this third line, another fourth line is displayed containing the values of DUEDAY. The normal PDISPLAY (that is without FULL=YES) does not display all this information in order to keep the overview of the queued jobs as short as possible. The following example shows some jobs read in on Wednesday, January 5th, 2005, before 10:30 a.m. The PDISPLAY FULL=YES command has been issued immediately afterwards.

```
D RDR,FULL=YES
=>
       1C39I COMMAND PASSED TO POWER

       1R46I  READER QUEUE   P D C S  CARDS B
       1R46I  HUS01    00001 7 D A        4   RUN=12:30,01/09
              D=01/05/2005 DBGP=000001
              DUETIME=12:30 DUEDATE=01/09/2005 RERUN=YES
              QNUM=00023
       1R46I  HUS02    00002 7 K A        3   RUN=16:30,01/09 FROM=(HUS)
              D=01/05/2005 DBGP=000001
              DUETIME=16:30 DUEDATE=01/09/2005 RERUN=YES
              QNUM=00024
       1R46I  HUS03    00003 7 K A        3   RUN=17:30,01/09 FROM=BOE(HUS)
              D=01/05/2005 DBGP=000001
              DUETIME=17:30 DUEDAY=(THU) RERUN=NO
              QNUM=00027
       1R46I  HUS04    00004 7 K A        3   RUN=16:30,01/08
              D=01/05/2005 DBGP=000001
              DUETIME=16:30 DUEDAY=(FRI,SUN) RERUN=YES
              QNUM=00030
       1R46I  HUS05    00005 7 K A        3   RUN=17:30,01/15
              D=01/05/2005 DBGP=000001
              DUETIME=17:30 DUEMONTH=(1-12) DUEDAY=(1,15) RERUN=YES
              QNUM=00037
       1R46I  HUS06    00006 7 K A       10   RUN=10:30,01/05
              D=01/05/2005 DBGP=000001
              DUETIME=10:30 DUEDAY=DAILY RERUN=NO
              QNUM=00038
       1R46I  HUS07    00007 1 L A        3   RUN=--
              D=01/05/2005 DBGP=000001
              DUETIME=10:30 DUEMONTH=(1,4,7,10) RERUN=NO
              DUEDAY=(1-12,20-25,31)
              QNUM=00039
```

Figure 16. Example 4: Layout for PDISPLAY FULL=YES

Notes to Example 4:

1. The jobs HUS01 and HUS02 are supposed to be scheduled just once. The next DUEDATE is the same as the specified scheduling date for each job.
2. The next due date of the job HUS03 is the date of the next Thursday
3. The next due date of the job HUS04 is the date of the next Friday.
4. The next due date of the job HUS05 is the date of the next 15th (job is scheduled on 1st and 15th of every month).
5. The job HUS06 is to be scheduled every day.
6. The job HUS07 has a non-dispatchable disposition (L) and therefore two dashes '--' are displayed instead of a next due date. The list of time event scheduling operands is too long. Therefore the DUEDAY information is displayed on a fourth line.

The following example shows the layout of a job with due date operands, queued in the transmission queue. No next due date is displayed, because it does not affect the transmission. The due date operands, however, are displayed the same way as they are displayed for jobs in the reader queue.

```
D XMT,FULL=YES
=>
       1C39I COMMAND PASSED TO POWER

       1R46I    XMIT QUEUE   P D C I  LINES B
       1R46I  HUS01    00001 7 D A R      4   TO=POK
              D=01/05/2005 DBGP=000001
              DUETIME=12:30 DUEDATE=01/09/2005 RERUN=YES
              QNUM=00023
```

Figure 17. Example 5: Layout for PDISPLAY of XMT Queue

Job Sequence in Queue Display

Due to the due date operands the dispatchable jobs of any class in the local reader queue are displayed in a different sequence. The first jobs to be displayed are jobs which are 'really dispatchable' in a specific class; that is, those jobs with disposition D or K, and either without any due date operands or with due date operands that have an expired due date. These jobs are displayed in the normal sequence; that is, according to their priority.

Dispatchable jobs of a specific class with a due date not yet expired are not scheduled before the due date expires. Therefore, they are queued at the end of all 'really dispatchable' jobs, prior to the non-dispatchable jobs. The jobs with a due date not yet expired are displayed in the sequence they have been queued, without regard to their due date or their priority. If a job with due date operands has a non-dispatchable disposition (any disposition other than D and K), the job is displayed according to class and priority.

Therefore, the jobs of the local reader queue build three subchains which are displayed in the following sequence and are repeated for each class:

'dispatchable' chain
: contains all dispatchable jobs of one class without any due date, or with due date, if the due date has expired, sorted by priority.

'wait for run' chain
: contains all dispatchable jobs (with disposition D or K) of one class whose due date has not yet expired, sorted by queuing date (the dates the jobs were queued the last time) and not by priority.

'non-dispatchable' chain
: contains all non-dispatchable jobs (any disposition other than D and K) with or without due date operands of one class and corresponds to the non-dispatchable queue of previous releases.

Jobs with and without due date operands are shown in the following example, which is not related to the previous examples:

```
D RDR
=>
       1C39I COMMAND PASSED TO POWER

       1R46I  READER QUEUE   P D C S  CARDS B
 ( 1)  1R46I  WOT01    00011 9 D A        4
 ( 2)  1R46I  HUS04    00004 7 K A        3   EXP=16:30,01/15
 ( 3)  1R46I  HUS05    00005 7 K A        3   EXP=17:30,01/15
 ( 4)  1R46I  WOT02    00012 2 K A        4
 ( 5)  1R46I  WOT03    00013 2 D A        4
 ( 6)  1R46I  HUS03    00003 1 K A        3   RUN=17:30,01/22
 ( 7)  1R46I  HUS06    00006 7 K A        3   RUN=18:30,02/01
 ( 8)  1R46I  HUS07    00007 3 K A        3   RUN=10:30,02/01
 ( 9)  1R46I  WOT04    00014 9 L A        4
 (10)  1R46I  HUS08    00008 8 H A        3   RUN=--
 (11)  1R46I  HUS02    00002 7 L A        4   RUN=--
 (12)  1R46I  WOT05    00015 6 H A        4
 (13)  1R46I  HUS09    00009 1 L A        3   RUN=--
 (14)  1R46I  BUS01    00016 9 K B        4   EXP=10:05,01/15
 (15)  1R46I  BUS02    00017 3 D B        4
 (16)  1R46I  BUS03    00018 2 K B        3   EXP=06:15,01/15
 (17)  1R46I  BUS04    00019 3 H B        4
 (18)  1R46I  BUS05    00020 1 L B        4   RUN=--
```

Figure 18. Example 6: Status Layout of Jobs with and without Time Event Scheduling Operands

The following notes apply to Example 6, which shows a display taken on January 15 after 5:30 p.m.:

Notes:

1. (1)-(5): The dispatchable jobs of class A are displayed according to the already existing rules (priority and time of queuing in the really dispatchable chain) without any due date operands or with expired due date.
2. (6)-(8): The jobs with a due date that has not yet expired are displayed according to the time of queuing (the priority is **not** considered).
3. (9)-(13): The non-dispatchable jobs of class A are displayed according to the already existing rules (priority, time of queuing).
4. (14)-(18): The jobs of class B are displayed, and the same rules apply as for the jobs with class A.

Display of the 'Wait for Run' Subqueue

Dispatchable jobs of all classes of the reader queue, with a due date that has not yet expired, are chained together in the 'wait for run' subqueue. This 'wait for run' subqueue contains the jobs of all 'wait for run' chains (one chain for each class) but in a different sequence. The sequence is determined by the next due date at which the job has to be scheduled; class and priority are not considered. A new operand for the PDISPLAY command can be used to display this 'wait for run' subqueue. The usual search arguments may be used in connection with the display of the 'wait for run' subqueue.

The following status layout of the PDISPLAY command has been created by the operand WRUN on January 5 before 12:30 p.m.; there is no relation to the previous examples.

```
D WRUN
=>
       1C39I COMMAND PASSED TO POWER

       1R46I  READER QUEUE   P D C S  CARDS B (WAIT FOR RUN SUBQUEUE)
       1R46I  HUS01    00001 7 D A        4   RUN=12:30,01/05
       1R46I  HUS04    00004 4 K A        3   RUN=16:30,01/05
       1R46I  HUS02    00002 7 K A        4   RUN=12:30,01/06
       1R46I  CUS01    00031 9 K C        9   RUN=11:30,01/07
       1R46I  HUS03    00003 7 K A        3   RUN=17:30,01/08
       1R46I  HUS05    00005 7 K A        3   RUN=17:30,01/15
       1R46I  BUS03    00018 2 K B        3   RUN=10:30,01/21
       1R46I  HUS07    00007 1 K A        3   RUN=10:30,02/01
       1R46I  HUS06    00006 7 K A        3   RUN=18:30,02/01
```

Figure 19. Example 7: Status Layout for PDISPLAY Using the Operand WRUN

The PDISPLAY FREE and FREER Commands

The PDISPLAY RDR,FREE command displays all jobs being processed or awaiting processing when a corresponding partition becomes available; these jobs have disposition '*' or D or K. This command, however, displays also 'WAIT for RUN' jobs if jobs with time event scheduling parameters are contained in the dispatchable part of the reader queue. These jobs have to wait first for the expiration of their current time interval before they may be processed by a partition. In such cases, it is recommended to use the PDISPLAY RDR, FREER (Free and Ready-to-run) command to inquire about all jobs being processed or awaiting processing which are not in the 'Wait for Run' subqueue.

Scheduling a Job More Than Once Per Day

Introduction

Using time event scheduling parameters a job may run more than once, for example every first day of a month, every Monday or even daily but be processed only once per scheduled day. Using the operand DUEFRQ (Due Time Frequency) in the VSE/POWER * $$ JOB statement, a job may be processed more than once per scheduled day.

DUEFRQ Operand

The VSE/POWER * $$ JOB statement may contain an operand which specifies a time after which a job is to be rescheduled and an ending time after which no more scheduling should occur. The format is

```
DUEFRQ=(iimm,llnn)
```

The DUEFRQ operand specifies a 'due frequency', meaning an interval using ii for hours and mm for minutes (from 0001 through 2359), after which the job has to be rescheduled, and specifies a last time using ll for the hour and nn for the minute (from 0000 through 2400) after which the job is no longer to be scheduled for processing. Leading zeros must be specified. This operand requires a DUETIME to be specified that defines the time when scheduling has to occur for the first time. This operand is accepted *only* if DAILY or a weekday list has been specified for the operand DUEDAY. DUEFRQ is rejected if DUEDATE or a day list has been specified for DUEDAY. RERUN=YES must not be used for jobs with a DUEFRQ specification. Instead RERUN=NO becomes the default. Hence, if a job with a DUEFRQ specification misses a scheduling event due to system down time, the job is scheduled at the next scheduling event.

The following samples describe the usage of the operands:

```
DUETIME=0000,DUEDAY=DAILY,DUEFRQ=(0030,2400)
```

A job with the above operands is scheduled every 30 minutes every day. If the job is read in at 3:01 p.m., the job is scheduled for the first run at 3:30 p.m.

```
DUETIME=0700,DUEDAY=(MON-FRI),DUEFRQ=(0100,1700)
```

A job with the above operands is scheduled every hour between 7 a.m. and 5 p.m., every day from Monday through Friday (including any holidays). The first time the job is scheduled is at 7 a.m. and the last time is at 5 p.m. If a job with the above operands is scheduled e.g. for the next run at 9 a.m. and its processing finishes after 10 a.m., the job is rescheduled for 11 a.m. The scheduling at 10 a.m. is omitted. This behavior cannot be changed by the RERUN operand.

Note: A job is scheduled every n minutes regardless of how long the job needs for processing. If, for example, a job is scheduled every 5 minutes and is scheduled for 7:00 for the next run and finishes its processing at 7:04, the job is rescheduled for 7:05 and not for 7:09. If however the job finishes its processing at 7:06, the job is rescheduled for 7:10. As no seconds are considered when calculating the scheduling time, it even may happen that a job finishes its processing at 7:04:59 and is rescheduled for 7:05:00.

Displaying the DUEFRQ Information

Once a job has been read in by VSE/POWER, the specified values for the DUEFRQ operand can be displayed by using the PDISPLAY command together with the FULL=YES operand. The time in the third display line after the constant DUETIME= is the value specified for DUETIME and identifies the time when the job has to be scheduled the first time for a selected day.

To illustrate the displayed scheduling information, assume the following sample jobs:

```
* $$ JOB JNM=HUFQ001,DUETIME=0000,DUEDAY=DAILY,DUEFRQ=(0030,2400)

* $$ JOB JNM=HUFQ002,DUETIME=0700,DUEDAY=(MON-FRI),DUEFRQ=(0100,1700)

* $$ JOB JNM=HUFQ003,DUETIME=0600,DUEDAY=TUE,DUEFRQ=(0105,1300)
```

And assume they were read in on Monday, January 10th, 2005, at 9:15 a.m. and the subsequent display was requested immediately:

```
PDISPLAY RDR,HU*,FULL=YES

1R46I  READER QUEUE   P D C S  CARDS B
1R46I  HUFQ001  01092 3 K A        3   RUN=09:30,01/10
       D=01/10/2005 DBGP=000001
       DUETIME=00:00 DUEDAY=DAILY DUEFRQ=(00:30,24:00) RERUN=NO
       QNUM=00469
1R46I  HUFQ002  01093 3 K A        3   RUN=10:00,01/10
       D=01/10/2005 DBGP=000001
       DUETIME=07:00 DUEDAY=(MON,TUE,WED,THU,SUN) DUEFRQ=(01:00,17:00) RERUN=NO
       QNUM=00490
1R46I  HUFQ003  01094 3 K A        3   RUN=06:00,01/11
       D=01/10/2005 DBGP=000001
       DUETIME=06:00 DUEDAY=(TUE) DUEFRQ=(01:05,13:00) RERUN=NO
       QNUM=00491

PDISPLAY WRUN
```

```
1R46I  READER QUEUE   P D C S  CARDS B (WAIT FOR RUN SUBQUEUE)
1R46I  HUFQ001  01092 3 K A        3   RUN=09:30,01/10
1R46I  HUFQ002  01093 3 K A        3   RUN=10:00,01/10
1R46I  HUFQ003  01094 3 K A        3   RUN=06:00,01/11
```

Altering the DUEFRQ Information

Once a job has been read in by VSE/POWER, the specified values for the DUEFRQ operand can be nullified by using the VSE/POWER PALTER command together with the DUETIME=NULL operand. All previous specifications of event scheduling time are now ignored.

Transferring Jobs With DUEFRQ Information to Other Systems

If a job is read in by VSE/POWER, the DUEFRQ operand is processed and the specified values are saved for this job. If the job is transferred to any other VSE/POWER system, this information is retained. The transfer may for example happen via PNET or by offloading the job to a tape and reloading it elsewhere. Whenever the job enters the reader queue, the next scheduling date is calculated anew. Whenever the job is passed from one transmission queue to another transmission queue (at a store and forward node), no next scheduling date is calculated.

DUEFRQ Information on Backlevel Systems Without DUEFRQ Support

Once a job has been read in by a VSE/POWER system, the job may get transferred to another VSE/POWER system, which might be at a different level. If a job with DUEFRQ information is sent to a system without DUEFRQ support, the following is true:

1. No information about DUEFRQ is displayed when using the FULL=YES operand.
2. The next scheduling time is the value determined as the next processing time at the originating system. The next scheduling day is calculated according to the usual rules when entering a system. (The next scheduling time is the time displayed after the constant RUN= when using the PDISPLAY command.)
3. The time displayed for DUETIME is the time for the next scheduling event. This time might now differ on the receiving system from that time which was originally specified for the DUETIME operand on the transmitting system.
4. If at the sending system a job enters the transmit queue through the read-in process and is transmitted, then at the receiving system the next scheduling time is the time specified for the DUETIME operand.
5. If a job is transferred from a system with DUEFRQ support to a system without DUEFRQ support and is transferred back to the originating or another system with DUEFRQ support, all information of the DUEFRQ operand is available to the receiving system with DUEFRQ support.

Scheduling of Jobs at VSE/POWER Startup Time

Jobs of the 'Wait for Run' Subqueue

If VSE/POWER starts processing after the system had been down and if - in the meantime - the due date has expired for a job of the 'wait for run' subqueue, this job is automatically set eligible for processing. If, for example, a job with a due date of June 2nd has entered the system on June 1st, but the system was down on June 2nd and restarted on June 4th, the job is set eligible to be processed on June 4th (Figure 20 on page 174).

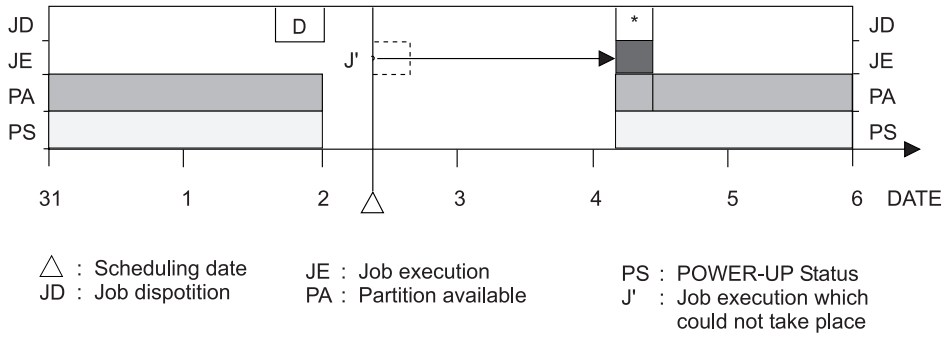


Figure 20. Scheduling of a 'One-Time-Running' Job (System Down)

If the job was meant to be processed more than once, it is processed just once. For example, a job has to be processed daily and the system was not running for 3 days, the job is processed only once instead of three times.

In these cases it may occur that a daily job runs twice a day: for example, a job runs daily at 7 o'clock, the system is stopped on Monday at 6 o'clock and restarted on Tuesday at 4 o'clock. Here, the job runs immediately on Tuesday at 4 o'clock due to the missing event 'run on Monday' and once more on Tuesday at 7 o'clock due to the scheduling event 'run on Tuesday' (Figure 21).

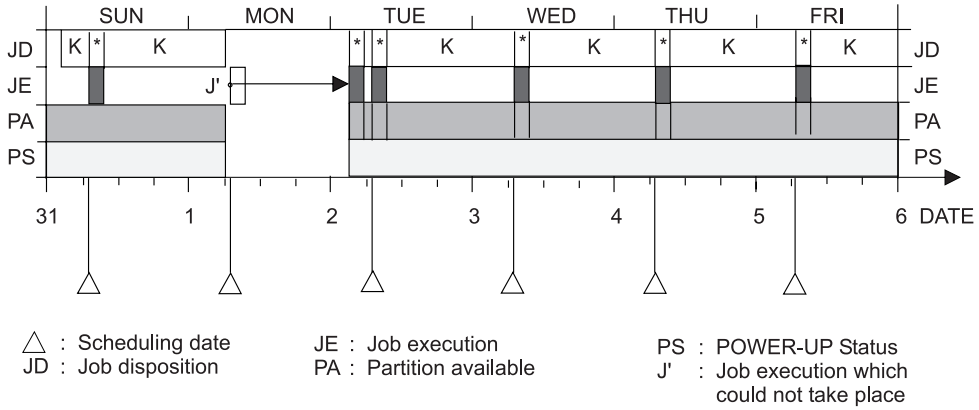


Figure 21. Scheduling of an 'N-Time-Running' Job (System Down)

In order to avoid that a job runs twice a day (as described above) an operand RERUN=NO may be specified in the * $$ JOB statement. If this operand is used, a job does not become eligible for processing when the due date has expired during the time VSE/POWER was down.

Expired Jobs

Once the due date of a job has expired, the job is available for immediate processing. If VSE/POWER was terminated and is restarted after quite a while, then VSE/POWER does not reschedule already expired jobs. Instead this job is going to be processed as soon as a partition is available.

It may be possible that this job runs twice a day, if the next due date is later on that same day. No operand can be specified to avoid this. The proceeding is the same for all VSE/POWER startups (warm start, partial or full recovery). (Figure 22 on page 175).

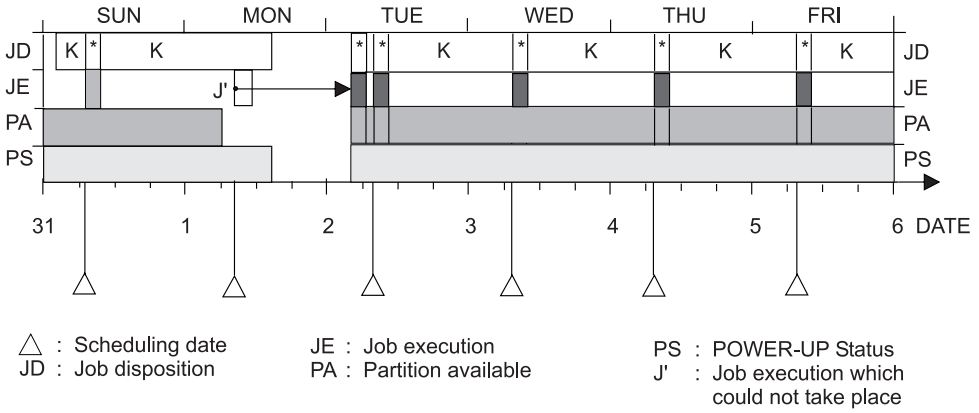


Figure 22. Scheduling of an 'N-Time-Running' Job (No Partition Available)

Expiration at PEND Time

Once the PEND command is issued, the already processing jobs are allowed to continue until normal end of job. No new job is started. This means, although the due date of a job expires after PEND has been issued, this job is not started for processing. It is just added to the dispatchable queue. The job is then processed as soon as VSE/POWER is restarted.

Missing Events of Expired Jobs

If the job could not be processed because no partition was available for a few days, and if, meanwhile, the next due date expired, another scheduling date will be missed. This happens, because the next due date for a job is not calculated before the end of its processing.

Recovery of Jobs in Execution State

If VSE/POWER terminates abnormally and a job with an expired due date was processed at that time, this job is started for processing again as soon as VSE/POWER has completed its spool recovery.

If the autostart statement SET NORUN=YES has been specified, the job gets assigned the disposition X. If this disposition X is altered to K (or D, using the PALTER command with the operand DISP=K, DISP=D or DISP=*), the job is queued for 'immediate' processing.

If the disposition X is altered to L (or H) and later on to K (or D), the job is not queued for 'immediate' processing. Instead a new next due date is calculated with regard to the current date.

This means for example:

1. A job is to be scheduled at January 1st, 5th and 10th.
2. The job starts to be processed at January 1st, but VSE/POWER terminates abnormally.
3. VSE/POWER enters recovery on January 1st and assigns disposition X to the job.
4. The disposition gets altered to L (no matter at which date, just before January 5th).
5. The disposition gets altered to K (no matter at which date, just before January 5th). The job gets eligible for processing on January 5th.

Scheduling within Shared Systems

There is no difference between time event scheduling of shared systems and non-shared systems. The due date of all jobs is known in every single shared system in order to guarantee the scheduling of a job whose due date has already expired.

Clearing the Time Event Scheduling Information

With the PALTER command you can nullify the time event scheduling information. Once this has been done, all the time event scheduling information is lost and can no longer be retrieved. The job then is processed according to the already existing scheduling rules, that is, a job with disposition D or K gets queued for 'immediate processing'.

Search Argument

The operand CDUE=* is supported as a search argument to address all jobs with due date operands. This operand may be used in connection with the following commands:

PALTER for altering characteristics of a job.
PDELETE for deleting a job.
PDISPLAY for displaying a job.
PHOLD for making a job non-dispatchable.
POFFLOAD SELECT for copying a job from tape to disk.
PRELEASE for making a job dispatchable.

The CDUE operand is important only for reader jobs which are queued either in the RDR or XMT queue. If the CDUE operand and LST or PUN for the queue operand is specified, the following message is issued:

1R34I *commandcode* **OPERAND** *nn* **NOT MEANINGFUL FOR LST OR PUN QUEUE**

Saving Jobs onto Tape

If the POFFLOAD command is used with the SAVE operand, dispatchable jobs of the local reader queue with due date operands are not copied to tape. If the POFFLOAD command is used with the BACKUP or PICKUP operand, all jobs with due date operands also are written to tape.

If a job gets reloaded from tape to disk, a new due date is calculated according to the scheduling rules (see "Scheduling Rules" on page 164).

Accessing Jobs via the Spool-Access Support Interface

A job with due date operands can be submitted/processed by means of the PUT and CTL services in the same way as a job without due date operands. The fixed format display - as a result of a CTL service request - is extended to show the next due date and whether the due date has expired or not.

A dispatchable job of the local reader queue with due date operands can be retrieved by means of the GET service only if the BROWSE option is used. If the GET service without the BROWSE option is used, the return code PXPRCOKF (X'04') and the feedback code PXP04NDS (X'04') are supplied.

Jobs with Disposition I

If the disposition I is specified in the * $$ PUN statement, the punch output is queued to the reader queue. The newly built queue entry gets assigned some of the characteristics (for example, jobname, priority, etc.) of the queue entry that produces the output. However, the due date operands are not assigned to this newly built queue entry.

Ignoring the Time Event Scheduling Operands

With special features of VSE/POWER or other products it is possible to write jobs to the spool file containing * $$ JOB statements. These * $$ JOB statements are neither checked nor processed at the time the * $$ JOB statements are written to the spool file. On the contrary, the * $$ JOB statements are scanned for completeness at the time a partition retrieves the job and starts processing.

Most operands, however, do not affect the job anymore, for example, the time event scheduling operands. It would be too late to consider a due date, when processing already has started. Nevertheless, the due date operands are scanned for correct syntax, because they may be followed by other operands which can affect the job and therefore must be searched for.

The following examples explain these situations:

- If the disposition I is specified in the * $$ PUN statement, the punch output is queued to the reader queue. If within this punch output an * $$ JOB statement has been punched, the * $$ JOB statement is just written to the spool file and scanned at processing time.
- If an output job is received via network, this job also may be queued directly to the reader queue (using PWR$JOB as destination remote name within the network data set header record). In this case, too, the job written to the reader queue may contain an * $$ JOB statement that is just written to the spool file and scanned at processing time.
- If a reader job is received via network, this job may contain * $$ JOB statements, which the sending system did not process. This happens if the sending system is a non-VSE system (RSCS or z/OS) or a VSE-system and the job was produced using the DISP=I operand in the * $$ PUN statement. The network receiving part of VSE/POWER processes these * $$ JOB statements not before the job reaches its final destination node (that is the job gets queued to the local reader queue). Sometimes the network receiving part did not process the * $$ JOB statements (for example, if the first record of the received job was not an * $$ JOB statement). In this case, the * $$ JOB statements are not scanned before the job starts processing. For more details, see *VSE/POWER Networking*.

  If the job is received at an intermediate (store and forward) node, the job is spooled to the transmission queue and the * $$ JOB statements are written to the spool file without any processing. If on the intermediate node the destination gets now changed in a way that the job is queued into the local reader queue, the * $$ JOB statements are not scanned before the job starts processing.

Additionally, there are 'writer-only' partitions. That is, VSE/POWER controls only the output of these partitions. The input (the job to be read in) and to be processed, is not part of the VSE/POWER spool file.

If such a job contains a VSE/POWER * $$ JOB statement with time event scheduling information, then this information is ignored and the job starts

processing as soon as it enters the system. As seen above, the due date operands are just scanned for correctness, but time event scheduling itself does not take place.

Invocation of Time Event Scheduling Support

As described, the * $$ JOB statement offers due date operands. With them you may specify the day(s) and time the job has to be scheduled for processing. For a detailed description of these operands see "* $$ JOB: Marking the Start of a VSE/POWER Job" on page 413.

Notes:

1. For simplicity reasons the due date operands are specified in the description of the * $$ JOB statement in some sequence. But this sequence is not mandatory. Other operands might be specified in between the due date operands.
2. If any of the operands have been specified more than once, no message is issued and the last specification overwrites the previous ones.
3. If no operand after the keyword is specified (for example, DUETIME=,), the operand is ignored. No error message is issued, and the last specification (if there has been one) is not changed.
4. If 29, 30 or 31 is specified for DUEDAY and the month does not have so many days, the last day of the month is assumed. This means for example, if DUEDAY=31 and DUEMONTH=(1-12) (which is also the default) is specified, the job is scheduled on January 31, February 28 (or 29, if leap year), March 31, April 30, etc. Or if DUEDAY=30 and DUEMONTH=(1-12) is specified, the job is scheduled on January 30, February 28 (or 29, if leap year), March 30, April 30, etc.
5. With the existing logic you can select a month list and a day list which specify the *same* selected days for *all* specified months. If you need different day lists for specific months, you may submit multiple copies of the same job to the reader queue, so that each job stands for a unique day list.
6. If the DUEDATE operand has been specified, the two-digit year is expanded to a four-digit year (applying the 'fix-88-window rule' defined on page 417), and the four-digit year is checked for a meaningful range:

   current year - 10 <= year specified <= current year + 3.

   If the specified year is not within this range, the following message is issued:

   1R37I *jobname jobnumber* **WITH IMPROBABLE YEAR SPECIFICATION**

   The job is placed in hold status. Note that all due date operands are valid and all other operands remain unchanged except the disposition which is changed to H.

   If the disposition gets changed to a dispatchable one, the due date operands are considered to be valid and the job is scheduled according to the due date operands.
7. If one of the time event scheduling operands is invalid, all information already checked concerning time event scheduling is ignored. No defaults are used.
8. If any operand is invalid, even if it is not a time event scheduling operand, all information already checked concerning time event scheduling is ignored. No defaults are used.

9. When VSE/POWER processes all operands (not only the due date operands) of the * $$ JOB statement, a consistency check is performed:
   - At least the DUETIME operand must have been specified.
   - Some of the due date operands are mutually exclusive, for example, DUEDAY and DUEDATE.

   In case of an error, the following message with the appropriate return code is issued:

   1R36I *jobname jobnumber* **WITH INCOMPLETE OR CONFLICTING TIME EVENT SCHEDULING PARAMETERS, RC=***nnnn*

   The job is placed in hold status. All due date operands are ignored and all other operands remain unchanged except the disposition which is changed to H.

Recording of Spooled Data on the Data File

Whenever VSE/POWER spools job or output records on the data file, blank truncation takes place automatically in order to save disk space. That means trailing blanks are removed from the data records and are never added to the records again. For execution of jobs and local processing of output, blank truncation remains transparent.

However, when VSE/POWER acts as a store-and-forward node amongst non-VSE nodes, or when output processing programs access VSE/POWER queue entries via the Spool-Access-Support interface, even trailing blanks may be of importance. Therefore, blank truncation may optionally be suppressed at
- Local or RJE read-in-time, using the BTRNC=NO operand of the * $$ JOB statement for job data
- Job execution, that is output production time, using the BTRNC=NO operand of the * $$ LST or PUN statement for output data
- Spool-Access-Support PUT time, specifying the SPLGO2BT option flag in the PUT-OPEN-SPL for job/output data
- PNET receive time, using the NETEXIT return code X'80yy' per received job or output entry.

Options When Data File Is Full

Expected Shortage

In general the data file should be large enough that it can accommodate even peak loads of spooling. When for a given size of the data file your spool space usage grows gradually due to increased workload, this means that, when the high-water-mark message

```
MAX. NUMBER OF DBLK-GPS USED SINCE LAST COLDSTART
```

of the PDISPLAY STATUS report shows rising data block group consumption, VSE/POWER will notify you sooner or later by message 1QF0I. This indicates that the default spool limit value of 90% (or your generated SPLIM= value of the POWER macro) has been reached. The message is repeated every minute and reminds you to plan for an orderly extension of the data file by either

- a warm start (see "Extending the Data File During Warm Start" on page 38), which does not touch the existing job queues and formats one *additional* data file extent while VSE/POWER spooling continues, or
- a cold start (see "Extending the Queue or Data File During a Cold Start" on page 38), which is only recommended when
  - the extent of the existing data file must be relocated for an extension, or
  - the queue file must be extended at the same time.

The latter total 'cold start' reformatting of an extended/relocated data file (see Figure 1 on page 29) addresses all data file extents including your extension. For a medium-sized or large data file, the formatting step can take from one to three hours. If you have twice the spool space on disk and if you can run a test machine in parallel to your production machine, you can avoid production outage by taking the following steps:

1. Run your test system with a VSE/POWER phase generated for the same DBLK and DBLKGP values as your production system.
2. If your production system supports PNET, ensure that the VSE/POWER phase of the test system either runs without PNET support or with PNET, that uses the same local node name as your production system.
3. If your production system supports shared spooling, the VSE/POWER phase of the test system may either be generated for non-shared or shared support, which does not take influence.
4. Cold start your test system for the new extended data file and queue file to have them formatted. Make sure the test system switches back to its private (existing) queue and data file for the subsequent startup. Then shut down the test system by PEND.
5. On the production system, offload queue entries required later. Update the STDLABEL and DTRPOWR procedures for the DLBL and ASSGN's of the new queue/data file pre-formatted on the test system. Then shut down the production system.
6. Warm-start your production system, which takes over the extended queue/data file for immediate spooling. Reload offloaded queue entries.

Unexpected Shortage

When you are notified by message 1QF0I that spool space is running out and cannot or do not react to effect an orderly extension of the data file in time, and when some dumping or looping program produces more and more output leading finally to message 1Q38A NO MORE DASD SPACE AVAILABLE, which is issued for various tasks, the following remedies may help to recover from the spooling-stalled state:

- Let VSE/POWER search for and identify the largest consumers of spool space (data block groups) in any known VSE/POWER queue (CRE/RDR/LST/PUN/XMT/DEL) using the PDISPLAY BIGGEST,LIMIT=nn comand. Then address selected large entries within their named queue as suggested below
- And/or request VSE/POWER to identify queue entries of the physical RDR/LST/PUN/XMT queues that hold more than a given number nnnnnnnnn of pages or cards using, for example, the PDISPLAY LST,CPAGES>nnnnnnnnn or PDISPLAY PUN,CCARDS>nnnnnnnnn command. Then address selected large entries as suggested below.

- Process (i.e., print, punch, or transmit) large queue entries which are usually deleted after processing and provide free DBLKGP's.
- Offload large queue entries. The SAVE function implicitly deletes processed entries, while the BACKUP/PICKUP functions require deletion by the operator.
- Delete large queue entries (use FULL=YES display for DBGP consumption) which are no longer required, e.g. dump output or outdated output.
- Check if entries in creation consume a large amount of spool space. Use the PDISPLAY A or PDISPLAY CRE command to determine, from 'LINES Spooled', which entry is consuming many data block groups and who is creating it.
- If executing partitions are the culprit, you may
  - either PFLUSH the partition or PCANCEL the job, so that its collected output becomes visible in the physical (RDR/LST/PUN/XMT) queues and can be processed or be deleted there, or
  - if job execution must not be interrupted, request that its in-creation output be segmented immediately by the PSEGMENT or PALTER...,SEGMENT= commands. Depending on its importance, the output segment can then be processed or deleted from one of the physical queues.
- Check if quasi-deleted queue entries still occupy spool space because they reside in the deletion queue to allow browsing by someone (see "Delaying the Deletion of Browsed Queue Entries" on page 122).
  - use the PDISPLAY STATUS command and check for the total of Queue-Records/DBLK-Groups IN DELETION
  - use the PDISPLAY DEL command and find out which entry is 'in delayed deletion', how much DBGP spool space is claimed, and how many Spool Access Support (SAS) user task(s) are still browsing the entry.

In order to stop the browsers and thereby trigger final deletion of the queue entry to re-gain spool space, do the following:

1. Use the PDISPLAY A,SAS command and find out which SAS user task(s) are browsing the queue entry in question.
2. Remember their SAS,nnnnn identification key for specification in the PSTOP SAS,nnnnn command to terminate browsing abruptly.

Separator Pages - Layout and Control

When n=1...9 separator pages have been ordered for printed output, VSE/POWER automatically adds an (n+1)-th page, called **'last-one'**. It contains less information than the first n pages. The contents of the first n pages is depicted in the following example.

Notes:

1. The (n+1)th page is dropped and not printed if the LST task has been started with operand 'DLSEP' or runs under control of the 'SET DLSEP' autostart statement or if the Spool Access (SAS) or Device Service (DST) task runs under the control of 'SET DLSEPSAS'.
2. The size and layout of the separator page is determined at output creation time, spooling time, by the FCB or LTAB specified in the * $$ LST statement. If neither is specified, the default LTAB determines the page size (refer to the POWER generation macro).

Following is an example of a heading separator page.

```
1                          MM        MM  YY        YY       JJJJJJ   0000000000   BBBBBBBBBBB
                           MMM      MMM  YY        YY       JJJJJJ  000000000000  BBBBBBBBBBBB
                           MMM      MMM  YY        YY           JJ  00        00  BB        BB
                           MMMM    MMMM   YY      YY            JJ  00        00  BB        BB
                           MMMM    MMMM    YY    YY             JJ  00        00  BB        BB
                           MM MM  MM MM     YY  YY              JJ  00        00  BBBBBBBBBBB
                           MM MM  MM MM      YYYY               JJ  00        00  BBBBBBBBBBB
                           MM  MMMM  MM       YY                JJ  00        00  BB        BB
                           MM  MMMM  MM       YY        JJ      JJ  00        00  BB        BB
                           MM   MM   MM       YY        JJ      JJ  00        00  BB        BB
                           MM   MM   MM       YY        JJJJJJJJJJ  000000000000  BBBBBBBBBBBB
                           MM        MM       YY         JJJJJJJJ    0000000000   BBBBBBBBBBB
2


3      0000000000       11                                                                      3333333333   555555555555
      000000000000      111                                                                    333333333333  555555555555
      00        00     1111                                                                    33        33  55
      00     00 00     1111                                                                              33  55
      00    00  00       11                                                                              33  55
      00   00   00       11                                                                      333333333   55555555555
      00  00    00       11                                                                      333333333   555555555555
      00 00     00       11                                                                              33            55
      00 0      00       11                                                                              33            55
      00        00       11                                                                    33        33  55        55
      000000000000     111111                                                                  333333333333  555555555555
      0000000000      111111                                                                   3333333333    5555555555
4









5 NODE: JEC4      USER:           ORG NODE: JEC4      ORG USER: MYUSER    EXEC.NODE: JEC4      ORG JOB-NO: 00035
6 DEV : 00E       FNO : MYFORM    FCB      :          LINES   : 00000005  CLASS    : T
7 NAME: MEOWN                     BLDG.    : MYHOUSE  ROOM    : MYROOM    DEPT.    : D3258     DIST      : BOX04
8

9 **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
  **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
  **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
  **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
  **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
  **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
  **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
  **** SYSID 4  BG ***   START   MYJOB      00035   ONLY    MY-PRIVATE-INFO    16 FEB 2005 13.34.42   ** VSE/POWER V7.1 **
```

1 VSE/POWER jobname presented in block letters of 12 lines.

2 Three blank lines

3 A two digit number which shows the sequence of printed output and the jobnumber both in block letters of 12 lines.

The sequence number starts with 01 and increases by one for every job (for local tasks only). If 99 is exceeded, the sequence number starts again with 01. The number is increased also for jobs not requesting separator pages.

4 A series of blank lines (10 in this example), depending on the page size defined by the Forms Control Buffer (FCB) or LTAB.

5 First detailed information line, showing
- Node and user ID of the target destination
- Node and user ID of the originator
- Name of the execution node
- Jobnumber on the originating node.

6 Second detailed information line, showing

- 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.
- Form number
- Name of the FCB image phase
  - if the separator page is printed by a physical printer, the name describes the FCB used at actual print time
  - if the separator page is passed by the spool access support (SAS), which includes device driving systems (DDS), the name describes the FCB specified in the * $$ LST statement
  - if the FCB specified in the * $$ LST statement was not found at spooling time or no FCB has been specified in the * $$ LST statement, no name is displayed on the separator page.
- Total number of lines for the output or output segment
- Output class.

7 Third (optional) detailed information line, showing

- Name of programmer
- Building number
- Room number
- Department number
- Output distribution code.

8 Two blank lines

9 Eight lines, planned to print over the perforation, with the following equal contents:

- 12 asterisks or, if the shared spooling function is used, 4 asterisks and the SYSID of the executing system
- ID of the partition in which the originating job was executed, provided the information is available
- START or END indication for heading or trailing separator pages
- Jobname, jobnumber, and segment number of the output

  Note: On separator pages which delimit the last or only segment of printed output, the segment number reads LAST or ONLY, respectively
- UINF (also USER) information from the * $$ JOB or * $$ LST statement
- Date and time the output was printed, with print date in the current date format (see “Functions Allowing Date Format Changes” on page 127).
- ID of the spooling system, including version and release level.

Notes:

1. The described default number of 8 lines can be modified via the autostart statement ’SET **LINE=n’ (’n’ is in the range from 1 to 8). This is used only if the LST task is started with the operand ’DLSEP’ (or under control of the ’SET DLSEP’ autostart statement), or if Spool-Access (SAS) or Device Service (DST) tasks run under control of ’SET DLSEPSAS’, and allows ending the separator page before the perforation. For laser printers which cannot print on the

perforation (e.g., 3800) or use non-continuous forms, 'SET **LINE=n' may be used in combination with the DLSEP operand or the 'SET DLSEP' or the 'SET DLSEPSAS' statement.

2. The (n+1)-th 'last-one' separator page contains only the information of lines **1** - **3** .

Drop 'Last-One' and Print Shorter Separator Page

Separator pages are intended to be printed on continuous forms across the perforation to show the printer operator where an output ends and a new output starts. The number of separator pages is specified in the POWER macro (JSEP=n) and may be influenced by the * $$ LST statement and by operands of the 'PSTART LST,...' command. All 'n' separator pages contain the same information and are identical. The last lines of the 'n'th separator page are printed on the next page, which is then filled with an additional separator page called 'last-one'. This 'last-one' separator page is different and does not print across the perforation because it contains fewer lines than the preceding separator pages. All separator pages can be forced to be equal by the SET ISEP or SET ISEPSAS statement of the VSE/POWER autostart or by the ISEP|ISEPJ operand of the 'PSTART LST,...' command. Then the last lines of the 'last-one' separator pages are printed on page '(n+2)'.

For laser printers, such as the 3800, which cannot print on the perforation, or for OEM page printers that emulate IBM impact printers but use single sheets, where no perforation exists, this VSE/POWER function is useless. The lines that are to cross the perforation are printed instead at the top of the next page or sheet.

To avoid this, the operator has the choice of starting the local list tasks with the operand DLSEP (or control task operation by the 'SET DLSEP' statement) or run Spool-Access (SAS) and Device Service (DST) tasks under control of 'SET DLSEPSAS', which drops the 'last-one' separator page and decreases the size of the preceding separator pages to let them fit on one page or sheet. The size of the separator page is then influenced by the VSE/POWER autostart statement 'SET **LINE=n'.

Controlling Separator Pages

The following controls may be specified to request separator pages:

JSEP=n
: operand of the POWER macro specifying the number of separator pages generally for the time when list output is printed locally, sent to an RJE line, or passed to a spool-access support GET request.

JSEP=m
: operand of the * $$ LST statement allowing to overwrite the POWER JSEP=n specification for a specific list queue entry at output spooling time.

SPLONSEP
: option of the PUT-OUTPUT PWRSPL allowing to overwrite the POWER JSEP=n specification for a specific list queue entry when spooled through the spool-access support interface.

SEP|NOSEP
: operand of the PSTART LST command to overrule any other separator specification for all output entries addressed by this list task.

DLSEP
: operand of the PSTART LST command to drop the 'last-one' separator

page (of those specified in the JSEP operand of the POWER generation macro), and to print all separator pages with the number of perforation lines specified by SET **LINE.

ISEP|ISEPJ
operand of the PSTART LST command to enforce that the 'last-one' separator page has the same contents as all previous ones for all output entries addressed by this list task.

SET ISEP=YES|FORCE
autostart statement to enforce identical layout of all separator pages for each list entry handled by a local list task.

SET ISEPSAS=YES|FORCE
autostart statement to enforce identical layout of all separator pages produced for list entries that are requested by Spool-Access (SAS) tasks or Device Service (DST) tasks.

SET DLSEP=YES|FORCE
autostart statement to force, for each list entry handled by a local list task, dropping of the 'last one' separator page and printing of all separator pages with the number of perforation lines specified by SET **LINE.

SET DLSEPSAS=YES|FORCE
autostart statement to force each list entry handled by a Spool-Access (SAS) or Device Service (DST) task, that the 'last one' separator page is dropped and that all ordered separator pages are created with a number of asterisk lines (over the perforation) as specified by the SET **LINE autostart statement.

The following table summarizes the influence of the described control operands on 'number-of-separator-pages' and their layout:

Table 7. Influencing Separator Pages and Layout

Invoke and influence printing function by	**Number of separator pages/cards determined from**	**'Last-one' identical to all previous separator pages**
PSTART LST/PUN	JSEP=n value of POWER macro, *optional overwrite* by - JSEP value of * $$ LST - SPLONSEP value of PUT-OPEN	NO
SET DLSEP=YES and PSTART LST	Same as above (no 'last-one')	Not applicable
SET DLSEPSAS=YES and Spool Access(SAS) or DST task	Same as above (no 'last-one')	Not applicable
SET ISEP=YES and PSTART LST	Same as above	YES
SET ISEPSAS=YES and Spool Access(SAS) or DST task	Same as above	YES
PSTART LST,...,ISEPJ	Same as above	YES
PSTART LST/PUN,...,SEP	JSEP=n value of POWER macro, (*without overwrite facility*) if n=0, force n=1	NO
SET DLSEP=FORCE and PSTART LST	Same as above	Not applicable

Table 7. Influencing Separator Pages and Layout (continued)

Invoke and influence printing function by	Number of separator pages/cards determined from	'Last-one' identical to all previous separator pages
SET DLSEPSAS=FORCE and Spool Access(SAS) or DST task	Same as above	Not applicable
SET ISEP=FORCE and PSTART LST	Same as above	YES
SET ISEPSAS=FORCE and Spool Access(SAS) or DST task	Same as above	YES
PSTART LST,...,ISEP	Same as above	YES
PSTART LST,...,DLSEP	Same as above (no 'last-one')	Not applicable
PSTART LST|PUN,...,NOSEP	0	Not applicable

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 workstations that the shutdown procedure is about to begin (for SNA terminal users, a message is issued automatically). Then enter the command

```
PEND
```

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 every partition under its control as soon as program execution in the partition is finished.
4. If dynamic partition scheduling has been activated, VSE/POWER stops allocation of further dynamic partitions.
5. Before VSE/POWER terminates, a statistics status report is displayed on the system console. The same report (see Figure 7 on page 46) may be obtained by the PDISPLAY STATUS command.
6. In case `PEND 00E` had been entered, VSE/POWER prints an additional status report on the printer at address 00E. This status report consists of a statistics part, as shown in Figure 7 on page 46, and a printout of the contents of all VSE/POWER queues. This printout is identical to a PDISPLAY ALL,LST status display as shown on page 257. If you do not specify the address of a printer, VSE/POWER does not print the status report.
7. VSE/POWER termination is complete when you get message

   ```
   1Q21I VSE/POWER HAS BEEN TERMINATED
   ```

Note: To avoid inadvertent VSE/POWER termination, you may request confirmation of the PEND command by providing the SET

CONFIRM=PEND autostart statement. For further explanations, see "SET: Setting VSE/POWER Startup Control Values" on page 473.

Expedited Termination by the PEND IMM Command

With PEND IMM, VSE/POWER terminates as fast and smoothly as possible. Jobs running in a partition are not allowed to run until normal end of job (in opposition to the PEND command), but are canceled. In contrast to the PEND FORCE command, the output - so far produced by these jobs - is not lost but terminated and added to the spool files.

Abnormal Ending of VSE/POWER

If VSE/POWER ends abnormally, the operator is informed by message 1Q2CI, and

1. for failure of a VSE service subtask, a formatted storage dump of the VSE/POWER partition is taken automatically and the subtask is detached. In reality, the VSE/POWER partition continues processing and re-attaches the former service subtask when needed.
2. for failure of the VSE/POWER main task (synonym for VSE/POWER partition):
   - either the SET 1Q30D=YES autostart statement has requested operator communication by message 1Q30D, giving an opportunity to take a standalone dump of the VSE/POWER and spooled partitions, or
   - no communication halt has been specified, and a formatted storage dump of the VSE/POWER partition is taken automatically.

When VSE/POWER operates in a private partition, a partition dump contains the storage copy of the queue file residing in the partition Getvis area. When operating in a shared partition, the queue file resides in the VIO area and, therefore, the automatic formatted dump does not contain a copy of it. However, if the SET 1Q30D=YES autostart statement has been provided at startup, VSE/POWER prompts the operator by message 1Q30D to indicate whether a storage dump is required. If the operator replies with YES, VSE/POWER prompts by message 1Q2ED to indicate whether a printout (SYSLST format) of the storage copy of the queue file is needed. If the operator replies with a valid printer or tape address, VSE/POWER prints, on the specified device, the VIO storage containing the storage copy of the queue file. This VIO dump is seldom needed and may be suppressed.

If a dump sublibrary has been defined for the VSE/POWER partition, VSE/POWER writes the formatted dump of the partition into this sublibrary. If no such sublibrary exists or if the sublibrary is full, VSE/POWER prompts by message 1QC5D for the address of the printer or tape on which the formatted dump is to be written in SYSLST format. In case of tape, use the VSE/DITTO tape-to-printer function to print the tape contents. 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.

For more information on this dump, refer to "Dump of the VSE/POWER Partition at Abnormal End" on page 512.

If during the described abnormal termination processing the VSE/POWER main task fails a second time, it will enter "termination processing" recursively. The

operator is then prompted by message 1Q25D to either take a standalone dump or continue without a dump and let VSE/POWER terminate by a CANCEL macro request.

Abnormal End in User Exit

VSE/POWER can tolerate most coding failures introduced by user written exit routines that used to terminate the spooling system of VSE/POWER 5.2 and earlier releases abnormally. The operator realizes recovery from failure of user exit when VSE/POWER

- Identifies the exit phase in the 1Q2CI termination message
- Rises from break down as accompanied by messages *1Q2KI* and *1Q2HI*
- Sets the faulty exit phase into 'FAILED' state
- Stops the task using the exit or tasks intending to enter a 'FAILED' exit
- Continues normal processing for all other tasks.

For more details, refer to "Writing Various Exit Routines" in *VSE/POWER Application Programming*.

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 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.

Restart and Recovery after Abnormal End

After abnormal end or emergency termination, VSE/POWER warm start enters automatically spool file recovery. Jobs found active (marked by DISP=*) will get their original disposition D or K and will execute again when a partition with a matching class becomes available.

If VSE/POWER is terminated abnormally by a job this job might execute again after restart and will terminate VSE/POWER again. To prevent these jobs from starting again, use the VSE/POWER autostart statement 'SET NORUN=YES'. This will set all active jobs found during recovery to disposition 'X' and will inform the operator.

If you are unable to include the 'SET NORUN=YES' statement in your VSE/POWER autostart you can trigger this function by setting the first bit of the UPSI byte to '1' before the VSE/POWER phase is executed. Issue the AR command 'PAUSE fn' for the VSE/POWER partition before VSE/POWER is started. The

partition will then wait for the operator response after the first '// JOB' or '/&' or '// EXEC phase' statement read. Note that a '// JOB' statement should preceed the execution of the VSE/POWER phase.

Now you can enter '// UPSI 1' on the console and all jobs found active during recovery will get the new disposition 'X' which prevents their execution.

Emergency Termination

When it is necessary to terminate VSE/POWER immediately because of an emergency situation or because a formatted dump is required, submit the command

```
PEND FORCE
```

As a result, message 1Q2DI will be issued relating the cancel request to phase IPW$$CE and a formatted dump will be written to the dump sublibrary. All input and output for the currently processed jobs will be broken off. A warm start enables you to restart interrupted output from the beginning. An input job stream from a card device or from a remote terminal is lost and must be resubmitted. The same is true for a job or for output submitted from a partition, unless the output has been checkpointed.

If VSE/POWER termination is not desired, the Attention Routine command

```
DUMP partition,...
```

may be used. It should be followed by the Attention Routine command

```
DUMP GETVIS
```

to obtain the System Getvis area too, which contains the nucleus module of VSE/POWER with the Common Address Table (CAT) and the VSE/POWER partition control blocks of spooled static and dynamic partitions. However, the attention routine dump facility should be used with care, because system processing continues during dumping.

Termination and Re-IPL in an Unattended Environment

In an unattended environment, VSE/OCCF controls the system. This means that as soon as one of the important system components (NetView, VTAM or VSE/POWER) terminates abnormally, VSE/OCCF tries to restart this component.

If VSE/OCCF itself terminates abnormally, VSE/POWER takes over the control of the system. This means:

1. VSE/POWER simulates internally a PEND IMM command to terminate itself as fast and as smooth as possible.
2. VSE/POWER waits until all partitions are unbatched. This means, all running jobs are canceled, but they get a time span to process an abnormal termination routine (if one has been defined). The time span allowed may be specified in a VSE/OCCF generation macro.
3. VSE/POWER performs a re-IPL of the system.

See also "Automated Operations", in particular "Automatic Re-Ipl" in *VSE/ESA Unattended Node Support*.

Chapter 5. VSE/POWER Operator Commands

Operator commands are used at a console to control the processing of jobs under VSE/POWER control.

VSE/POWER assigns a number to every job it logs so you can distinguish between jobs for which identical names were specified.

A job that has been spooled by VSE/POWER is referred to as a VSE/POWER job or as a queue entry.

At a remote terminal, you use "Terminal Operator Commands" which are discussed in *VSE/POWER Remote Job Entry*.

Note: For information on how to read the syntax diagrams, please refer to Chapter 1, "Understanding Syntax Diagrams," on page 1.

Overview of the Commands

In addition to VSE/POWER's central operation, the operator commands also allow you to control 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 startup is complete.

Table 8 on page 192 gives an overview of the VSE/POWER operator commands. They are grouped by function and, within the function, they are grouped alphabetically. Available short forms also are shown.

Table 8. IBM VSE/POWER Operator Commands

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, or segment output.
PDELETE	L	Delete queue entries or messages.
PDISPLAY	D	Display the status of jobs.
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.
PSEGMENT	M	Segment output spooled by execution writer tasks.
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.
PDISPLAY	D	Display the status of active tasks.
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; start a task trace.
PSTOP	P	Release a partition from VSE/POWER control, stop a task or the task trace, or end a link or a session between two nodes.
Control Commands		
PACCOUNT	J	Save account file records.
PBRDCST	B	Transmit a message.
PDISPLAY	D	Display messages and resources, the status of the active Dynamic Class Table, and the Network Definition Table.
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 one of the following: - a Network Definition Table - a user-written exit routine - or a Dynamic Class Table.
PRESET		Reset active jobs in a shared-spooling environment.
PSETUP	U	Print the page layout of one or more pages.
PVARY	V	Dis/enable one of the following: - exit routines - the task trace - dynamic classes - VSE/POWER messages for console display. - Modify the max number of concurrent SAS tasks
PXMIT	X	Route commands to another node or to an application program that controls the writing of spooled output records to a device.

Authorization to Use Commands

All VSE/POWER operator commands of Table 8 on page 192 are available to the central operator. Non-central operators or cross-partition command facilities are authorized to use only a certain subset of these commands. The following summarizes where to find the command subset that can be used by different operator types.

Own node RJE operator
VSE/POWER "Terminal Operator Commands" described in *VSE/POWER Remote Job Entry*.

Own node cross-partition user or user console
Commands of PWRSPL FUNC= command table described in *VSE/POWER Application Programming*.

Remote node central operator
Commands of first authorization table described in *VSE/POWER Networking*.

Remote node RJE operator
Commands of second authorization table described in *VSE/POWER Networking*.

Remote node cross-partition user or user console
Commands of second authorization table described in *VSE/POWER Networking*.

General Format of the Commands

A VSE/POWER operator command may have a maximum length of 130 bytes and consists of:

1. A command verb (operation field). You can specify the full command verb or just its short form, if there is one. If you want to forbid usage of the short form for certain commands on your system, consider specifying the SET LONGCMD autostart statement (see "SET: Setting VSE/POWER Startup Control Values" on page 473).
2. None, one, or more operands (operand field).

Coding Rules for VSE/POWER Commands

The following rules apply when coding VSE/POWER commands:

- Operands can start in any column after the command code.
- You must use a single blank space to separate the operand from the command. This blank space is not shown in the syntax of the commands.
- A comma separates one operand from another operand. This comma must be coded. The comma is shown in the syntax of the commands. You may *not* use any other separators.
- A blank ends the operand(s) and, thereby, the command.
- Keyword operands consist of a keyword, followed by an equal sign, followed by a value or list of values. If you specify a list of values, this list must be enclosed in parenthesis, and the specified values must be separated from each other by commas.
- Positional operands must be coded in a specific location relative to other operands.

- After the first keyword operand only further keyword operands- no positional operands- should be specified.
- Parentheses that can enclose several optional values may be omitted when there is only one such value specified.
- Specification of a user ID

  Various statements and commands require the specification of a user ID, 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 user_id. A user in this context can be one of the following:

  Type of User
  Explanation / Format of Your Specification

  A remote user
  For user_id, specify the number of the destination (RJE) terminal as defined to VSE/POWER. Specify this in the form R000 - R250, or 000 - 250.

  Note: R000 or 000 is the special remote ID of the "central operator", which means that no user ID is shown in PDISPLAY at all, making LST/PUN queue entries selectable for local list/punch tasks without the VM operand.

  A VSE/ICCF user
  For user_id, specify the four-character ID by which the destination user is known to VSE/ICCF.

  A VM CMS user
  For user_id, specify the ID used to define the destination user to VM.

  A user at another node
  For user_id, specify the ID used to define the destination user to the controlling system at the other node.

  A logical user
  For user_id, specify the logical name which is defined in the applicable subsystem or at the destination node. Do not use R000 - R250, or 000 - 250 as logical names.

  Reserved **logical user ID**
  - 'LOCAL' or 'SYSTEM' when specified as user ID alteration operand by the PALTER command
  - 'ANY' as target user ID of output queue entries, free for general spool-access support GET retrieval. For details, see *VSE/POWER Application Programming*.

  An ID or logical name specified for user_id cannot be any longer than eight (alphameric) characters.
- For VSE/POWER an *alphameric* character consists of the following:
  - Any letter of the alphabet
  - Any of the numerals 0 through 9
  - The number (#) sign
  - The dollar ($) sign
  - The commercial add (@) sign
  - The punctuation marks hyphen (-), period (.), and slash (/).

Note: If a keyword operand occurs more than once, VSE/POWER uses the specification supplied with the operand repeated last.

PACCOUNT: Emptying the VSE/POWER Account File

The command empties the VSE/POWER account file and, optionally, saves the processed account file records.

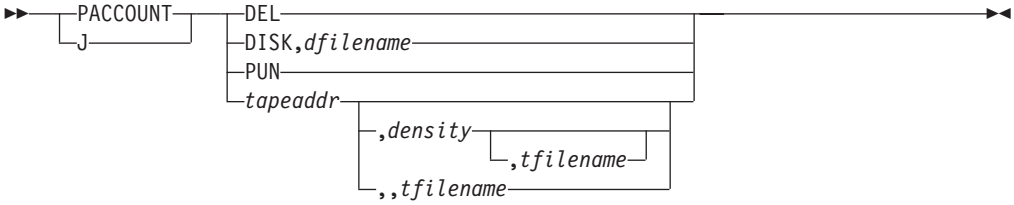
VSE/POWER issues a message whenever the account file is 80 percent full. It indicates, again by message, when the account file has been saved. You receive also a message if a VSE/POWER task requires space for writing an account record and no such space is available.

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.

For information on tape handling, see "Tape Processing with VSE/POWER" on page 151.

Note: For the format of account records saved on disk, tape or in the punch queue, the details are found in *VSE/POWER Application Programming*, chapter "Account-File-Full Condition and Record Format with or Without Prefix."

Format of the Command



DEL
To delete account records and clear the file.

DISK,dfilename
To write and save the contents of the account file onto disk. The file which is used to save the 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 in the DLBL statement which defines this sequential (SD) disk file to the system. Do not include 'VSAM' as parameter 4 (type of file). PACCOUNT does not support VSAM SAM ESDS as an output file. An ASSIGN statement for the logical unit chosen in the EXTENT statement must also be given before VSE/POWER startup.

PUN
To spool the contents of the account file to the punch queue with these values:
Priority 1
jobname PACCOUNT
Output class P
Disposition H

tapeaddr
The address of the tape drive to which the file is to be written. Specify this address in the form cuu. The tape OPEN/CLOSE/EOV handling will be done by VSE/SAM.

density
The mode setting that applies to the tape drive which is to be used. For a list see the table "Mode Settings for Tapes" of the ASSGN statement in *z/VSE System Control Statements*.

At completion of the PACCOUNT function with user specified density, VSE/POWER resets the density specification to the standard one defined at IPL or by a later permanent ASSGN statement.

tfilename
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. The tape is created using VSE/SAM support to write tape labels and handle the mounting of tapes.

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: Activating 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/receivers.

Format of the Command

PNET
To address the VSE/POWER networking support.

node_id
The ID 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. VSE/POWER activates, at the same time, the tasks RV1 through RV7.

JOB | OUT
Specify JOB if the named transmitter/receiver task is to be used for jobs.

Specify OUT if the named transmitter/receiver task is used for list/punch output.

Example of the PACT Command

PACT PNET,NODEA,TR3,OUT
Activates the output-transmitter task 3 for node NODEA.

PALTER: Alter Entry Attributes or Partition Classes, or Segment Output

Use the PALTER command to change:

- the attributes of a VSE/POWER job or output entry, for example, the priority or the disposition (Format 1). A change of the class, destination, disposition, or priority places this job/output at the end of the specified class or priority group.

  In general, you *cannot* alter the attributes of a job/output that is being processed (DISP=*) with this exception:

  – the 'number_of_copies' attribute can be altered even for output entries in DISP=*, provided this output is not being transmitted to a 3790 workstation (with PDIR=FMH2).

  – the 'original disposition' (see ORGDP= in a FULL=YES display) can be altered from D->K or K->D even for reader queue entries in DISP=* state.

  A job's name, class, disposition, and priority can be different in the input and output queues. You specify job-related attributes in the * $$ JOB statement. You may override output-related attributes in a * $$ LST or * $$ PUN statement.

  Note: A central operator PALTER command is confirmed on the console by

  ```
  1R88I  OK : 6 ENTRIES PROCESSED BY PALTER,LST,*SSL,DISP=K
  ```

  for example, thereby indicating the number (6) of modified queue entries together with the corresponding operator command.

- the input classes of a **static** partition (Format 2).

  You can change the assignment of classes to an active **static** partition under control of VSE/POWER without stopping the partition and without releasing the assignments for the partition.

- the state of job output from "in creation" to "queued". See also "Life Cycle of VSE/POWER Queue Entries" on page 30.

  With the SEGMENT=IMM|PAGE, you can request that output being created by executing jobs be segmented immediately or on the next page boundary, thus making it available for processing in the LST/PUN/XMT queue.

For VSE/POWER queues addressable by the PALTER command, refer to Table 9 on page 225.

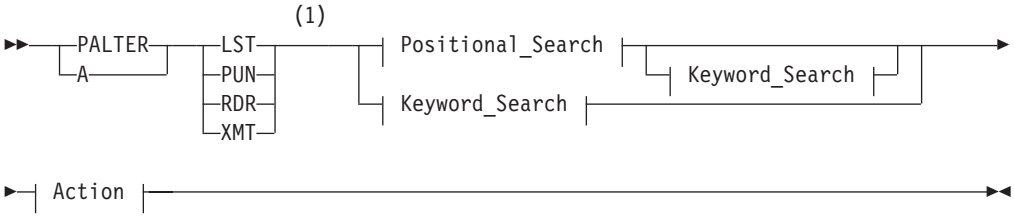
VSE/POWER accepts a PALTER command with up to 14 operands.

Unlike the PRELEASE command, the PALTER command **does not change the number** of the job in the reader queue when the disposition is changed to D or K. Therefore, if a job is made dispatchable more than once for execution, the resulting output queue entries have the same jobname and jobnumber. For unique command selection of these entries, use the CQNUM=nnnnn search operand.

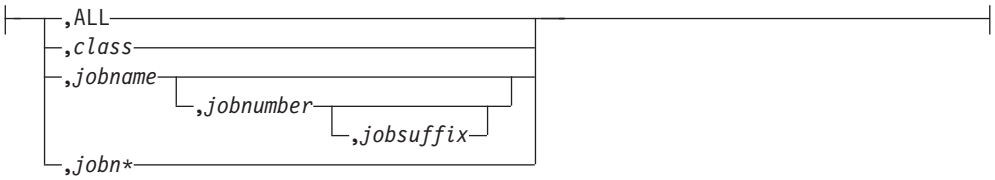
The formats of the PSTART command are as follows:

- Format 1: Altering Attributes of Queue Entries in Physical Queues
- Format 2: Altering Class Assignments for Static Partitions
- Format 3: Altering to Segment In-Creation Job Output

Format 1: Altering Attributes of Queue Entries in Physical Queues

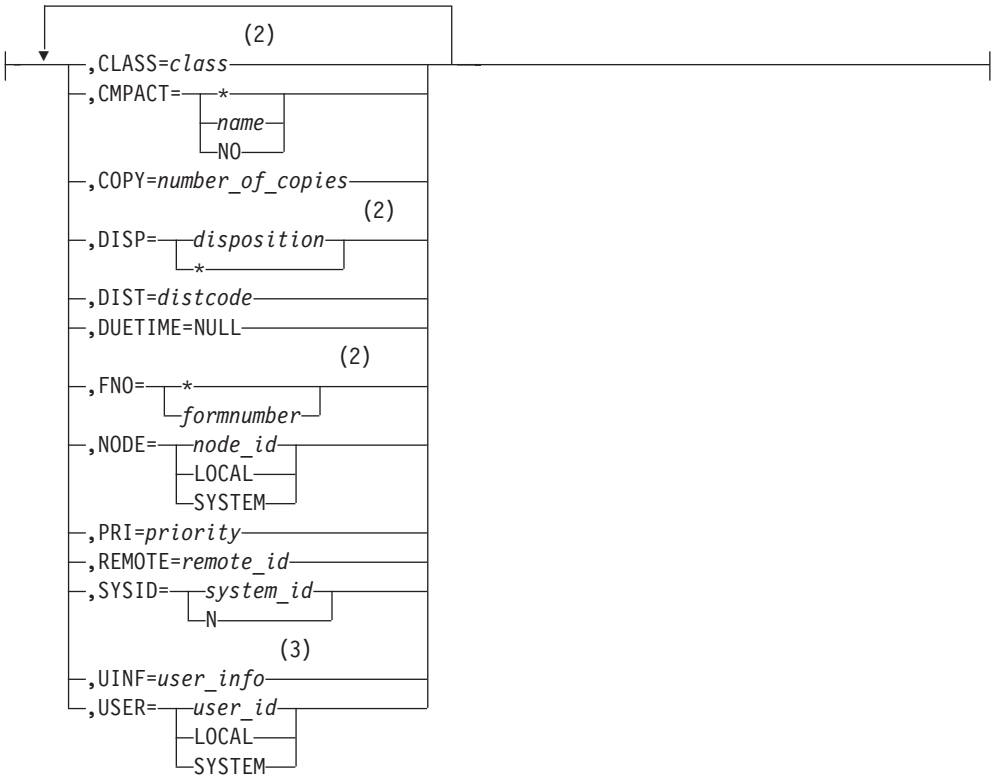


Positional_Search:



Keyword_Search:

Action:



Notes:

1 RDR/LST/PUN/XMT queues are called physical queues in contrast to the logical CRE/DEL queues.

2 A change of this operand may change the sequence of the queue entries. The altered queue entry is added to the appropriate class chain. If, for example, the changed disposition of the queue entry is D or K, the queue entry is added into the dispatchable class chain; if the changed disposition is L or H, the queue entry is added to the non-dispatchable class chain. Queueing is performed based on priority. Output queue entries with the same forms are grouped together.

3 If `user_info` contains spaces or single quotes (or apostrophes), it must be enclosed in single quotes. An embedded single quote must then be represented by two successive single quotes.

Specify the queue for which the command is to be executed, in the form:

LST
to alter the list queue.

PUN
to alter the punch queue.

RDR
to alter the reader queue.

XMT
to alter the transmission queue.

Positional Search Operands

ALL
Specify ALL to alter either specific attributes 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.

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. Specification of this operand expedites the access path to the VSE/POWER queue file.

jobname
The name by which a specific job is known to VSE/POWER. The jobname may consist of 2 to 8 alphameric characters.

Specifying PALTER with 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 belonging to the same group of 127 segments are also changed. See "Count-Driven Output Segmentation" on page 117. If you specify also a job suffix, the alter request applies only to the specified segment of the named output entry.
2. If you specify a jobname with one of the remaining search operands (CCLASS to XMTID), VSE/POWER performs the requested change for all occurrences of this name.

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 display this number.

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 job suffix S=nnn, issue a PDISPLAY command for the output entry.

jobn*
Specifies the queue entries whose name begin with the specified characters. As search argument, you can specify for *jobn* one to eight characters.

Keyword Search Operands

CCARDS=|<|>nnnnnnnnn
indicates that only reader and punch queue entries are to be addressed whose current total card count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified card nnnnnnnnn value, which must be greater than zero. A card count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P). When CCARDS is specified for the LST queue or for list type entries of the XMT queue (I=L), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

   ```
   1R52I LIST QUEUE NOTHING TO ALTER, or
   1R52I NOTHING TO ALTER
   ```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For reader or punch type entries in the XMT queue the 'LINES' column actually presents statements, that means cards. This is the value against the CCARDS operand is compared.
4. For active entries (DISP=*) in the PUN or XMT queue, 'cards left to be processed' are displayed, although the entries were selected according to their total card count, which is not displayed then.

CCLASS=current_class
indicates that the requested change is to be done if the class currently assigned to the queue entry (or entries) matches the specified class. This operand overwrites a possible specification of the positional 'class' operand. Specification of either 'class' or the 'CCLASS=' operand expedites the access path to the VSE/POWER queue file.

CDISP=*|current_disposition
indicates that the requested change is to be done if the disposition currently displayed for (as assigned to) the queue entry (or entries) in the specified queue matches the specified disposition.

For disposition, you may specify an asterisk (*) to have VSE/POWER select VSE/POWER jobs that are being processed.

CDUE=*
indicates that all jobs are to be altered for which time event scheduling operands have been specified. Such jobs can be either in the RDR or the XMT queue.

CFNO=*|current_form_number
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. One can specify '*' to process those queue entries which have no form number, or the four-character form number of the queue entries to be processed.

CPAGES=|<|>nnnnnnnnn
indicates that only list queue entries are to be addressed whose current total page count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified page nnnnnnnnn value, which must be greater than zero. A page count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the LST queue or for list type entries of the XMT queue (I=L). When CPAGES is specified for

the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

```
1R52I READER QUEUE NOTHING TO ALTER or
1R52I NOTHING TO ALTER
```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For list type entries in the XMT queue the standard display presents 'LINES'. Use the 'P=n...n' page count of the FULL=YES display, when you select entries by the CPAGES operand.
4. For active entries (DISP=*) in the LST or XMT queue, 'pages left to be processed' are displayed, although the entries were selected according to their total page count, which is not displayed then.

CPRI=current_priority
indicates that the requested change is to be done if the priority currently assigned to the queue entry (or entries) matches the specified priority.

CQNUM=nnnnn
indicates that the requested change is to be done if the queue entry's internal queue number matches the specified `nnnnn` value.

This unique number assigned by VSE/POWER can be made visible by the PDISPLAY command QNUM=nnnnn display field. See also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257. Use the CQNUM search operand if more than one queue entry has the same jobname and jobnumber and the entries otherwise cannot be distinguished by other attributes for command selection. You can specify 1 to 5 digits.

Note: Only numbers within the range indicated by `TOTAL NUMBER OF QUEUE RECORDS` (see Figure 7 on page 46) are accepted for searching. Only one queue entry can satisfy the search criterion.

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 (<)                  ──►  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

```
1) mm/dd/yy or dd/mm/yy
2) mm/dd/yyyy or dd/mm/yyyy
```

Do not specify mm greater than 12, dd greater than 31, and yy or yyyy greater than the current year.

Notes:

1. If you specify a 2 digit yy-year, VSE/POWER expands it for internal processing and comparison to a 4 digit year according to the 'fix-88-window' rule:

```
if yy is greater than (>) 88, then 19yy
if yy is less than or equal (<=) 88, then 20yy
```

2. The format of the system date must not be changed after VSE/POWER has been initialized or between warm starts of VSE/POWER (see "Functions Not Expecting Date Format Changes" on page 128).
3. In a shared spooling environment, all systems must have the same date format.

CSYSID=current_system_id
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.

CUINF=current_user_info
indicates that the requested change is to be done if the queue entry's user information matches the specified current_user_info value. User information is provided to a queue entry by the * $$ JOB/LST/PUN JECL statements or by the PALTER command, and can be made visible by a PDISPLAY ...,FULL=YES request, which presents this information left-justified with trailing blanks, all enclosed in quotes via the U='...' display field (see also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257). When user information contains

1. neither embedded blanks nor commas and is displayed, for example, as

   U='PETER ', it can be searched for using
   CUINF=PETER, or
   CUINF='PETER', or
   CUINF='PETER '
2. an embedded blank (or comma) and is displayed, for example, as

   U='PETER BOY ', it can be searched for using
   CUINF='PETER BOY'
3. an embedded blank (or comma) and a single quote, shown for example as

   U='PETER'S BOOK ', it can be searched for using
   CUINF='PETER''S BOOK'

The CUINF specification (when enclosed in single quotes) can contain trailing blanks up to the field limit of 16 characters.

CUSER=current_user_id
indicates that the requested change is to be done if the queue entry's "from user" or "to user" matches the ID given in this operand. For a definition of user ID, see Chapter 1, "Understanding Syntax Diagrams," on page 1.

FNODE=from_node_id
indicates that the requested change is to be done if the queue entry's "from node" name matches the node name specified by *from_node_id*. However, VSE/POWER does not verify that the specified node name is defined in the network definition table.

FUSER=from_user_id
indicates that the requested change is to be done if the queue entry's "from

user" ID matches the user ID specified by *from_user_id*. For a definition of user ID, see Chapter 1, "Understanding Syntax Diagrams," on page 1.

Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user ID of LOCAL. It does not cause a change of attributes of entries originated by local users.

TNODE=to_node_id
indicates that the requested change is to be done if the queue entry's "to node" ID matches the node name specified by *to_node_id*. Up to eight alphameric characters can be specified for *to_node_id*. This operand is valid only if queue entries residing in the transmission queue are to be altered.

TUSER=to_user_id
indicates that the requested change is to be done if the queue entry's "to user" ID matches the user ID specified by *to_user_id*. For a definition of user ID, see Chapter 1, "Understanding Syntax Diagrams," on page 1.

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 ID of LOCAL. It does not cause a change of attributes of entries destined for local users.

XMTID=R|L|P
This operand is effective only if your command applies to the XMT queue; for other queue types the operand is ignored. For XMT queue, you can address one of the three transmission subqueues presented as R=Reader, L=List, P=Punch by the 'I' identification column of a PDISPLAY XMT command display line.

The operand indicates that the requested change is to be done if the queue entry's 'I'-identification matches the XMTID value given in this command. For a PDISPLAY XMT example, see page 261.

Action Operands

CLASS=class
specifies the new class that is to be assigned to the named VSE/POWER queue entry. "class" may be specified as any alphameric character from A to Z or 0 to 9.

CMPACT=*|name|NO
specifies whether data compaction is to be performed for the transmission of job output to an SNA terminal. 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.

The operand is accepted only for job output residing in the local list queue.

COPY=number_of_copies
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.

Note: If you terminate 3800 printing by the PSTOP command while copy modification is active with a current copy-index count (maintained by the 3800 printer support outside of VSE/POWER) and if you increase the number of copies by the PALTER command, and if thereafter printing is resumed again by the PSTART command, the 3800 printing subsystem will continue with the previously active copy index.

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
```

If by this alteration the affected job remains in its current queue, the new disposition means the

- local disposition when the job resides in the RDR/LST/PUN queue
- transmission disposition when the job resides in the XMT queue

However, if the job is moved at the same time by this alteration from a local queue to the XMT queue or vice-versa, the new disposition means the

- transmission disposition when it is a move to the XMT queue
- local disposition when it is a move to the RDR/LST/PUN queue.

For details, see also “Disposition in the Transmit Queue” in *VSE/POWER Networking*.

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. This disposition is then presented by the ’ORGDP=’ field in the date-display line of a `PDISPLAY ...,FULL=YES` request.

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 a reader queue entry being processed (DISP=* in display) you can alter the ’original disposition’ (see ORGDP= in a FULL=YES display) from D->K or K->D when no additional alteration action operand is specified, and when either
  – the operator command qualifies the jobname uniquely by jobnumber or CQNUM=, or

– the command is submitted by Spool Access Direct CTL request.

For further information on dispositions see Appendix A, "VSE/POWER Disposition Codes," on page 497.

DIST=distcode
specifies the new distribution code which is assigned to the affected queue entry. The distribution code consists of up to eight alphameric characters and is passed for a VM writer/punch task by the CP CLOSE command to VM.

You can alter the distribution code of queue entries in the LST and PUN queue. You also can alter the distribution code of output queue entries in the XMT queue.

DUETIME=NULL
specifies that the due date is to be nullified. That is, all specifications of time event scheduling are ignored and lost.

If the due date of a job with disposition D or K has not yet expired, this job becomes now immediately dispatchable.

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=node_id | LOCAL | SYSTEM
specifies that the new final destination must be changed.

- For node_id, specify the new node ID. The node ID you use must be defined in the active network definition table.
- LOCAL or SYSTEM, to tell VSE/POWER that the current destination ID must be replaced by the ID 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.

  Note: Whenever a job/output is lifted from
  – a local queue to the XMT queue, it will acquire the transmission disposition
    - as specified by the JECL TDISP= or PWRSPL SPLOTDP operand, or
    - as preserved if the entry has ever resided in the XMT queue
  – the XMT queue to a local queue, it will acquire the local processing disposition
    - as specified by the JECL DISP= or PWRSPL SPLDDP operand, or
    - as preserved if the entry has ever resided in a local queue

  This may be overridden by specifying both the NODE= and DISP= operands in the same PALTER command. For details, see the DISP operand of the applicable command.

PRI=priority
specifies the new priority (an integer from 0 to 9, where 9 is the highest priority) of the named VSE/POWER queue entry(ies).

REMOTE=remote_id
Your specification, the new remote ID, applies only to output. The specification is ignored if given for an input queue entry.

For remote_id, specify the ID of the terminal to which the applicable output is to be routed. Specify this ID in the form Rnnn or just nnn (where nnn = the remote ID 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=system_id|N
The operand applies to a shared spooling environment. Specify:

- For system_id, a new system ID 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.

UINF=user_info
For `user_info`, specify up to 16 characters or blanks. If the character string includes a blank or comma, the entire string must be enclosed in a pair of single quotes; otherwise, the first blank or comma is interpreted as a delimiter by VSE/POWER. If you are nesting single quotes inside single quotes, you must specify them as two adjacent single quotes, as shown in "Examples of the PALTER Command" on page 211.

It is recommended to specify characters whose hexadecimal representation is not affected by uppercase conversion. All values are converted to uppercase, as described in "Uppercase Conversion of Characters in JECL/JCL Statements" on page 405.

The specified character string replaces whatever character string was specified in the * $$ JOB statement (if any) for the job, or whatever was specified in the * $$ LST|PUN statement for the output (or passed from * $$ JOB to the output within the UINF= (also USER=) operand.

See also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257 for the displayed U='user_info', which can be modified by the command

```
PALTER ...,UINF=user_info
```

Note: This change can be requested only for inactive (not DISP=*) entries.

USER=user_id|LOCAL|SYSTEM
Use the operand to change the user ID of the affected queue entry (or entries). Specify:

- For user_id, the ID to be used by VSE/POWER. This ID must be known to your VSE system or to the target node, whichever applies. Specify a remote terminal in the form Rnnn or nnn (where nnn = the remote ID as defined for VSE/POWER table generation).
- LOCAL or SYSTEM to have VSE/POWER set the user ID 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. For more information, see also the PSTART command "Format 1: Processing Disk-Spooled Output" on page 359.

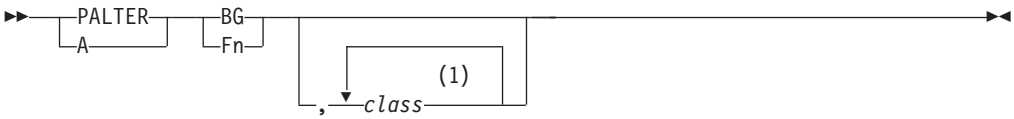
Note: VSE/POWER does **not** allow to assign a TO user-ID to an entry in the RDR queue by the PALTER RDR,...,USER= command. This is only acceptable for job entries residing in or being altered to the XMT queue, which might be

transmitted to another (OS/390 MVS or VM) system. Therefore, if an XMT queue entry with a TO user-ID is requeued into the RDR queue, the existing TO user-ID is not displayed.

When altering a RDR queue entry to the XMT queue, use the NODE and USER operands in the following sequence:

```
PALTER RDR,...,NODE=node_id,USER=user_id
```

Format 2: Altering Class Assignments for Static Partitions



Notes:

1 You may specify up to four input classes.

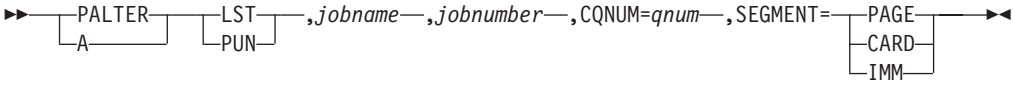
Use this command format to change the class assignment for a VSE/POWER controlled **static** partition.

BG | Fn
The static partition id.

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 (A-Z) or the partition number (0-9). If you do not specify any class(es), only partition-related jobs can run in the specified partition.

For dynamic partitions, classes are defined in the Dynamic Class Table and only there they can be altered. Any attempt to alter the processing class of a dynamic partition will be rejected and the message 1R52I will be displayed.

Format 3: Altering to Segment Job Output In Creation



Use this command to alter the state of job output from "in creation" to "queued" before the job has completed. The output queue entry is made available for processing in the LST/PUN/XMT queue as if it had been created by count-driven (RBS) segmentation (see "Output Segmentation" on page 117). The same segmentation function can also be requested by the PSEGMENT command (see "PSEGMENT: Segmenting Output Spooled by Execution Writer Tasks" on page 355).

LST
Specify this queue if 'PDISPLAY A' presents 'LINES SPOOLED' for the named queue entry or if 'PDISPLAY CRE' presents 'L' in the type indicator column ('I')

PUN
Specify this queue if 'PDISPLAY A' presents 'CARDS SPOOLED' for the named queue entry or if 'PDISPLAY CRE' presents 'P' in the type indicator column ('I')

CQNUM=qnum
Specify the current (internal) queue entry number as presented by 'QNUM=' in 'PDISPLAY A' or by the QNUM column in 'PDISPLAY CRE' for the named queue entry

SEGMENT=PAGE|CARD|IMM
Specify the desired segmentation boundary

- PAGE - for LST output, to segment at the next page boundary
- CARD - for PUN output, to segment at the next data statement
- IMM - for LST|PUN output, to segment immediately

Notes:

1. 'PAGE' and 'CARD' are used synonymously.
2. The 'PALTER...SEGMENT=IMM' command produces output with a record count of zero if the command is requested again before the execution writer task has spooled a record in the meantime.
3. The command may be issued by a local operator or by a SAS program and initiate segmentation, which is then processed by the execution writer task.
4. If the Spool-Access Support (SAS) program issues a *direct* CTL (PALTER...SEGMENT) request, the direct option is ignored.

Returned messages and codes:

- When the command is executed successfully, message 1R88I OK is returned to the local operator and RC/FDBK=00/01 to the SAS program.
- When the segment request has been finally processed (e.g., page boundary reached and segment created) by the execution writer task, the task issues

  ```
  1Q53I OUTPUT SEGMENTED FOR jobname jobnumber suffix partition-id, cuu
  ```

- When the specified queue entry cannot be found due to incorrect specification or because the queue entry is not being created by an execution writer task or has meanwhile been completed, message

  ```
  1R88I NOTHING TO ALTER
  ```

  is returned to the local operator and to the SAS program.
- When the queue entry is spooled with DISP=I|T, the command is rejected with message

  ```
  1R9BI commandcode SEGMENT REQUEST IGNORED FOR DISP=I|T
  ```

- When no DBLK group is available at all, meaning that the DBLK group cushion is used up, the command is rejected with

  ```
  1R9BI commandcode SEGMENT REQUEST IGNORED DUE TO EMPTY DBLKGP CUSHION
  ```

Interaction with Count-Driven (RBS) Segmentation

If RBS segmentation was previously requested by the '* $$ LST|PUN RBS=nnnn' statement for the output addressed by the 'PALTER...SEGMENT' command, the specified output entry will be segmented not on the desired RBS boundary but rather at the next page or card, or even immediately. This depends on the option of the PALTER command. Therefore, the created segment will not show the RBS number of pages or cards. The subsequent segment will be created using the original RBS value.

Collecting Specification Values for Command Example

Output of an executing job is spooled by VSE/POWER execution writer tasks according to the intercepted spooled device, such as FEE of partition F2, for example. For these execution writer tasks, the 'PDISPLAY CRE,PART' or 'PDISPLAY A,PART' display lines present all information needed to specify the requested command operands as shown below.

```
pdisplay cre,part,f2
F1 0001 1R4BI   CREATE QUEUE  C I  LINES B DBGP   QNUM  TASK    OWNER
F1 0001 1R4BI  CICSICCF 01855 A L   1436   000003 00519  F2 FEE JOB=CICSICCF
- - -
pdisplay a,part,f2
F1 0001 1R48I   F2,FEC,L2,  CICSICCF,01855,2
F1 0001 1R48I   F2,FEE,,  CICSICCF,01855,A      1436 LINES SPOOLED,QNUM=00519
- - -
PALTER LST,CICSICCF,1855,CQNUM=519,SEGMENT=PAGE
```

Note: To determine whether LST or PUN output is created, use L|P information in column 'I' of message 1R4BI or the LINES or CARDS information of message 1R48I.

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<01/31/01,DISP=D
Searches the list queue for all entries that are older than January 31, 2001, and changes their disposition to D. Or with 4-digit years:

PALTER LST,CRDATE<01/31/2001,DISP=D
Searches the list queue for all entries that are older than January 31, 2001, 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=4
Changes all class P jobs in the LST queue to class 4.

A F1,ABC1
Jobs with a class specification of A, B, C, and 1 can now run in partition F1.

A F1 Only those jobs with partition-related input class 1 can now run in partition F1.

A LST,HOTJOB,UINF=PETER

Defines or changes the 16-byte user information for the list entry HOTJOB, so that PDISPLAY ...,FULL=YES displays

```
U='PETER           '
```

A LST,HOTJOB,UINF='PETER''S BOOK'

Defines or changes the 16-byte user information for the list entry HOTJOB, so that PDISPLAY ...FULL=YES displays

```
U='PETER'S BOOK    '
```

PBRDCST: Sending Messages

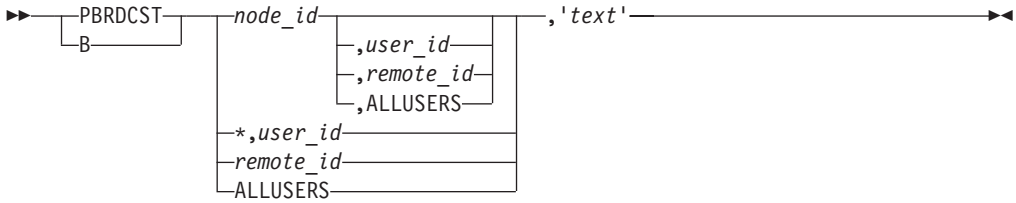
The command enables you to send (or broadcast) a message to:

- A specific user or to all interactive users in one of these environments:
  - CMS under VM if your virtual VSE system includes networking support and can link to RSCS of VM.
  - TSO under z/OS
  - VSE/ICCF

  The user(s) may be linked to your own node or to another node.
- A remote terminal.

  A message routed to a remote terminal is displayed as soon as this terminal is ready to receive the message.

Format of the Command



node_id
Specify the name of the destination node if your message is to be sent to another node in the network.

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 will be lost.

user_id
Specify the ID of the user to whom the message is to be routed.

If you do not specify a user ID, VSE/POWER directs your message to the central operator of the specified node.

***,user_id**
Specify an `asterisk` (*) instead of a node name if the message is to be sent to a non-RJE user at your own node; (a user of the same VSE system).

remote_id
If your message is to be sent to an RJE terminal connected to your own system, specify for *remote_id* this terminal's ID (in the form nnn or Rnnn). Specify 0 for *remote_id* 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.

You may also specify 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 is lost if ALLUSERS is not defined as the name of a user at the destination node.

'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 terminal 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 ID 150.

PBRDCST ALLUSERS,'RJE STARTS TOMORROW AT 800 HRS.'
Places this message into the ALLUSERS message queue.

B NODEB,'WHEN WILL YOU SHUT DOWN.'
Broadcasts this message to the 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.

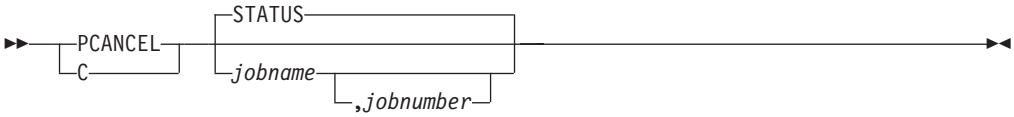
PCANCEL: Terminating a Status Display or Canceling a Job

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 the queue file in storage
- Further processing of a job. For the disposition changes of the job after a PCANCEL command has been issued, see the disposition changes with the PFLUSH command "PFLUSH: Canceling the Processing for a Queue Entry" on page 294.

For terminating jobs that have established an 'ON $CANCEL' condition, see "Interaction with z/VSE Conditional Job Control Language" on page 108.

Format of the Command



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 (by PXMIT) has no effect because processing of the transmitted PDISPLAY command is completed by the time the transmitted PCANCEL command can be processed.

jobname
name of a specific job 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. When SLI is in process during cancellation, message 1QC1I RC=0005 may be issued in addition.

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: Deleting Queue Entries or Messages

The command removes one or more VSE/POWER jobs from the specified queue or deletes messages. 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 any task are not affected by this command.

Note: A central operator PDELETE command is confirmed, for example, on the console by

```
1R88I  OK : 6 ENTRIES PROCESSED BY PDELETE,LST,*SSL
```

indicating that the corresponding command has requested 'delayed freeing' for 6 queue entries and has posted the initialization/termination task for 'final freeing' of those entries.

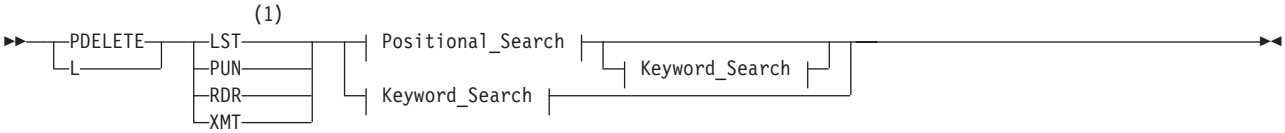
For VSE/POWER queues addressable by the PDELETE command, refer to Table 9 on page 225.

VSE/POWER accepts a PDELETE command with up to 14 operands.

The formats of the PDELETE command are as follows:

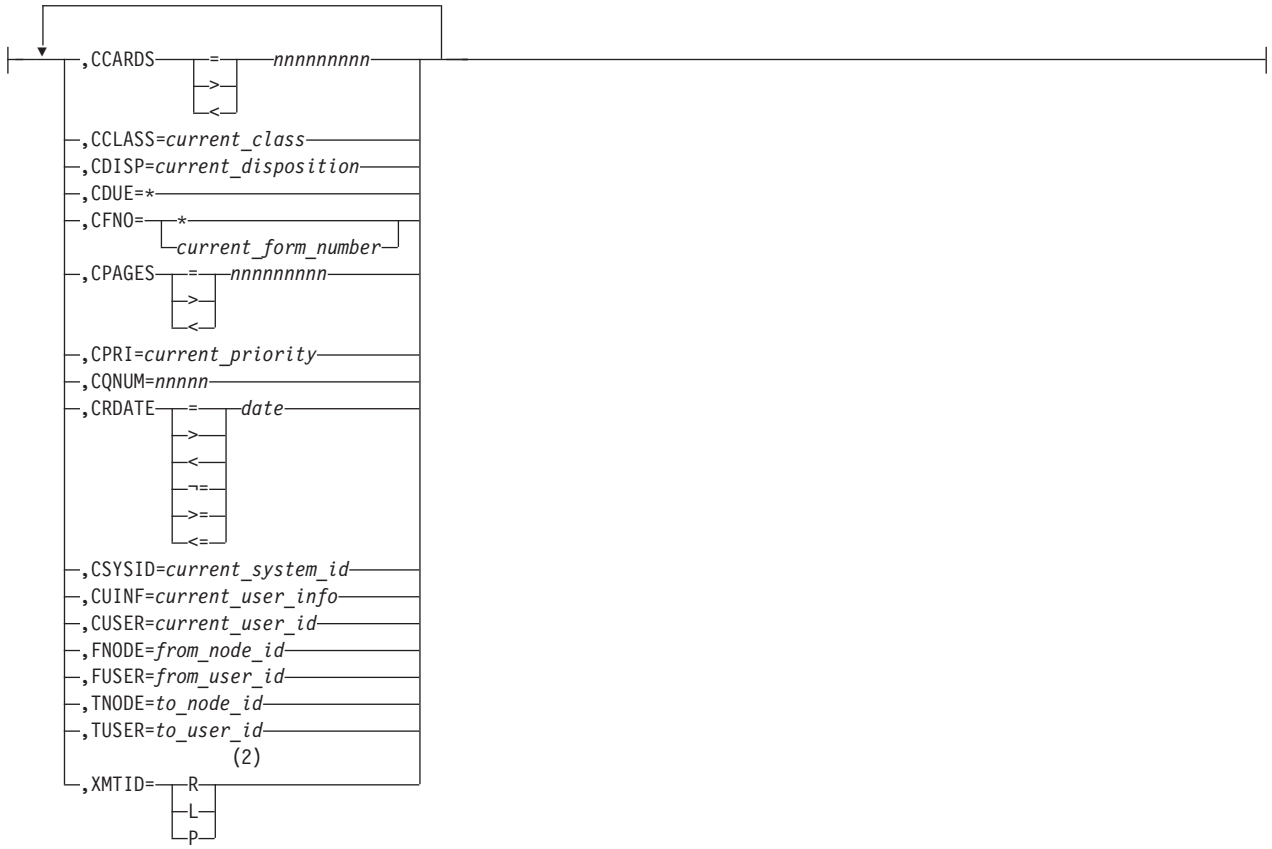
- Format 1: Deleting Queue Entries in Physical Queues
- Format 2: Deleting Messages
- Format 3: Clearing the VSE/POWER FCB Table

Format 1: Deleting Queue Entries in Physical Queues



Positional_Search:

Keyword_Search:



Notes:

1 RDR/LST/PUN/XMT queues are called physical queues in contrast to the logical CRE/DEL queues.

2 XMTID is meaningful only if XMT is specified.

Specify the queue for which the command is to be executed, in the form:

LST
to address the list queue.

PUN
to address the punch queue.

RDR
to address the reader queue.

XMT
to address the transmission queue.

Positional Search Operands

ALL
Specify ALL to delete either all VSE/POWER jobs in the named queue or just a certain group of jobs in the queue as defined by one or more additional keyword-type search operands.

Note: a request to really delete 'ALL' entries of a named queue will trigger command confirmation by message 1QZ3D.

class
specifies that all VSE/POWER jobs of a certain class are to be deleted from the specified queue. Any alphameric character from A to Z or 0 to 9 may be specified.

jobname
specifies the name by which the entry is known to VSE/POWER. The jobname may contain 1 to 8 alphameric characters.

Specifying PDELETE with 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 also are deleted. If you specify also a job suffix, the delete request applies only to the specified segment of the named job.

jobnumber
specifies the jobnumber assigned to the job by VSE/POWER. 1 to 5 digits may be specified. Use the DISPLAY command to determine this number.

jobsuffix
specifies the 1-3 digit job suffix associated with the queue entry to be deleted. Specify this operand 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 jobnumber. To get the correct job suffix S=nnn, issue a PDISPLAY command for the output entry.

jobn*
specifies that all VSE/POWER jobs whose jobnames begin with the specified characters are to be deleted. Up to eight alphameric characters may be specified for *jobn*.

Keyword Search Operands

CCARDS=|<|>nnnnnnnnn
indicates that only reader and punch queue entries are to be addressed whose current total card count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified card nnnnnnnnn value, which must be greater than zero. A card count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P). When CCARDS is specified for the LST queue or for list type entries of the XMT queue (I=L), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

   ```
   1R52I LIST QUEUE NOTHING TO DELETE, or
   1R52I NOTHING TO DELETE
   ```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For reader or punch type entries in the XMT queue the 'LINES' column actually presents statements, that means cards. This is the value against the CCARDS operand is compared.
4. For active entries (DISP=*) in the PUN or XMT queue, 'cards left to be processed' are displayed, although the entries were selected according to their total card count, which is not displayed then.

CCLASS=current_class
indicates that the requested deletion is to be done if the class currently assigned to the queue entry (or entries) matches the specified class. This operand overwrites a possible specification of the positional 'class' operand. Specification of either 'class' or the 'CCLASS=' operand expedites the access path to the VSE/POWER queue file.

CDISP=current_disposition
indicates that the requested deletion is to be done if the disposition currently assigned to the queue entry (or entries) matches the specified disposition.

CDUE=*
indicates that all jobs are to be deleted for which time event scheduling operands have been specified. Such jobs can be either in the RDR or the XMT queue.

CFNO=*|current_form_number
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. One can specify '*' to process those queue entries which have no form number, or the four-character form number of the queue entries to be processed.

CPAGES=|<|>nnnnnnnnn
indicates that only list queue entries are to be addressed whose current total page count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified page nnnnnnnnn value, which must be greater than zero. A page count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the LST queue or for list type entries of the XMT queue (I=L). When CPAGES is specified for

the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

```
1R52I READER QUEUE NOTHING TO DELETE or
1R52I NOTHING TO DELETE
```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For list type entries in the XMT queue the standard display presents 'LINES'. Use the 'P=n...n' page count of the FULL=YES display, when you select entries by the CPAGES operand.
4. For active entries (DISP=*) in the LST or XMT queue, 'pages left to be processed' are displayed, although the entries were selected according to their total page count, which is not displayed then.

CPRI=current_priority
indicates that the requested deletion is to be done if the priority currently assigned to the queue entry (or entries) matches the specified priority.

CQNUM=nnnnn
indicates that the requested deletion is to be done if the queue entry's internal queue number matches the specified `nnnnn` value.

This unique number assigned by VSE/POWER can be made visible by the PDISPLAY command QNUM=nnnnn display field. See also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257. Use the CQNUM search operand if more than one queue entry has the same jobname and jobnumber and the entries otherwise cannot be distinguished by other attributes for command selection. You can specify 1 to 5 digits.

Note: Only numbers within the range indicated by `TOTAL NUMBER OF QUEUE RECORDS` (see Figure 7 on page 46) are accepted for searching. Only one queue entry can satisfy the search criterion.

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 in relation to the specified date:

```
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

```
1) mm/dd/yy or dd/mm/yy
2) mm/dd/yyyy or dd/mm/yyyy
```

Do not specify mm greater than 12, dd greater than 31, and yy or yyyy greater than the current year.

Notes:

1. If you specify a 2 digit yy-year, VSE/POWER expands it for internal processing and comparison to a 4 digit year according to the 'fix-88-window' rule:

   ```
   if yy is greater than (>) 88, then 19yy
   if yy is less than or equal (<=) 88, then 20yy
   ```

2. The format of the system date must not be changed after VSE/POWER has been initialized or between warm starts of VSE/POWER (see "Functions Not Expecting Date Format Changes" on page 128).
3. In a shared spooling environment, all systems must have the same date format.

CSYSID=current_system_id
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.

CUINF=current_user_info
indicates that the requested deletion is to be done if the queue entry's user information matches the specified current_user_info value. User information is provided to a queue entry by the * $$ JOB/LST/PUN JECL statements or by the PALTER command, and can be made visible by a PDISPLAY ...,FULL=YES request, which presents this information left-justified with trailing blanks, all enclosed in quotes via the U='...' display field (see also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257). When user information contains

1. neither embedded blanks nor commas and is displayed, for example, as

   U='PETER ', it can be searched for using
   CUINF=PETER, or
   CUINF='PETER' or
   CUINF='PETER '

2. an embedded blank (or comma) and is displayed, for example, as

   U='PETER BOY ', it can be searched for using
   CUINF='PETER BOY ', or
   CUINF='PETER BOY'

3. an embedded blank (or comma) and a single quote, shown for example as

   U='PETER'S BOOK ', it can be searched for using
   CUINF='PETER''S BOOK'

The CUINF specification (when enclosed in single quotes) can contain trailing blanks up to the field limit of 16 characters.

CUSER=current_user_id
requests an affected queue entry to be deleted if the entry's "from user" or "to user" ID matches the ID given in this operand. For a definition of user ID, see Chapter 1, "Understanding Syntax Diagrams," on page 1.

FNODE=from_node_id
requests an affected queue entry to be deleted if the entry's "from node" name matches the node name specified by *from_node_id*.

FUSER=from_user_id
requests an affected queue entry to be deleted if the entry's "from user" ID matches the ID specified by *from_user_id*.

Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user ID of LOCAL. It does not cause a deletion of entries originated by local users.

TNODE=to_node_id
requests an affected queue entry to be deleted if the entry's "to node" name matches the node name specified by *to_node_id*. This operand is valid only if your specification for queue is XMT.

TUSER=to_user_id
requests an affected queue entry to be deleted if the entry's "to user" ID matches the ID specified by *to_user_id*.

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 ID of LOCAL. It does not cause a deletion of entries destined for local users.

XMTID=R|L|P
This operand is effective only if your command applies to the XMT queue; for other queue types the operand is ignored. For XMT queue, you can address one of the three transmission subqueues presented as R=Reader, L=List, P=Punch by the 'I' identification column of a PDISPLAY XMT command display line.

The operand indicates that the entry is to be deleted if the queue entry's 'I'-identification matches the XMTID value given in this command. For a PDISPLAY XMT example, see page 261.

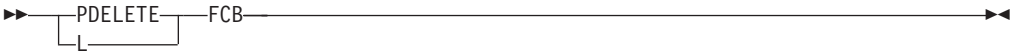
Format 2: Deleting Messages

MSG
Use this set of operands to delete a specific ALLUSERS message or all such messages. You can use the PDISPLAY MSG command to have VSE/POWER display the queued messages and their numbers.

ALL
VSE/POWER deletes all queued ALLUSERS messages. ALL is the default.

number
For *number*, specify the number of the message you want to be deleted by VSE/POWER.

Format 3: Clearing the VSE/POWER FCB Table



FCB
The command clears VSE/POWER's internal FCB table, which is used at job execution time when spooling list output and channel 9/12 posting has to be

done. Specify this operand if VSE/POWER should use new FCB data along with an already existing FCB name specified in an * $$ LST statement.

Hint: Specify the PDELETE FCB command to have VSE/POWER's FCB table deleted without restarting VSE/POWER. Use this command if you change an FCB image and you want to use your existing jobs without changing your original FCB names as specified in the * $$ LST statements.

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<02/15/01 or PDELETE PUN,CRDATE<02/15/2001
Deletes all punch queue entries that are older than February 15, 2001.

PDELETE MSG
Deletes all ALLUSERS messages.

PDELETE LST,ASSM*
Deletes all queue entries in the list queue whose jobnames 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: Displaying VSE/POWER Status

You can use this command to display the following information:

- Various kinds of job- and resource-status information.
- Status information about queue entries on tape or network-related information, if VSE/POWER's networking function is used.
- Characteristics of classes of the currently active Dynamic Class Table.

Examples of and explanations to VSE/POWER produced status reports are given under "PDISPLAY Output Examples" on page 249.

You can specify the PDISPLAY command with up to 14 operands.

The formats of the PDISPLAY command are as follows:

- Format 1: Displaying Information for a Selected Physical Queue
- Format 2: Displaying Information for All Physical Queues
- Format 3: Displaying Information for the Logical Create Queue[13]
- Format 4: Displaying Information for the Logical Deletion Queue[13]
- Format 5: Displaying Information for All Physical and Logical Queues
- Format 6: Displaying Information for the 'nn' largest entries of all physical and logical queues.
- Format 7: Displaying Various Status Information
- Format 8: Displaying Activated Task Status
- Format 9: Displaying Network Definition Table Status (NDT)
- Format 10: Listing the Storage Copy of the Queue File
- Format 11: Displaying Queue Entries Residing on Tape
- Format 12: Displaying the Active Dynamic Class Table
- Format 13: Displaying Exit Information
- Format 14: Displaying Information About Used Autostart Statements
- Format 15: Displaying Information About Spooled Devices
- Format 16: Displaying Internal Information About All Tasks

Formats 1-5 of the PDISPLAY command address the following VSE/POWER queues:

- CRE (Create)
- RDR (Reader)
- LST (List)
- PUN (Punch)
- XMT (Transmission)
- DEL (Deletion)

For a description of the transition of queue entries through these queues, see "Life Cycle of VSE/POWER Queue Entries" on page 30.

13. The Create and Deletion queues are referred to as *logical queues*. They are not physically linked sets of allocated queue entries like the *physical queues*RDR/LST/PUN/XMT but rather simply subsets of allocated queue entries in the 'in-creation' and 'delayed deletion' states, respectively. The entries in these logical queues are located for display purposes by stepping through all allocated queue entries sequentially and examining the state indicator.

The following table provides an overview of how these queues are identified and how they can be

- addressed by queue manipulation commands
- displayed
- used by update or browsing tasks

Table 9. Applicability of Queues by Command and Processing Task

Queue Designation	Type	Selected Queue DISPLAY Possible	"ALL" Queue DISPLAY Selection	"TOTAL" Queue DISPLAY Selection	ALTER DELETE HOLD RELEASE	PSTART Update Task for Entries	Spool-Access BROWSE Entries
CRE	Logical	+	-	+	+[3]	-	+[1]
RDR	Physical	+	+	+	+	+	+
LST	Physical	+	+	+	+	+	+
PUN	Physical	+	+	+	+	+	+
XMT	Physical	+	+	+	+	+	+
DEL	Logical	+	-	+	-	-	+[2]

Notes:

1. Only output of executing jobs selectable by Direct SAS GET BROWSE.
2. Only ongoing SAS browsing as selected in RDR/LST/PUN/XMT queue but deleted in between by command or update task.
3. Only with PALTER...,SEGMENT or PSEGMENT, the CRE queue can be addressed indirectly.

Sequence of Display and Processing in Physical Queues

PDISPLAY presents queue entries in the sequence as they have been created (or have been PALTER'ed, see "Format 1: Altering Attributes of Queue Entries in Physical Queues" on page 199) and have been added to a physical queue of equal entry types, namely

RDR, LST, PUN, or XMT queue,

and within such a queue type according to execution class (related to processing class of tasks), namely

0, 1,..., 9, A, B,..., Z for RDR/LST/PUN queue, or

R(eader), L(ist) and P(unch) indicator for XMT queue,

and within a class of a queue type according to processing disposition, namely

D|K (dispatchable) followed by H|L (non-dispatchable),

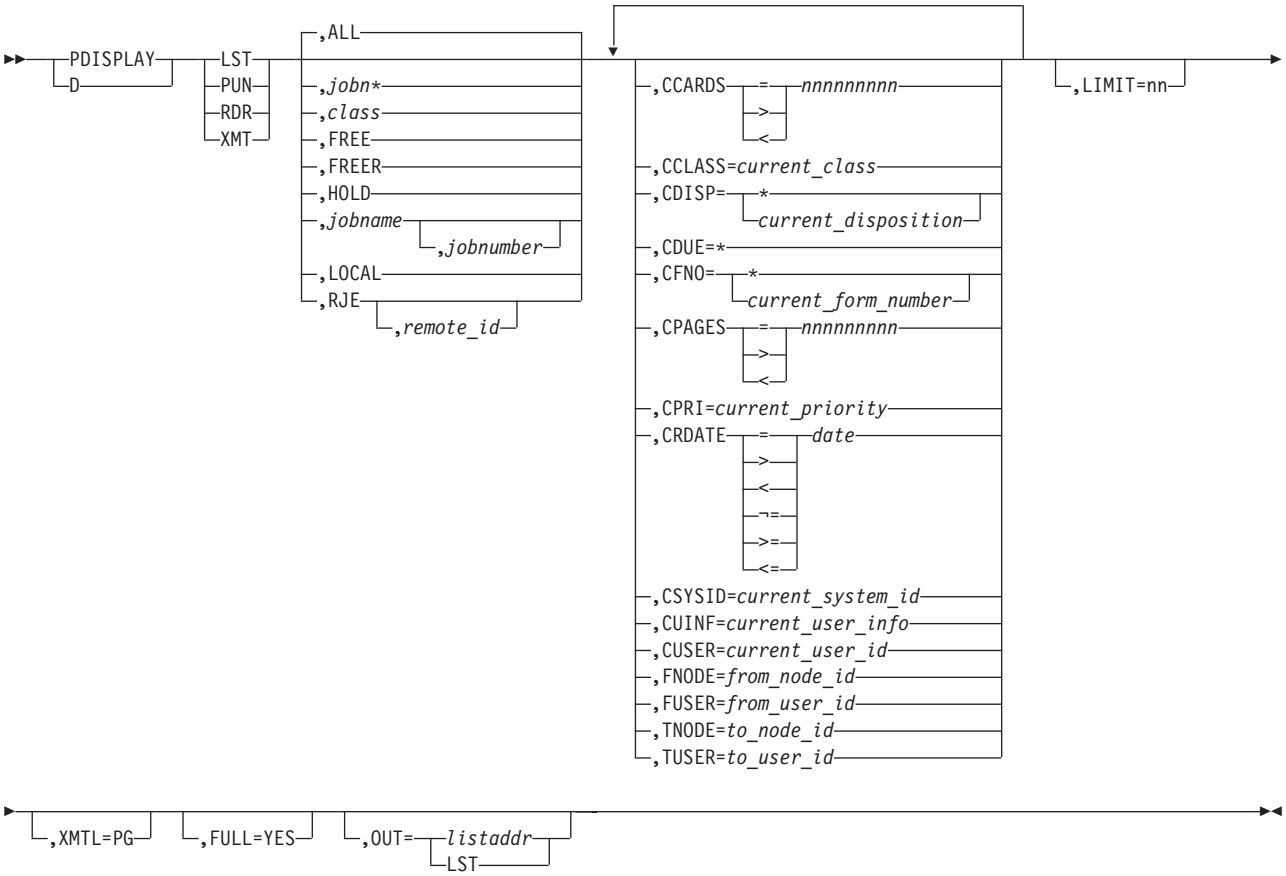
and within such a dispatchable/non-dispatchable class of a queue type according to the selection priority, namely

9, 8,..., 1, 0.

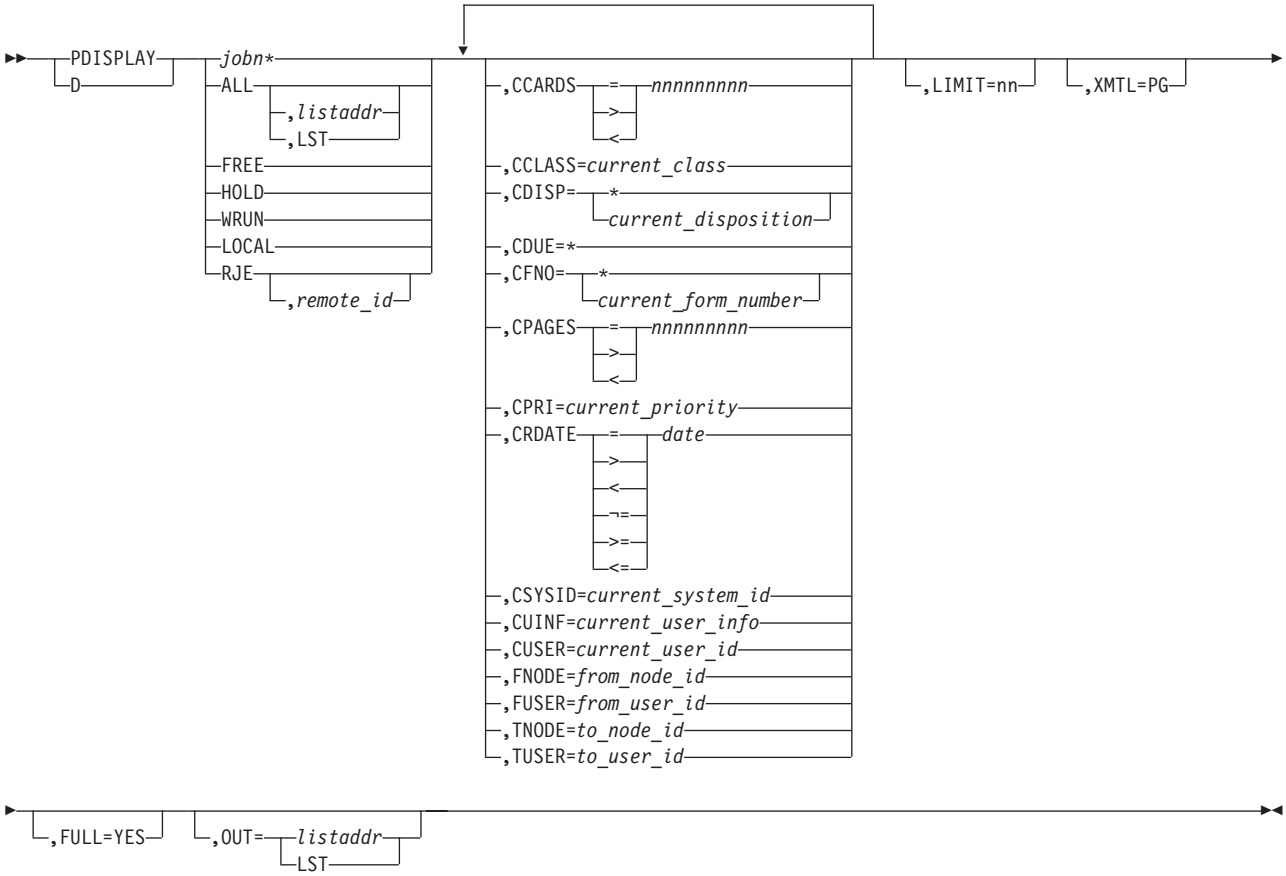
Adhering to this scheme, a new queue entry is always added behind existing ones. There is one exception: within a priority group of the LST/PUN queue, a new entry specifying a formsnumber is added behind existing ones with the same formsnumber, called forms-grouping.

Following the displayed (also internally queued) sequence, queue entries are also selected for operation by their related processing tasks.

Format 1: Displaying Information of a Selected Physical Queue



Format 2: Displaying Information of All Physical Queues



Since format 1 and format 2 are nearly identical, their operands are explained only once.

Specify the queue for which the command is to be executed, in the form:

LST
to address the list queue.

PUN
to address the punch queue.

RDR
to address the reader queue.

XMT
to address 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 (Format 1).

If you do not specify a queue, VSE/POWER selects and displays the status of the affected queue entries in all of its queues. (Format 2)

Positional Search Operands

jobn*
requests a status report on all VSE/POWER jobs whose names begin with the specified characters. For `jobn` you can specify any combination of up to eight alphameric characters.

If the queue operand is not specified, status information for all such jobs in all queues is displayed.

jobname
The name of the VSE/POWER job whose status is to be displayed. The jobname may consist of 2 to 8 alphameric characters. Specifying PDISPLAY with jobname (and optionally causes VSE/POWER to display **all:** occurrences of this name (and number).

jobnumber
The VSE/POWER assigned jobnumber, if already known.

ALL| ALL,listaddr| ALL,LST
Specify ALL **with** a queue specification to get a status report on all VSE/POWER jobs in that queue or on a certain group of jobs in the queue as defined by one or more additional keyword-type search operands (Format 1).

Specify ALL **without** a queue specification to get a status report on all affected jobs in the physical RDR/LST/PUN/XMT queues (Format 2).

ALL is default if you specify at least one keyword-type search operand.

ALL is the default in Format 1.

listaddr
To print a status report. Specify the address of a printer in the form cuu. The printer must be available to the VSE/POWER partition and cannot be used for printing spooled output at the same time.

LST
To spool the queue display as LST queue entry. The LST queue entry has the following attributes:
- it is announced by message 1Q8GI
- $LSTxxxx as jobname, where xxxx are the last four digits of the VSE/POWER-assigned jobnumber.
- Priority 9
- Disposition D (or Disposition H when autostart statement SET HOLDCL=class specified)
- Class A (or any class, as specified by the autostart statement SET HOLDCL=class)

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.

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

*** =** Executing

FREER

Specifying FREER causes VSE/POWER to produce a status report on all jobs of the reader queue (**excluding jobs in the 'wait for run' subqueue**) whose disposition is one of the following:

D = Delete after processing

K = Keep after processing

*** =** Executing

FREER displays all dispatchable 'ready-to-run or running' entries of the reader queue. FREER is rejected by message 1R52I if LST, PUN, or XMT queue is specified.

Jobs positioned in the 'wait for run' subqueue are not displayed. For additional details on the FREER command, see "The PDISPLAY FREE and FREER Commands" on page 171.

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.
```

WRUN

Specifying WRUN causes VSE/POWER to display the 'wait for run' subqueue, that is, all jobs of all classes with disposition D or K whose due date has not yet expired. If this operand is specified, the jobs are displayed according to the next due date, not to class nor priority. For examples refer to "Display of the 'Wait for Run' Subqueue" on page 170.

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|RJE,remote_id

Specifying RJE only causes VSE/POWER to produce a status report on all RJE-type jobs in the specified queue (or in all queues).

remote_id

For *remote_id*, specify a remote ID 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 ID in the form Rnnn or just nnn (where nnn = the ID as defined for VSE/POWER table generation).

Keyword Search Operands

CCARDS=|<|>nnnnnnnnn
indicates that only reader and punch queue entries are to be addressed whose current total card count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified card nnnnnnnnn value, which must be greater than zero. A card count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P). When CCARDS is specified for the LST queue or for list type entries of the XMT queue (I=L), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

   ```
   1R52I LIST QUEUE NOTHING TO DISPLAY, or
   1R52I NOTHING TO DISPLAY
   ```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For reader or punch type entries in the XMT queue the 'LINES' column actually presents statements, that means cards. This is the value against the CCARDS operand is compared.
4. For active entries (DISP=*) in the PUN or XMT queue, 'cards left to be processed' are displayed, although the entries were selected according to their total card count, which is not displayed then.
5. Consider to use the LIMIT= operand at the same time to reduce the amount of displayed output.

CCLASS=current_class
requests a status report on an affected queue entry (or entries) if the class currently assigned to the entry matches the specified class. This operand overwrites a possible specification of the positional 'class' operand. Specification of either 'class' or the 'CCLASS' operand expedites the access path to the VSE/POWER queue file.

CDISP=*|current_disposition
requests a status report on an affected queue entry (or entries) 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 processed. This operand overwrites a possible specification of the positional 'FREE' or 'HOLD' operand.

CDUE=*
indicates that all jobs are to be displayed for which time event scheduling operands have been specified. These jobs can be either in the RDR or the XMT queue.

CFNO=*|current_form_number
requests a status report on an affected queue entry (or entries) if the form number defined for the output of the entry matches the specified form number. One can specify '*' to process those queue entries which have no form number, or the four-character form number of the queue entries to be processed.

CPAGES=|<|>nnnnnnnnn
indicates that only list queue entries are to be addressed whose current total page count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified page nnnnnnnnn value, which must be greater than zero. A page count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the LST queue or for list type entries of the XMT queue (I=L). When CPAGES is specified for the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

   ```
   1R52I READER QUEUE NOTHING TO DISPLAY, or
   1R52I NOTHING TO DISPLAY
   ```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For list type entries in the XMT queue the standard display presents 'LINES'. Use the 'P=n...n' page count of the FULL=YES display, when you select entries by the CPAGES operand.
4. For active entries (DISP=*) in the LST or XMT queue, 'pages left to be processed' are displayed, although the entries were selected according to their total page count, which is not displayed then.
5. Consider to use the LIMIT= operand at the same time to reduce the amount of displayed output.

CPRI=current_priority
requests a status report on an affected queue entry (or entries) if the priority currently assigned to the entry matches the specified priority.

CRDATE=|>|<|¬=|>=|<=date
This string requests a status report for those queue entries whose creation date is one of the following in relation to the specified date:

```
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

```
1) mm/dd/yy or dd/mm/yy
2) mm/dd/yyyy or dd/mm/yyyy
```

Do not specify mm greater than 12, dd greater than 31, and yy or yyyy greater than the current year.

Notes:

1. If you specify a 2 digit yy-year, VSE/POWER expands it for internal processing and comparison to a 4 digit year according to the 'fix-88-window' rule:

```
if yy is greater than (>) 88, then 19yy
if yy is less than or equal (<=) 88, then 20yy
```

2. The format of the system date must not be changed after VSE/POWER has been initialized or between warm starts of VSE/POWER (see "Functions Not Expecting Date Format Changes" on page 128).
3. In a shared spooling environment, all systems must have the same date format.

CSYSID=current_system_id
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.

CUINF=current_user_info
indicates that the requested display is to be done if the queue entry's user information matches the specified current_user_info value. User information is provided to a queue entry by the * $$ JOB/LST/PUN JECL statements or by the PALTER command, and can be made visible by a PDISPLAY ...,FULL=YES request, which presents this information left-justified with trailing blanks, all enclosed in quotes via the U='...' display field (see also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257). When user information contains

1. neither embedded blanks nor commas and is displayed, for example, as

   U='PETER ', it can be searched for using
   CUINF=PETER, or
   CUINF='PETER' or
   CUINF='PETER '
2. an embedded blank (or comma) and is displayed, for example, as

   U='PETER BOY ', it can be searched for using
   CUINF='PETER BOY ', or
   CUINF='PETER BOY'
3. an embedded blank (or comma) and a single quote, shown for example as

   U='PETER'S BOOK ', it can be searched for using
   CUINF='PETER''S BOOK'

The CUINF specification (when enclosed in single quotes) can contain trailing blanks up to the field limit of 16 characters.

CUSER=current_user_id
requests a status report on an affected queue entry if the entry's "from user" or "to user" ID matches the ID given in this operand.

FNODE=from_node_id
requests a status report on an affected queue entry if the entry's "from node" name matches the node name specified by *from_node_id*.

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 partial list of information that VSE/POWER displays in addition if you specify FULL=YES:

- Time event scheduling information (for RDR queue entries only)
- Distribution code (for output queue entries only)
- Information for 3800 printer
- Creation date
- User information

- Job-suffix (output-segment) number
- Line count (for LST queue entries)
- Page count (for LST entries in the XMT queue)
- Original disposition if a job has a temporary disposition
- Number of allocated DBLK groups for the queue entry (0 for tape display)
- Output format, that is, one of the following:

```
  ASA = Records with American National Standard control
        characters
  BMS = Basic mapping support (used by CICS)
 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)
```

- VSE/POWER internal queue record number (QNUM)
- NORUN=ignore indication for jobs

FUSER=from_user_id
requests a status report on an affected queue entry if the entry's "from user" ID matches the ID specified by *from_user_id*.

Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user ID of LOCAL. It does not result in a status report on entries originated by local users.

LIMIT=nn
Requests to limit the output of the current display command to the first *nn* entries. The limit specification may range from 1-99. If less than or equal *nn* selectable entries are found, the limit operand has no visible effect. If more than *nn* are found, the display is terminated after *nn* entries without an additional warning. This operand is recommended in combination with either the CPAGES or the CCARDS operand.

OUT=LST | listaddr
Specify where the information about the queue entries that should be displayed. Specifying OUT= forces the FULL=YES display function and format.

LST
Specifies that a LST queue entry with the following characteristics is to be built:
- It is announced by message 1Q8GI
- $LSTnnnn as job name, where nnnn are the last four digits of the VSE/POWER-assigned job number
- Priority 9
- Disposition D (or Disposition H when autostart statement SET HOLDCL=class specified)
- Class A (or any class, as specified by the autostart statement SET HOLDCL=class)

listaddr
Indicates the information is to be printed on the specified printer. Specify the address as cuu.

TNODE=to_node_id
requests VSE/POWER to display the status of an affected queue entry if the entry's "to node" name matches the node name specified by *to_node_id*. This operand is valid only if your specification for queue is XMT.

TUSER=to_user_id
requests a status report on an affected queue entry if the entry's "to user" ID matches the ID specified by *to_user_id*.

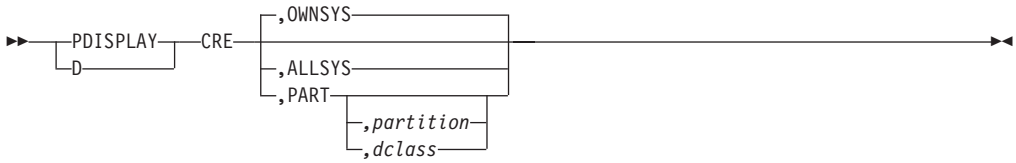
The TUSER specification cannot be used for displaying the status of queue entries residing in the reader queue.

Specifying TUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user ID of LOCAL. It does not result in a status report on entries to be handled at and destined for the central location.

XMTL=PG
Use this operand if VSE/POWER is to display spooled 'pages' for LST-type (I=L) entires in the XMT queue instead of spooled lines.

Note: For such entries, when being transmitted (DISP=*), the displayed page count is the 'total' spooled and not the 'remaining to be transmitted' count as with displayed lines. Remaining line counts are presented by PDISPLAY A,PNET or by PDISPLAY without XMTL=PG.

Format 3: Displaying Information of the Create Queue



Use this format if you want to know which queue entries are currently being created (see "Life Cycle of VSE/POWER Queue Entries" on page 30 and what their spooling progress is. Concerning the currency of the spooling state, consider the following:

Whenever an entry in creation is found, the task processing the 'PDISPLAY CRE' command refreshes the information found in the queue record with the latest level found in the queue record copy attached to the creating task. For a shared spooling complex, this can only be done for queue entries in creation on the system that processes the command. Therefore, information about queue entries of other systems may not be as current as the information about queue entries of the local system. Requesting information about queue entries on all systems must be done explicitly. By default, only the information for the local system is shown.

CRE
This operand limits the display to queue entries in creation.

OWNSYS
This operand limits the display to all queue entries in creation on the system where the command is processed.

ALLSYS
This operand enables the command to show all queue entries in creation on all systems of a shared spooling complex. Information about entries in creation on systems other than the one where the command is processed may not show the actual line counts.

PART
This operand limits the display to queue entries in creation by execution writers of static or dynamic partitions of the local system.

partition
The static or dynamic partition ID.

dclass
The operand limits the display to queue entries in creation by all active execution writers started for dynamic class *dclass*.

Example: For a display of in-creation queue entries, see "Format 3: PDISPLAY CRE" on page 263.

Note: When the queue file comprises more than about 20,000 entries, the PDISPLAY command may not complete immediately. For an explanation see "Large Queue File Considerations" on page 35

Format 4: Displaying Information of the Deletion Queue

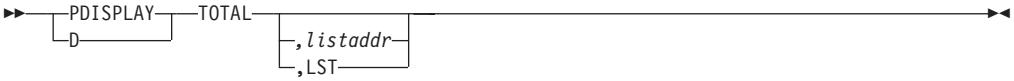
DEL
This operand limits the display to queue entries in *delayed deletion*. For details on queue records pending in this state, see

- "Life Cycle of VSE/POWER Queue Entries" on page 30
- "Delaying the Deletion of Browsed Queue Entries" on page 122

Example: For a display of queue entries in deletion, see "Format 4: PDISPLAY DEL" on page 265.

Note: When the queue file comprises more than about 20,000 entries, the PDISPLAY command may not complete immediately. For an explanation see "Large Queue File Considerations" on page 35

Format 5: Displaying Information for All Physical and Logical Queues



With the introduction of the Creation and Deletion queues in VSE/POWER 6.5, the existing PDISPLAY ALL command reflects only queue entries of the "physical" RDR/LST/PUN/XMT queues but not entries of the "logical" CRE and DEL queues. An additional format of the PDISPLAY command has therefore been implemented to show absolutely all queue entries:

TOTAL
Specify TOTAL to get a status report on all queue entries of the "physical" RDR/LST/PUN/XMT queues and the "logical" DEL and CRE queues, in the cited order. For a Spool-Access fixed-format control request, entries of the Deletion queue are identified by flag PXFM3DEL, entries of the Create queue by PXFM3CRE. For details, refer to the PWRSPL macro, OPT=FORMAT, in *VSE/POWER Application Programming*.

Note: In a shared environment, the total includes all queue entries, including those in-creation on individual shared systems, that is, as if the PDISPLAY CRE,ALLSYS command had also been entered.

listaddr
To print a status report, specify the address of a printer in the form cuu. The printer must be available to the VSE/POWER partition and cannot be used for printing spooled output at the same time.

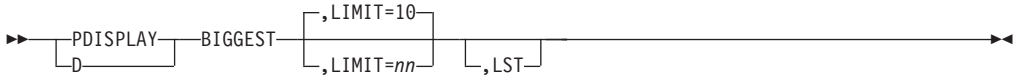
LST
To spool the queue display as a LST queue entry. The LST queue entry has the following attributes:

- It is announced by message 1Q8GI
- $LSTxxxx as jobname, where xxxx represents the last four digits of the VSE/POWER assigned job number.
- Priority 9
- Disposition D (or Disposition H when autostart statement SET HOLDCL=class specified)
- Class A (or any class, as specified by the autostart statement SET HOLDCL=class)

Specifying TOTAL,listaddr or TOTAL,LST forces the FULL=YES display option for the ″physical″ RDR/LST/PUN/XMT queues.

Note: When the queue file comprises more than about 20,000 entries, the PDISPLAY command may not complete immediately. For an explanation see “Large Queue File Considerations” on page 35

Format 6: Displaying Information for the ’nn’ Largest Entries



Whenever the VSE/POWER data file becomes full, and before one can extend it during a warm start by an additional disk extent, one may have to concern oneself with the spool entries consuming the greatest spooling space according to the queue in which they reside.

Use this display format to identify the ’nn’ largest queue entries (sorted by DBLKGPs used) within the total VSE/POWER spool space.

BIGGEST
Specify BIGGEST to get a status report of entries in the ’physical’ RDR/LST/PUN/XMT queues and in the ’logical’ CRE and DEL queues. The entries may be presented intermixed by queues, but they are sorted in descending sequence by the data block groups used (DBLKGP), meaning the spool space consumed.

LIMIT=10 | nn
Specify this operand and request to limit the output of the BIGGEST display to the first *nn* entries. The limit specification may range from 1-99. The specified value is reflected in the display header line by *nn* BIGGEST SORTED’. If less than or equal *nn* queue entries are currently housed by VSE/POWER, the limit operand has no visible effect. If more than *nn* are found, the display presents the *nn* largest spool entries.

LST
To spool the BIGGEST display as a list queue entry with the following attributes:

- It is indicated by message 1Q9GI
- ’$BIGxxxx’ as the jobname, where ’xxxx’ represents the last four digits of the VSE/POWER assigned jobnumber
- Priority 9
- Disposition H
- Class A

Note:

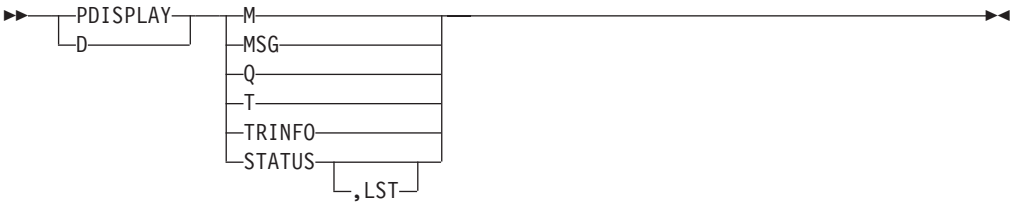
1. The display BIGGEST command needs a sorting area of *nn* times 384 Bytes. If not available in VSE/POWER partition GETVIS, the command is terminated by message 1Q7AI for a later retry or for a retry with a lower LIMIT value
2. When the queue file comprises more than about 20,000 entries, the PDISPLAY command may not complete immediately. For an explanation see “Large Queue File Considerations” on page 35

3. The display BIGGEST queue command may also be requested by a programmed Spool-Access CTL access for message buffers, however **not** in fixed format.
4. The display BIGGEST command scans the total VSE/POWER queue file and at the same time flags those queue entries by message 1Q2MI, whose queue identification field has been destroyed, that is, *this command may be used to verify the validity of a given queue file.*

.

Example: For a display of the largest queue entries, see "Format 6: PDISPLAY BIGGEST,LIMIT=7" on page 267

Format 7: Displaying Various Status Information



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 every message to delete a specific message.

Q To cause a display of:
- The percentage of the queue file current in use, the number of free queue records, the distribution of the used queue entries among the logical (CRE,DEL) and physical (RDR,LST,PUN,XMT) queues and disk extent information of the queue file.
- The percentage of the data file currently in use, the number of free DBLK groups, the DBLK and DBLK group size, and information about the disk extent(s) of the data file. The last extent may show '(IN FORMAT)' when data file extension is in progress.
- The percentage of the account file (if accounting is supported) currently in use, and the disk extent information for the account file.

Note: When the queue file comprises more than about 20,000 entries, the PDISPLAY command may not complete immediately. For an explanation see "Large Queue File Considerations" on page 35

Example: For a display of spool file characteristics, see "Format 7: PDISPLAY Q" on page 268.

T To cause a display of:
- The current time and date with four-digit year in the current date format (see "Functions Allowing Date Format Changes" on page 127).
- 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 POWER macro generation.
- The node ID of the local node if networking is used
- The SECNODE value of the local system if the z/VSE Access Control function is activated.
- The SECNODE value of other shared systems if any.

TRINFO
To cause a summary display of the following two VSE/POWER internal trace areas:

- Telecommunication Trace Area, displayed as TELETR. For detailed information on this trace, refer to "PNET Telecommunication Trace" in *VSE/POWER Networking* and "RJE,BSC Telecommunication Trace" in *VSE/POWER Remote Job Entry*.
- Task Trace Area, displayed as TASKTR. For details on this trace, see "VSE/POWER Task Dispatching Trace" on page 516.

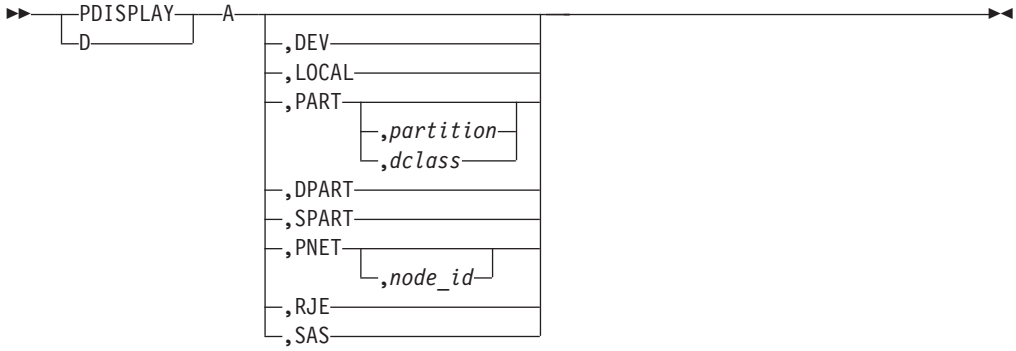
STATUS
To cause a display of the statistics status report about the current VSE/POWER session at the console. For an example, see Figure 7 on page 46.

LST
To create the statistics report in a list queue entry with the following attributes:

- indicated by message 1Q8BI
- $STAxxxx as jobname; xxxx are the last four digits of the VSE/POWER-assigned jobnumber
- Priority 9
- Disposition H
- Class A

The message 1Q8BI indicates the completion of the command and identifies the unique name of the list queue entry.

Format 8: Displaying Activated Task Status



A Code this operand as shown.

Use this operand alone to get a list of all important external tasks which have been activated. You can restrict this type of display by specifying one of the optional operands described below. The entire group of operands defines what in VSE/POWER is called an 'external task'.

The display active presents tasks from an external view, as an operator may be interested in, and shows:

- Queue entries being processed (DISP=*) by tasks

- Queue entries being created
- The spooling progress for the queue entry, for example 'left to be printed'

For invisible VSE/POWER internal tasks, use the PDISPLAY TASKS command as described on page 246.

DEV
The operand limits the display to activated device-service tasks.

LOCAL
The operand limits the display to activated reader and writer tasks.

PART
The operand limits the display to the activated tasks associated with static or active dynamic partitions.

You may further limit the display by specifying one of the subsequent optional operands:

partition
The static or dynamic partition id.

dclass
The operand limits the display to the activated tasks associated with all active dynamic partitions started for dynamic class *dclass*.

DPART
The operand limits the display to all activated tasks associated with all active dynamic partitions.

SPART
The operand should be specified if you want to display all activated tasks associated with active static partitions.

PNET
The operand limits the display to all activated network tasks if you specify PNET alone.

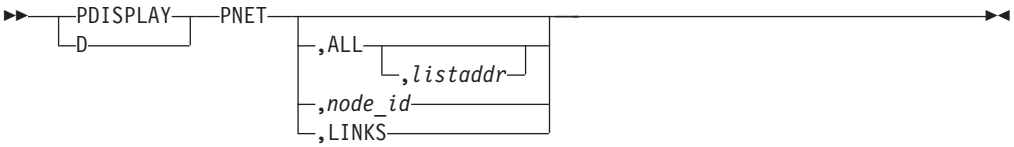
node_id
Limits the display to networking tasks activated for a certain node. For *node_id*, specify the name of that node.

RJE
The operand limits the display to all activated RJE tasks.

SAS
The operand limits the display to all activated spool-access support tasks.

Example: For a display of an activated tasks example, see "Format 8: PDISPLAY A" on page 270.

Format 9: Displaying Network Definition Table Status (NDT)



PNET
Code this operand as shown.

If you specify PNET alone, VSE/POWER displays networking information about your own node, the VSE system you are using. You may in addition, specify one of the (optional) operands described below.

ALL
Specify ALL to have VSE/POWER display networking information about all nodes contained in the active Network Definition Table (NDT).

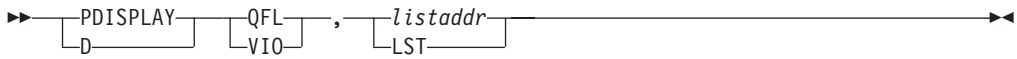
listaddr
Specify, for *listaddr*, the address of a printer if you want a printed status report. VSE/POWER then uses the printer for this purpose. Specify the address in the cuu. The printer must be available to the VSE/POWER partition and cannot be used for printing spooled output at the same time.

node_id
Specify, for *node_id*, 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.

Example: For a display of the currently loaded network definition table, see "Format 9: PDISPLAY PNET" on page 275.

Format 10: Listing the Storage Copy of the Queue File



Use this command to get either a printout or to produce a LST queue entry of the in-storage copy of the queue file, and the Master Record in hex-translated dump format.

QFL|VIO
Specify one of the operands as shown. Both address the queue file independent of where it actually resides in the running VSE/POWER system (partition GETVIS area or VIO).

listaddr
Specify the cuu address of the printer to be used.

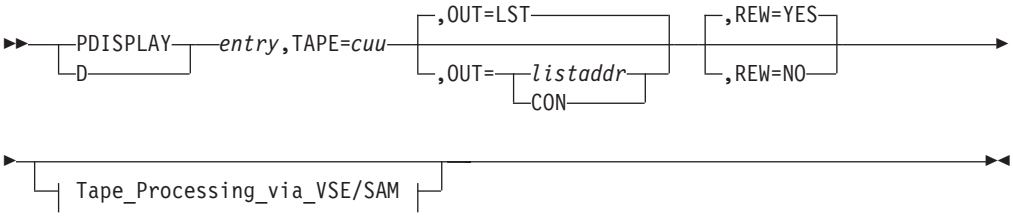
In a shared spooling environment, use the `QFL|VIO,listaddr` command with care, since another sharing system that needs queue file write access has to wait until processing of the command is completed.

LST
Specify this operand to create a LST queue entry with:
- indicated by message 1Q2GI
- $VIOxxxx as jobname, where xxxx are the last four digits of the VSE/POWER-assigned jobnumber if the queue file resides in the VIO area.
- $QFLxxxx as jobname, where xxxx are the last four digits of the VSE/POWER-assigned jobnumber if the queue file resides in the partition GETVIS area.
- Priority 9
- Disposition H
- Class A

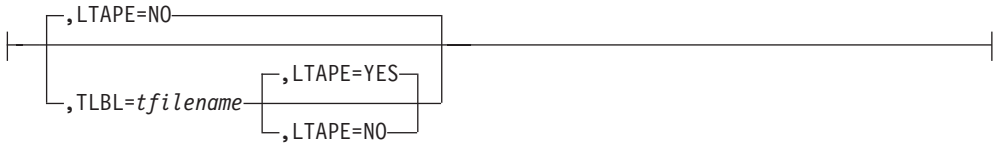
Message 1Q2GI signals completion of the PDISPLAY request and identifies the jobname and jobnumber of the created list queue entry. The layout of the printed queue file is similar to the dump format. Every queue record is identified by its relative queue record number, every queue record block by its relative queue record block number.

Format 11: Displaying Queue Entries Residing on Tape

This format displays queue entries residing on either a POFFLOAD tape or a spool tape. For further information on this command, please refer to "Tape Processing with VSE/POWER" on page 151.



Tape_Processing_via_VSE/SAM:



entry
To define which queue entry or entries you want to have displayed, specify any of the operands listed for Format 1 or 2 of the PDISPLAY command on page 226.

TAPE=cuu
Specify the cuu address of the tape to be used.

OUT=LST|listaddr|CON
Specify where the information about the queue entries on the tape should be displayed.

LST

Forces the FULL=YES display function. Specifies that a LST queue entry with the following characteristics is to be built:
- It is announced by message 1Q8GI
- $TAPnnnn as job name, where nnnn are the last four digits of the VSE/POWER-assigned job number
- Priority 9
- Disposition D (or Disposition H when autostart statement SET HOLDCL=class specified)
- Class A (or any class, as specified by the autostart statement SET HOLDCL=class)

Due to the FULL=YES function, the LST queue entry contains the following information for every queue entry on the tape:
- Job name, number, and suffix, if applicable
- Class

- Queue ID (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 cuu. Forces the FULL=YES display function.

CON
Specifies that the information about the queue entries is to be displayed at the 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. For positioning on the tape at PDISPLAY termination and for hand-positioning on the tape, see the NOREW operand of the POFFLOAD command 334.

TLBL=tfilename
For tfilename specify one to seven characters of a tape label `// TLBL filename...`. It is assumed that the `//` TLBL statement is available to the VSE/POWER partition. For the syntax of the TLBL statement, see *z/VSE System Control Statements*.

If specified and the LTAPE= operand is not specified, then LTAPE=YES is assumed.

For more information about using the TLBL statement see the description of LTAPE.

LTAPE=YES|NO
If specified, VSE/SAM is used during OPEN, CLOSE and end-of-volume (EOV) tape processing.

The REW=NO operand is not allowed for LTAPE=YES.

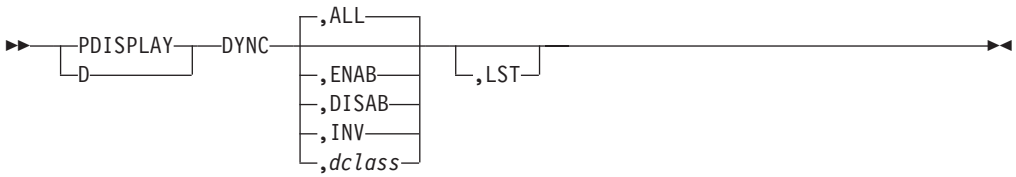
YES
Indicates VSE/SAM labeled tape processing.

NO
Indicates VSE/SAM unlabeled tape processing.

If TLBL= is specified in addition to LTAPE=NO, an unlabeled tape will still be processed using VSE/SAM. However, VSE/POWER will pass the tfilename value to VSE/SAM as for labeled tape processing. This operand value is normally needed only by Tape Management Systems which might use the // TLBL statement to indicate tape information, such as tape density. The TLBL= value then applies to all unlabeled tapes produced.

Example: For a so-called "tape display", see "Format 11: PDISPLAY queue,TAPE=cuu,OUT=CON" on page 278.

Format 12: Displaying the Active Dynamic Class Table



DYNC
Specify this operand to display characteristics of classes contained in the currently active Dynamic Class Table.

ALL
Specify ALL to display all classes of the currently active Dynamic Class Table. ALL is the default.

ENAB
Specify ENAB to display the characteristics of enabled classes only. The following classes are 'enabled' by definition:

- All enabled active classes (state: ENAB)
- All enabled suspended classes (state: SUSPEND)

DISAB
Specify DISAB to display the characteristics of only disabled classes. The following classes are 'disabled' by definition:

- All invalid classes (state: *INV* or INV-SP)
- All disabled valid classes (state: DISAB)

INV
Specify INV to display the characteristics of only those classes flagged 'invalid'.

dclass
Specify one of the alphabetic classes as defined in the currently active Dynamic Class Table. The specified class is considered for a display of its characteristics.

LST
Specify LST if the class characteristics should not be displayed on the console, but should be collected in a list queue entry with the following attributes:

- $DYDxxxx as job name, where xxxx are the last four digits of the VSE/POWER-assigned job number
- Priority 9
- Disposition H
- Class A

  The completion of PDISPLAY DYNC,,LST is signalled to the command originator by message 1Q6BI. It identifies the job name and number of the created list queue entry.

Example: For display samples and further information refer to "Display Features of Dynamic Classes" on page 137 and "Format 12: PDISPLAY DYNC,ALL" on page 280.

Format 13: Displaying Exit Information

Exit
Specify this operand to get an overview of the exit routines actually loaded (JOBEXIT, OUTEXIT, NETEXIT, and XMTEXIT).

For each loaded exit, VSE/POWER issues a message line containing information about type of exit, state (enabled or disabled), name, size of work area (decimal), address of the exit (hexadecimal) and its size (decimal), and type of workunit for which the exit should be given control.

Example: For a display exit example, see"Format 13: PDISPLAY EXIT" on page 282.

Format 14: Displaying Information About Used Autostart Statements

AUSTMT
Specify this operand to display all autostart statements which have been used when VSE/POWER has been started.

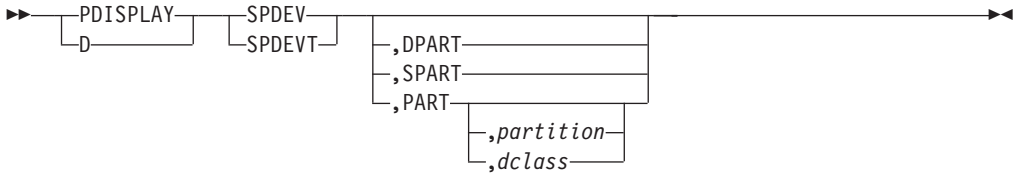
If no autostart statements have been provided at startup, message

```
1R48I NO AUSTART STATEMENTS PROCESSED
```

is returned.

Example: For a display AUSTMT example and return information, seepage 283.

Format 15: Displaying Information About Spooled Devices



Use this format if you want to know which reader, printer, or punch device should be intercepted for spooling by VSE/POWER when corresponding I/O requests are issued from currently active spooled static or dynamic partitions. Such devices are called 'spooled devices'.

SPDEV
Specify this operand to display the 'spooled devices' of **all** currently active spooled partitions. In a static VSE/POWER controlled partition, I/O devices become spooled devices if they have been

- specified by means of the autostart statements READER=, PRINTERS=, or PUNCHES=
- or been replied to the message 1R50D.

In a dynamic partition I/O devices become a 'spooled device', if they have been specified as such in the class entries of the dynamic class table.

SPDEVT
Specify this operand to display the *spooled devices* of all spooled partitions together with the PUB-device type code and the device type code used within the ADD statement during IPL time. For an explanation of 'spooled', refer to the SPDEV operand.

DPART
The operand limits the display to spooled devices of all active dynamic partitions.

SPART
The operand limits the display to spooled devices of all static partitions.

PART
The operand presents a display of spooled devices of all static and active dynamic partitions.

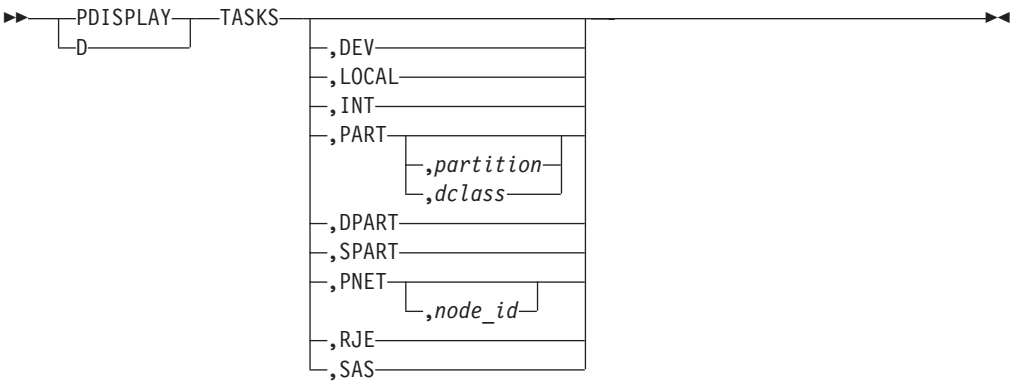
You may further limit the display by specifying one of the subsequent optional operands:

partition
The static or dynamic partition ID.

dclass
This operand limits the display to all active dynamic partitions started for dynamic class *dclass*.

Example: For a display of information on spooled devices using PDISPLAY SPDEV/SPDEVT, see page 284.

Format 16: Displaying Internal Information About All Tasks



TASKS
Code this operand as shown.

Use this operand to get a list of **all internal** and **external** VSE/POWER tasks which are active. For a definition of external tasks, refer to the PDISPLAY ACTIVE command on page 239.

You may be asked by your IBM representative to use this command. Its display shows all active tasks in an internal view which may be helpful to analyze VSE/POWER problems.

You can restrict the type of display by specifying only one of the optional operands below. In that case, not all internal tasks are displayed which might match the selection criterion. For example, if 'PNET' is selected, the LLDR task will not be displayed, although this task is the PNET line driver task.

DEV
The operand limits the display to active device-service tasks.

LOCAL
The operand limits the display to active reader and writer tasks.

INT
The operand limits the display of active VSE/POWER *internal* tasks, such as the command processors or the dynamic partition scheduling task.

PART
The operand limits the display to the active tasks associated with static or active dynamic partitions.

You may further limit the display by specifying one of the subsequent optional operands:

partition
The static or dynamic partition id.

dclass
The operand limits the display to the active tasks associated with all active dynamic partitions started for dynamic class *dclass*.

DPART
The operand limits the display to all active tasks associated with all active dynamic partitions.

SPART
The operand should be specified if you want to display all active tasks associated with active static partitions.

PNET
The operand limits the display to all active networking tasks if you specify PNET alone.

node_id
Limits the display to networking tasks active for a certain node. For *node_id*, specify 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.

Example: For a display of an active tasks example, see "Format 16-1: PDISPLAY TASKS" on page 286.

Examples of the PDISPLAY Command

PDISPLAY RDR,ASSM* or PDISPLAY RDR,*ASSM
Displays the status of all VSE/POWER jobs in the reader queue whose names begin with ASSM.

PDISPLAY RDR,ASSEM,CRDATE<=05/15/2005
Displays the status of that VSE/POWER job ASSEM in the reader queue whose creation date is earlier than or equal to May 15, 2005.

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 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 ID) 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.

D DYNC,ENAB
Displays all classes of the active Dynamic Class Table which are in state ENAB or SUSPEND.

PDISPLAY Output Examples

VSE/POWER produces status displays in 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 (for example PDISPLAY RDR,ALL) may cause many status lines to be written to the console in rapid succession. You may have problems reading them all and finding 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 (for example, PDISPLAY RDR,C,CPRI=3).

In the status display examples that follow, the reference numbers in front of the display lines refer to the 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.

For status display examples of jobs with time event scheduling operands see also “Display of the Next Due Date” on page 166.

For status display examples of the verified or active Dynamic Class Table, see also “Dynamic Partition Support - Operation” on page 135.

The following status display examples are discussed:
- Format 1-1: PDISPLAY RDR
- Format 1-2: PDISPLAY RDR,FULL=YES
- Format 1-3: PDISPLAY LST
- Format 1-4: PDISPLAY LST,FULL=YES
- Format 1-5: PDISPLAY XMT
- Format 3: PDISPLAY CRE
- Format 4: PDISPLAY DEL
- Format 6: PDISPLAY BIGGEST
- Format 7: PDISPLAY Q
- Format 8: PDISPLAY A
- Format 9: PDISPLAY PNET
- Format 11: PDISPLAY queue,TAPE=cuu,OUT=CON
- Format 12: PDISPLAY DYNC,ALL
- Format 13: PDISPLAY EXIT
- Format 14: PDISPLAY AUSTMT
- Format 15-1: PDISPLAY SPDEV
- Format 15-2: PDISPLAY SPDEVT
- Format 16-1: PDISPLAY TASKS
- Format 16-2: MSG F1,DATA=D TASKS

Format 1-1: PDISPLAY RDR

```
1  1R46I  READER QUEUE   P D C S  CARDS B
2  1R46I  VTAMSTRT 00163 3 * 3       20   PART=F3 FROM=(SVSA)
3  1R46I  HUS04    00004 7 K A        3   EXP=16:30,06/15 FROM=(SCHMIDTH)
4  1R46I  HUS01    00001 7 D A        4   RUN=12:30,06/05
   1R46I  WST07    00027 5 D A       10
          SECN=ZONE04
   1R46I  HUS09    00009 1 L A        3   RUN=-- FROM=BOEBLVM1(SCHMIDTH)
    ...  ...  ...
```

Figure 23. Display of the Reader Queue

```
1  1R46I  READER QUEUE   P D C S  CARDS B
```

The first line displays the header line and presents the display column ns:

1R46I The message number.

READER QUEUE
Identifies the queue selected.

P The priority column.

D The disposition column, presenting the local (processing) disposition valid for the RDR queue. For information on dispositions, see Appendix A, "VSE/POWER Disposition Codes," on page 497.

C The class column.

S The system-ID column which applies only if shared spooling is used. VSE/POWER displays the system on which the queue entry is to be or is processing. Instead of system-ID (1-9), VSE/POWER displays 'M' (multiple access) if the queue entry is accessed by the GET Service for browse on more than one sharing system.

CARDS
The number of statements of the reader job. Up to 7 digits may be displayed. When the number exceeds 9999999 then only 9999999 is displayed.

B This column may indicate with an asterisk ('*') that the queue entry is "being browsed". The MACC count of a FULL=YES display will identify the number of browsing tasks. See also "Browsing of Queue Entries" on page 121.

```
2  1R46I  VTAMSTRT 00163 3 * 3       20   PART=F3 FROM=(SVSA)
```

A reader queue entry being processed (D column = '*') by a partition that serves execution class 3. This partition is further identified by PART=F3. For sharing systems, PART= is shown only when the job is executed and the display command was entered on the *same* sharing VSE/POWER system.

```
3  1R46I  HUS04    00004 7 K A        3   EXP=16:30,06/15 FROM=(SCHMIDTH)
```

The first and only display line of a dispatchable reader queue entry for which the specified due date has expired - namely at 16:30 on the 15th of June - provided the system date is in the format month/day/year. Since expiration, no partition serving class A has been available to process the job.

For additional examples with time event scheduling information displayed, see "Display of the Next Due Date" on page 166.

```
4   1R46I  HUS01    00001 7 D A        4   RUN=12:30,06/05
```

The first and only display line of a dispatchable reader queue entry for which the specified due date has not yet expired. The job is planned to be scheduled at 12:30 on the 5th of June, provided the system date is in the format month/day/year. The year is not displayed and may be the

- Current or any coming year, in case the job has to be scheduled only once. The FULL=YES display provides the year then.
- Current or next year, in case the job has to be scheduled repetitively. Current or next year may be determined by comparing the displayed month and day with the current date.

```
     1R46I  READER QUEUE   P D C S  CARDS B
     1R46I  VTAMSTRT 00163 3 * 3       20   PART=F3 FROM=(SVSA)
     1R46I  HUS04    00004 7 K A        3   EXP=16:30,06/15 FROM=(SCHMIDTH)
     1R46I  HUS01    00001 7 D A        4   RUN=12:30,06/05
5    1R46I  WST07    00027 5 D A       10
            SECN=ZONE04
6    1R46I  HUS09    00009 1 L A        3   RUN=-- FROM=BOEBLVM1(SCHMIDTH)
      ...  ...  ...
```

Figure 24. Display of the Reader Queue (continued)

```
5   1R46I  WST07    00027 5 D A       10
           SECN=ZONE04
```

A dispatchable reader queue entry that is ″trusted″ or ″authenticated″ for the security zone ″ZONE04″. The continuation line indicates that the entry has inherited security authorization for a security SECNODE different than for the local system. For display information on the local SECNODE value, see the PDISPLAY T command on page 238.

```
6   1R46I  HUS09    00009 1 L A        3   RUN=-- FROM=BOEBLVM1(SCHMIDTH)
```

Only display line of a non-dispatchable reader queue entry. The 'RUN=--' indicates that time event scheduling operands have been specified. Use the FULL=YES display to obtain these specifications.

The RDR queue display may also be requested by a programmed Spool-Access CTL command for messages in fixed format (either by D RDR, D ALL, or D TOTAL).

Format 1-2: PDISPLAY RDR,FULL=YES

```
      1R46I  READER QUEUE   P D C S  CARDS B
      1R46I  HUS05    00005 7 K A        3   RUN=17:30,12/15
             D=12/08/2005 DBGP=000001
1           DUETIME=17:30 DUEMONTH=(1-12) DUEDAY=(1,15) RERUN=YES
2           UNSEC QNUM=00999 NORUN=IGN
      1R46I  WST07    00027 5 D A       10
             D=12/09/2005 DBGP=000001
3           SECN=ZONE04 QNUM=01234
       ...  ...  ...
```

Figure 25. Full Display of the Reader Queue

1 `DUETIME=17:30 DUEMONTH=(1-12) DUEDAY=(1,15) RERUN=YES`

This text indicates time event scheduling information. For detailed information about full display of time event scheduling operands and for additional examples, refer to “Display of the Time Event Scheduling Operands” on page 167.

2 `UNSEC QNUM=00999 NORUN=IGN`

This line indicates that VSE/POWER Spool Access Protection is active and that the given reader queue entry is specified not to be spool access protected against access or manipulation (including its output by default). Furthermore, NORUN=IGN indicates that this operand has been specified in the * $$ JOB statement to prevent DISP=X from being assigned during queue file recovery due to SET NORUN=YES.

3 `SECN=ZONE04 QNUM=01234`

This line is the third line for the second job in the queue. This line identifies a reader queue entry that is ″authenticated″ for the security zone ″ZONE04″ (SECNODE). A ’PDISPLAY FULL=YES’ command displays all ″authenticated″ reader queue entries with their SECNODE values.

QNUM shows the VSE/POWER internal queue record number in a 5-digit decimal format for use in Spool Access Support Direct CTL or GET requests as described in *VSE/POWER Application Programming*. You can also specify this number for the ″CQNUM=″ selection operand of the PALTER/PDELETE/PHOLD/PRELEASE commands if output entries cannot be distinguished by jobname and jobnumber or other attributes.

Format 1-3: PDISPLAY LST

```
1  1R46I    LIST QUEUE   P D C S  PAGES   CC FORM B
   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)
     ...  ...  ...
```

Figure 26. Display of the List Queue

```
1  1R46I    LIST QUEUE   P D C S  PAGES  CC FORM B
```

The first line displays the header line and presents the display columns:

1R46I The message number.

LIST QUEUE
Identifies the queue selected.

P The priority column.

D The disposition column, presenting the local (processing) disposition valid for the LST queue. For information on dispositions, see Appendix A, "VSE/POWER Disposition Codes," on page 497.

C The class column.

S The system-ID column which applies only if shared spooling is used. VSE/POWER displays the system on which the queue entry is to be or is being processed. Instead of system-ID (1-9), VSE/POWER displays 'M' (multiple access) if the queue entry is accessed by the GET Service for browse on more than one sharing system.

PAGES
The number-of-pages column. For list output, VSE/POWER displays the number of pages; for punched output (using the PDISPLAY PUN command), this column reads 'CARDS', and it displays the number of card-image records. Up to 7 digits may be displayed. When the number exceeds 9999999 then only 9999999 is displayed.

For list output, VSE/POWER displays the number of pages

- left to be written to the output device which selected the list entry for printing, or
- left to be passed via the spool-access support GET service to an application.

This is only done if the disposition column contains an '*' (asterisk), indicating that the entry is being processed.

At output spooling time, VSE/POWER increments the page count of the queue entry by 1 at

- every occurrence of a "skip to channel 1" command, or
- overflow of the intended print page as defined by an FCB or LTAB.

Any lines spooled *before* a "skip to channel 1" or before page overflow occurs do not increment the VSE/POWER page count. Therefore, the user

program's page count and the VSE/POWER page count may differ by 1 in this case. If a queue entry is added to the list queue containing some spooled lines but still having a page count of 0, VSE/POWER will set the page count to 1.

For output containing CPDS records (X'5A'), for example resulting from a PAGEDEF or a FORMDEF macro, the page count is derived from the structured field identifiers

- BPG (Begin Page)
- IDM (Invoke Data Map)
- IMM (Invoke Media Map)

Their sequence and mixture with non-CPDS records increments the page count such that it comes as near as possible to the actual number of pages printed for this queue entry by the Print Support Facility (PSF/VSE). For structured field identifiers, refer to *PSF Data Stream Reference*, SH35-0073. For VSE/POWER handling, refer to "CPDS Record Spooling and Page Counting" on page 145.

VSE/POWER considers every record as a page if the queue entry contains data of one of the following formats:

- Basic mapping format (BMS)
- 3270 mapping

```
1   1R46I    LIST QUEUE   P D C S  PAGES   CC FORM B
2   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)
     ...  ...  ...
```

Figure 27. Display of the List Queue (continued)

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, the number shows how often VSE/POWER transfers the output to the IBM 3800. VSE/POWER indicates this by an * preceding the number of transmissions. That depends on the number of copy groups specified in COPYG. For example, if COPYG=(3,6,3,6,3,6,3,6) then CC shows '* 8'.

FORM
The form-number column.

B This column may indicate with an asterisk ('*') that the queue entry is "being browsed". The MACC count of a FULL=YES display will identify the number of browsing tasks. See also "Browsing of Queue Entries" on page 121.

```
2  1R46I  PWR001   00162 3 D B 1      3  * 8 FFFF   FROM=(SYSA)
```

The second line displays the first job in the queue and displays the following:

PWR001
The name by which VSE/POWER knows this job.

00162 The VSE/POWER assigned job number.

3 D B 1
Explained by P D C S – See the explanations to the preceding status display line **1** on page 253.

3 Number of pages.

*** 8** When copy grouping (COPYG) is specified, the number shows how often VSE/POWER will transfer the output to the IBM 3800 printer.

FFFF The identification of the output form.

FROM=(SYSA)
The originator's user ID.

```
      1R46I    LIST QUEUE   P D C S  PAGES   CC FORM B
      1R46I  PWR001   00162 3 D B 1      3  * 8 FFFF   FROM=(SYSA)
3           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)
        ...  ...  ...
```

Figure 28. Display of the List Queue (continued)

3 `FLASH=FLSH BURST COPIES= 36`

This line is 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. These are 3800 printer operands of the * $$ LST statement and are displayed only if the output was spooled to a 3800 printer at job creation time. On this line:

FLASH=FLSH
The name (FLSH in the example) of the forms overlay. When spooled to a non-3800, only the FLASH= operand takes effect and is then displayed as FLASHN38=.

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 operand of the * $$ LST statement. For example, if COPYG=(3,6,3,6,3,6,3,6) then COPIES= 36 is displayed.

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 ID (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 being

- printed on a printer started for class Z and attached to the system with the ID 2 (PAGES=211 means the number of pages left to be printed, also called 'remaining'), or
- retrieved by a Spool-Access GET task, or
- viewed by one or more Spool-Access GET-BROWSE tasks (for additional information, see the $STA0009 queue entry for "Format 1-4: PDISPLAY LST,FULL=YES" on page 257), or
- saved to tape by a POFFLOAD task.

Notes:

1. If the output entry was currently being processed
   - by spool-access GET, the display would also present 'remaining' pages
   - by POFFLOAD, the display would present 'total' number of pages.
2. Only the system with the active list task (system_id=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 LST Queue display may also be requested by a programmed Spool-Access CTL command for messages in fixed format (either by D LST, D ALL or D TOTAL).

Format 1-4: PDISPLAY LST,FULL=YES

```
1  1R46I    LIST QUEUE   P D C S   PAGES  CC FORM B
2  1R46I  PAYROLL  00172 8 K P       815   2 F001   FROM=(HBAK)
3         D=07/12/2005 U='MONTHLY REPORT  ' DBGP=000322 L=00004620
          QNUM=00022
   1R46I  COMPILE1 00175 5 D P       100   1        S=001   TO=(ICHA)
          D=07/12/2005 U='SUPERV ASSEMBLY ' S=001 DBGP=000475 L=00006056
          QNUM=00063
   1R46I  COMPILE1 00175 5 D P        83   1        S=002   TO=(ICHA)
          D=07/12/2005 U='SUPERV ASSEMBLY ' S=002 DBGP=000273 L=00002789
          QNUM=00007
   1R46I  DIRLST   04711 9 D X        35   1        TO=(JDAR)
          D=07/12/2005 DBGP=000026 L=00000100 RF=BMS
          DIST=BOX25
          QNUM=00073
   1R46I  $STA0009 00009 9 L A         1   1      * TO=(ANY)
          D=07/12/2005 DBGP=000001 L=00000040
          MACC=1|0|0|2|0|0|0|0 QNUM=00001
          UNSEC
     ...  ...  ...
```

Figure 29. Full Display of the List Queue

VSE/POWER provides all of the displayable information as shown above for all queue entries:

- During VSE/POWER startup if SYSLST is assigned.
- After a PEND cuu command.

```
1  1R46I    LIST QUEUE   P D C S   PAGES  CC FORM B
```

The status display header line. For an explanation, see **1** of the first example in this section on page 253.

```
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 ID (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=node_id(user_id).
```

```
3         D=07/12/2005 U='MONTHLY REPORT  ' DBGP=000322 L=00004620
```

A display line provided if FULL=YES is specified. On the line:

D=07/12/2005
: The job creation date (in the date format active at creation time of the list queue entry PAYROLL; see "Functions Not Expecting Date Format Changes" on page 128).

U='MONTHLY REPORT '
: The 16-byte user information supplied by the originator via UINF= (also USER=), which is recorded left-justified in an internal field whose default value is blanks (X'40') (in which case the field is not displayed).

DBGP=000322
: The number of allocated DBLK groups on disk for the queue entry; 322 in this example (0 for tape display).

L=00004620
The number of lines that make up the job, 4620 in this example. (A write-without-line-spacing command counts as line.) If the disposition column contains an '*' (asterisk), indicating that the entry is processing, VSE/POWER displays the number of lines left to be written to the selected output device, or left to be passed via the spool-access support GET service to an application.

Note: For entries with record format (RF=) CPDS, lines (L=) is displayed as recorded lines (RL=). This is to remind the operator that the displayed amount must not be interpreted as traditional ″print lines″.

```
      1R46I    LIST QUEUE   P D C S   PAGES  CC FORM B
      1R46I  PAYROLL  00172 8 K P       815   2 F001   FROM=(HBAK)
             D=07/12/2005 U='MONTHLY REPORT  ' DBGP=000322 L=00004620
             QNUM=00022
 4    1R46I  COMPILE1 00175 5 D P       100   1        S=001 TO=(ICHA)
 5           D=07/12/2005 U='SUPERV ASSEMBLY ' S=001 DBGP=000475 L=00006056
             QNUM=00063
 6    1R46I  COMPILE1 00175 5 D P        83   1        S=002 TO=(ICHA)
 7           D=07/12/2005 U='SUPERV ASSEMBLY ' S=002 DBGP=000273 L=00002789
             QNUM=00007
 8    1R46I  DIRLST   04711 9 D X        35   1        TO=(JDAR)
             D=07/12/2005 DBGP=000026 L=00000100 RF=BMS
             DIST=BOX25
             QNUM=00073
      1R46I  $STA0009 00009 9 L A         1   1      * TO=(ANY)
             D=07/12/2005 DBGP=000001 L=00000040
             MACC=1|0|0|2|0|0|0|0|0 QNUM=00001
             UNSEC
      ...  ...  ...
```

Figure 30. Full Display of the List Queue (continued)

4 `1R46I  COMPILE1 00175 5 D P       100   1        S=001 TO=(ICHA)`

The first display line for the second job in the queue. The line includes a job suffix (output-segment number) S-001 and the ID (ICHA in the example) of the user to which the output is to be routed.

5 `D=07/12/2005 U='SUPERV ASSEMBLY '   S=001 DBGP=000475 L=00006056`

This is similar to line **3** of the example, except that the line includes a job suffix (or output-segment number), S=001.

Note: The job suffix is shown in both the first and the second display line, for compatibility.

6 `1R46I  COMPILE1 00175 5 D P        83   1      S=002 TO=(ICHA)`

This is similar to line **4** of the example.

7 `D=07/12/2005 U='SUPERV ASSEMBLY ' S=002 DBGP=000273 L=00002789`

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.

```
      1R46I    LIST QUEUE   P D C S   PAGES  CC FORM B
      1R46I  PAYROLL  00172 8 K P       815   2 F001   FROM=(HBAK)
             D=07/12/2005 U='MONTHLY REPORT  ' DBGP=000322 L=00004620
 15          QNUM=00022
      1R46I  COMPILE1 00175 5 D P       100   1        S=001 TO=(ICHA)
             D=07/12/2005 U='SUPERV ASSEMBLY ' S=001 DBGP=000475 L=00006056
             QNUM=00063
      1R46I  COMPILE1 00175 5 D P        83   1        S=002 TO=(ICHA)
             D=07/12/2005 U='SUPERV ASSEMBLY ' S=002 DBGP=000273 L=00002789
             QNUM=00007
      1R46I  DIRLST   04711 9 D X        35   1        TO=(JDAR)
  9          D=07/12/2005 DBGP=000026 L=00000100 RF=BMS
 10          DIST=BOX25
             QNUM=00073
 11   1R46I  $STA0009 00009 9 L A         1   1      * TO=(ANY)
 12          D=07/12/2005 DBGP=000001 L=00000040
 13          MACC=1|0|0|2|0|0|0|0|0 QNUM=00001
 14          UNSEC
      ...  ...  ...
```

Figure 31. Full Display of the List Queue (continued)

9 `D=07/12/2005 DBGP=000026 L=00000100 RF=BMS`

No user information was supplied. On the line:

DBGP=000026
The number of allocated DBLK groups on disk for the queue entry; 26 in this example.

RF=BMS
The record format is other than machine operation code (MCC) and, therefore, indicated by VSE/POWER. The following abbreviations are used:

BMS basic mapping support

SCS standard character stream data

CPDS composed page data stream (for details refer to "CPDS Record Spooling and Page Counting" on page 145)

ESC escape character stream

3270 3270 data stream

ASA American Standards Association data stream

Note: If ASA, ESC, 3270, BMS, SCS, or MCC has been intermixed during output spooling with CPDS records, then only the prevailing RF=CPDS is displayed.

10 `DIST=BOX25`

The distribution code which VSE/POWER will pass to VM by the CP CLOSE command for a VM writer task. It is printed on VSE/POWER separator pages for non-VM writer tasks.

11 `1R46I  $STA0009 00009 9 L A         1   1      * TO=(ANY)`

This is similar to line **4** of the example, however with the target userid='ANY'. Parallel GET-Service for browse requests have the list queue entry in access. This is reflected by the '*' indicator in column 'B' (see the explanation under 'Format 1-3').

12 `D=07/12/2005 DBGP=000001 L=00000040`

This is similar to line **3** of the example. For entries being accessed by browse requests, both

- the page count of line **11** and
- the line count L=40

show the total page/line count.

13 `MACC=1|0|0|2|0|0|0|0|0 QNUM=00001`

The Multiple Access Counts (as displayed on a sharing system) show how many parallel GET-Service-for-browse requests are active per different sharing system, ranging from system 1 to 9. In this case, we see one browsing task on system-ID =1 and two browsing tasks on system-ID=4. The maximum number of browsing tasks per system-ID may be 15. On non-shared systems the Multiple Access Count can appear as MACC=007. In that case, the maximum number of browsing tasks may be 255.

QNUM shows the VSE/POWER internal queue record number in a 5-digit decimal format for use in Spool Access Support Direct CTL or GET requests as described in *VSE/POWER Application Programming*. You can also specify this number for the 'CQNUM=' selection operand of the PALTER/PDELETE/PHOLD/PRELEASE commands if output entries cannot be distinguished by jobname and jobnumber or other attributes.

14 `UNSEC`

This line indicates that VSE/POWER Spool Access Protection has been activated for the system and the output spool entry is specified not to be spool access protected against access or manipulation.

15 `QNUM`

This line indicates the internal queue entry number of this entry.

Format 1-5: PDISPLAY XMT

```
1   1R46I    XMIT QUEUE   P D C I  LINES B
    1R46I  PAUSEC   00149 3 D C R      4   TO=JEC1(USERA)
    1R46I  PROC130  00050 3 D A L     99   TO=JEC1
    1R46I  PUN3     00003 3 D 6 P    100   TO=JEC2(USERB)
   ...  ...  ...
```

Figure 32. Display of the XMIT (Transmission) Queue

1 1R46I XMIT QUEUE P D C I LINES B

The header line for the display. On the line:

P The priority column with transmission priority, which influences selection by job or output transmitters. For details, see "Priority in the Transmit Queue" in *VSE/POWER Networking*.

D The disposition column, presenting the transmission disposition valid for the XMT queue. For details, see "Disposition in the Transmit Queue" in *VSE/POWER Networking*.

C The class column. It presents the processing class of local (RDR/LST/PUN) queues. It has no effect on transmission because transmitter tasks are not oriented by class but rather by 'job' or 'output' type (see the 'I' column).

I Identifier of the original queue entry type. R for RDR, L for LST, P for PUN.

LINES
The number of lines to be transmitted in case of LST type. The number of statements to be transmitted in case of RDR/PUN type. For entries being transmitted (D=*), LINES means lines/statements left to be transmitted, also called 'remaining' lines. When the number exceeds 9999999 then only 9999999 is displayed.

Note: If PDISPLAY,...,XMTL=PG has been specified, then the 'LINES' column appears as 'PG/CD', namely the number of pages to be transmitted for LST type or cards (statements) to be transmitted for RDR/PUN byte entries. In this case, for LST entries **being** transmitted (D=*), the total number of spooled pages is presented and not the remaining number of pages.

B This column may indicate with an asterisk ('*') that the queue entry is "being browsed". The MACC count of a FULL=YES display will identify the number of browsing tasks.

For all other explanations see **1** on page 253.

If a job or output is to be routed to a user at another node, VSE/POWER displays the name and user of this node in the format:

```
TO=node_id(user_id)
```

If the entry is directed to the central operator of another node for execution of a job or local printing of list output for example, no user_id is specified. The display has then the format:

```
TO=node_id
```

The XMT queue display may also be requested by a programmed Spool-Access CTL command for messages in fixed format (either by D XMT, D ALL or D TOTAL).

Format 3: PDISPLAY CRE

```
1  1R4BI   CREATE QUEUE  C I  LINES B DBGP   QNUM  TASK    OWNER
   1R4BI  NEWJOB   00114 3 R     78   000001 00078 SAS PSP SAS=APPLID
   1R4BI  LISTLOG  00116 0 L   1999 * 000002 00233  BG FEE JOB=MYOWNJOB
                                                        S=002 MACC=2
   1R4BI  NJEOUTP  00118 R P    339   000001 01245 RV1-OUT NOD=OTHERNOD
   1R4BI  JOBR020  00231 Y R    129   000001 09056 RDR SNA REM=R020
   1R4BI  LOCALJOB 00233 T R    551   000002 12005 RDR 00C
```

Figure 33. Display of Entries Currently In Creation

```
1  1R4BI   CREATE QUEUE  C I  LINES B DBGP   QNUM  TASK    OWNER
```

The header line for the display contains the following elements:

C The class column. It shows the processing class to which the in-creation entry will be queued when creation has completed.

I Identifier of the actual queue entry type: R for RDR, L for LST, P for PUN, showing to which type of queue (except XMT) the entry will be queued when creation has completed.

LINES The number of lines already spooled in case of LST type. The number of statements already spooled in case of RDR or PUN type. When the number exceeds 9999999 then only 9999999 is displayed.

B This column indicates by an '*' that the queue entry is currently being browsed by at least one Spool-Access Support BROWSE task. The number of currently browsing tasks (even on different sharing VSE/POWER systems) is identified by the MACC count of the second line in each queue entry display.

DBGP The number of allocated DBLK groups on disk for this queue entry.

QNUM Shows the VSE/POWER internal queue record number in a 5-digit decimal format, for use in the PALTER ...,SEGMENT= command.

TASK Identifies which task on the local (shared) system is creating this queue entry, or whether the entry is being created on another shared system. For details, see the explanation of selected entries within this display. For task types, see "Format 8: PDISPLAY A" on page 270.

OWNER Provides further qualification of the creating task. See the explanation of selected entries within this display.

```
     1R4BI   CREATE QUEUE  C I  LINES B DBGP   QNUM  TASK    OWNER
2    1R4BI  NEWJOB   00114 3 R     78   000001 00078 SAS PSP SAS=APPLID
3    1R4BI  LISTLOG  00116 0 L   1999 * 000002 00233  BG FEE JOB=MYOWNJOB
                                                          S=002 MACC=2
4    1R4BI  NJEOUTP  00118 R P    339   000001 01245 RV1-OUT NOD=OTHERNOD
5    1R4BI  JOBR020  00231 Y R    129   000001 09056 RDR SNA REM=R020
6    1R4BI  LOCALJOB 00233 T R    551   000002 12005 RDR 00C
```

Figure 34. Display of Entries Currently In Creation (continued)

```
2    1R4BI  NEWJOB   00114 3 R     78   000001 00078 SAS PSP SAS=APPLID
```

A new job is being created by Spool-Access Support with SAS application ID APPLID.

```
3    1R4BI  LISTLOG  00116 0 L   1999 * 000002 00233  BG FEE JOB=MYOWNJOB
                                                          S=002 MACC=2
```

New list output is being created by job MYOWN running in partition BG and spooling to FEE. The entry is currently being browsed (B column = '*') by 2 browse tasks as indicated by MACC=2.

```
4    1R4BI  NJEOUTP  00118 R P    339   000001 01245 RV1-OUT NOD=OTHERNOD
```

New punch output is being received from PNET node OTHERNOD.

```
5    1R4BI  JOBR020  00231 Y R    129   000001 09056 RDR SNA REM=R020
```

New job is being received from RJE,SNA station with Remote ID R020.

```
6    1R4BI  LOCALJOB 00233 T R    551   000002 12005 RDR 00C
```

New job is being read in by local card reader 00C.

Notes

- On shared systems, the heading line shows the SYSID of the system where the PDISPLAY CRE command is executed and will return only queue entries in creation on the SYSID where the command is executed:

  ```
  1R4BI  CREATE SYSID=1 C I  LINES B DBGP   QNUM  TASK     OWNER
  1R4BI  NEWJOB   00114 S R     78   000001 00078 SAS PSP SAS=APPLID
  ```

- For 'PDISPLAY CRE,ALLSYS,' queue entries in creation on other systems shows SYSID=n instead of a task identifier.

  ```
  1R4BI  CREATE SYSID=1 C I  LINES B DBGP   QNUM  TASK     OWNER
  1R4BI  AUTONAME 00116 A R      0   000001 00328 SYSID=2
  ```

- Nearly all in-creation queue entries need only one display line.
- Only queue entries in creation by execution writers and already segmented via RBS segmentation show the current suffix number in a second line.
- Only queue entries in the in-creation state that have been accessed for browsing identify this by the MACC (Multiple ACcessCount) in a second line, and the 'B' column shows an '*'.
- The Create Queue display may also be ordered by a programmed Spool-Access CTL request for messages in fixed format (either by D CRE or D TOTAL). The following flag in DSECT PXFMDSCT identifies a queue entry of the Create Queue:
  - PXFM3CRE (X'80') in PXFMFLG3 indicates that the entry is being created.

Format 4: PDISPLAY DEL

```
1  1R4BI  DELETION QUEUE P D C I  LINES B
   1R4BI  NDT631   00114 3 K 8 R     78 * FROM=(WALB)
          D=03/09/2005 DBGP=000001
          MACC=1|0|0|0|1|0|0|0|0 QNUM=00012
   1R4BI  NDT631   00093 3 D 8 R     78 * FROM=(WALB)
          D=03/09/2005 DBGP=000001
          MACC=3|0|0|0|1|0|0|0|0 QNUM=00344
   1R4BI  NDT631   00114 3 D H L    500 * TO=OTHERVSE(WALB) FROM=(WALB)
          D=03/12/2005 DBGP=000003
          MACC=1|0|0|0|0|0|0|0|0 QNUM=00465
   1R4BI  NDT631   00114 3 D H L    266 * FROM=(WALB)
          D=03/12/2005 S=002 DBGP=000002
          MACC=0|0|2|0|0|0|0|0|0 QNUM=01986
```

Figure 35. Display of Entries Currently In Delayed Deletion

```
1  1R4BI  DELETION QUEUE P D C I  LINES B
```

The header line for the display contains the following elements:

P The priority column.

D The Disposition column presents the queue entry's disposition as found

- before an operator issued a PDELETE request, or
- by an Update task before setting it to active

C The class column.

I Identifier of the actual queue entry type: R for RDR, L for LST, P for PUN.

LINES

The number of lines spooled for a queue entry if LST type. The number of statements spooled if RDR or PUN type. When the number exceeds 9999999 then only 9999999 is displayed.

B This column indicates by an '*' that the queue entry is currently still being browsed by at least one Spool-Access Support BROWSE task (otherwise, the entry would no longer be in the deletion queue). The number of currently browsing tasks is identified by the MACC count of the third line per queue entry display. In this sample, the MACC count shows slots with the number of browsers per sharing system (1-9).

This column may indicate by a 'D' that a queue entry is bound for Delition, meaning the init/termination task should request 'final freeing' for this entry.

```
    1R4BI  DELETION QUEUE P D C I  LINES B
2   1R4BI  NDT631   00114 3 K 8 R     78 * FROM=(WALB)
           D=03/09/2005 DBGP=000001
           MACC=1|0|0|0|1|0|0|0|0 QNUM=00012
3   1R4BI  NDT631   00093 3 D 8 R     78 * FROM=(WALB)
           D=03/09/2005 DBGP=000001
           MACC=3|0|0|0|1|0|0|0|0 QNUM=00344
4   1R4BI  NDT631   00114 3 D H L    500 * TO=OTHERVSE(WALB) FROM=(WALB)
           D=03/12/2005 DBGP=000003
           MACC=1|0|0|0|0|0|0|0|0 QNUM=00465
5   1R4BI  NDT631   00114 3 D H L    266 * FROM=(WALB)
           D=03/12/2005 S=002 DBGP=000002
           MACC=0|0|2|0|0|0|0|0|0 QNUM=01986
```

Figure 36. Display of Entries Currently In Delayed Deletion (continued)

```
2   1R4BI  NDT631   00114 3 K 8 R     78 * FROM=(WALB)                  {1}
           D=03/09/2005 DBGP=000001
           MACC=1|0|0|0|1|0|0|0|0 QNUM=00012
```

The entry is RDR (I=R) queue entry. Disposition 'K' shows that this entry has been deleted by a command.

```
3   1R4BI  NDT631   00093 3 D 8 R     78 * FROM=(WALB)                  {2}
           D=03/09/2005 DBGP=000001
           MACC=3|0|0|0|1|0|0|0|0 QNUM=00344
```

The entry is a RDR (I=R) queue entry. Disposition 'D' shows that this entry has been processed by an update task (such as an execution reader) and then deleted.

```
4   1R4BI  NDT631   00114 3 D H L    500 * TO=OTHERVSE(WALB) FROM=(WALB){3}
           D=03/12/2005 DBGP=000003
           MACC=1|0|0|0|0|0|0|0|0 QNUM=00465
```

The entry is an XMT (see TO=OTHERVSE) queue entry (LST type, I=L). Disposition 'D' shows that this entry was transmitted to another node and then deleted by the transmitting task.

```
5   1R4BI  NDT631   00114 3 D H L    266 * FROM=(WALB)                  {4}
           D=03/12/2005 S=002 DBGP=000002
           MACC=0|0|2|0|0|0|0|0|0 QNUM=01986
```

The entry is a LST (I=L) queue entry. Disposition 'D' and no TO user ID show that this entry was printed by a local list task and then deleted. This queue entry has a SEGMENT number of 002.

Each queue entry could also have been deleted by a PDELETE command. For background information, see "Delaying the Deletion of Browsed Queue Entries" on page 122.

Note: The DELETION queue display may also be requested by a programmed Spool-Access CTL command for messages in fixed format (either D DEL or D TOTAL). The following flag in DSECT PXFMDSCT identifies a queue entry of the deletion queue:

- PXFM3DEL (X'40') in PXFMFLG3 indicates that the queue entry is in delayed-deletion state.

Format 6: PDISPLAY BIGGEST,LIMIT=7

```
1 1R4BI  07 BIGGEST SORTED C I CARD/LINE    DBGP  QNUM SUF     PAGES QUE
  1R4BI  01 JOBM271  07005 A L         0 0001889 32406            0 CRE
  1R4BI  02 LISTLOG  51005 B L   1000789 0000992 44406        31770 DEL
  1R4BI  03 NJEOUT   00508 G L    767500 0000789 02406 001    20000 LST
  1R4BI  04 NJEOUT   00508 G L    522004 0000499 02411 002    16307 LST
  1R4BI  05 LIBRPUN  06133 P P    377000 0000344 04408               PUN
  1R4BI  06 NJEPUN   00235 G P    350781 0000340 01401               XMT
  1R4BI  07 FATJOB   03134 C R    250060 0000266 11566               RDR
```

Figure 37. Display of Entries Currently In Creation

```
1 1R4BI  07 BIGGEST SORTED C I CARD/LINE    DBGP  QNUM SUF     PAGES QUE
```

The header line for the display contains the following elements:

07 The ″sorted number″ column. In the header line repeating the LIMIT=value (or its default of *10*) of the display command. In the example, *01* indicates the largest entry (with respect to the DBGP column), followed in descending order *02, 03, ...* as identified by jobname and jobnumber.

C The class column — class is not displayed for internal $SPLnnnn. entries

I Identifier of the actual queue entry type: 'R' for RDR, 'L' for LST, 'P' for PUN, which may be helpful to distinguish XMT queue entries.

CARD/LINE Displays the 9–digit value of the lines or cards, instead of the 7–digit value of other displays. For entries ″in-creation″(queue ″CRE″) the value displayed is ″0″ — the actual value in this case can be seen using the PDISPLAY CRE command (and, in any case, the PDISPLAY BIGGEST indicates the actual spool space used in the DBGP column).

DBGP The number of allocated DBLK groups on disk for this queue entry as a 7–digit value. The display is sorted in descending DBGP order.

QNUM Shows the VSE/POWER internal queue record number in a 5-digit decimal format.

SUF The segment suffix number (1–127) of RBS= segmented list or punch output.

PAGES The number of list pages displayed as a 9–digit number for LST output. For entries ″in-creation″(queue ″CRE″) the value displayed is ″0″ — the actual value in this case can be seen using the PDISPLAY CRE command (and, in any case, the PDISPLAY BIGGEST indicates the actual spool space used in the DBGP column).

QUE The name of the VSE/POWER queue where the named queue entry resides and can be addressed by other commands for further processing:

- RDR/LST/PUN/XMT for deletion, transmission and offloading

- CRE for segmenting of partition output or flushing of partition so that output gets added to the LST/PUN/XMT queue(s) for further processing.
- DEL for identification of the browsing task using the PDISPLAY A,SAS command followed by PSTOP SAS,connect_id so that the browsing task is terminated and the queue entry finally deleted.

Note: For active (DISP=*) spool entries being processed by a an output LST, PUN, or XMT task, this display of the cards/lines/pages differs in that here the ″total″ value is shown, whereas in the output task the ″remaining to be processed″ value is displayed.

Format 7: PDISPLAY Q

```
   1R49I  QUEUE FILE 004% FULL - 703 FREE QUEUE RECORDS
3  1R49I  USED QUEUE RECORDS: 31, CRE-Q: 2, DEL-Q: 0
   1R49I  RDR-Q: 16, LST-Q: 10, PUN-Q: 1, XMT-Q: 2
1  1R49I  QUEUE FILE EXTENT ON CKD-150, SYS001, 1005, 8
4  1R49I  DATA FILE 003% FULL - 1230 FREE DBLK GROUPS
   1R49I  CURRENT DBLK SIZE=07548, DBLK GROUP SIZE=00008
2  1R49I  DATA FILE EXTENT 1 ON CKD-151, SYS002, 6690, 2025
   1R49I  DATA FILE EXTENT 2 ON CKD-152, SYS003, 165, 900
   1R49I  ACCOUNT FILE 1 % FULL
   1R49I  ACCOUNT FILE EXTENT ON CKD-151, SYS000, 8715, 90
```

Figure 38. Display of the Spool File Characteristics

```
1  1R49I  QUEUE FILE EXTENT ON CKD-150, SYS001, 1005, 8
```

The disk extent of the queue file (and of the data and account files) is shown as:

CKD The disk device type (can also be FBA)

150 The physical disk address

SYS001 The logical unit used to assign the queue file to the physical disk address.

1005 The start track of the queue file (or starting FBA block)

8 The number of tracks used for the queue file (or number of FBA blocks)

```
2  1R49I  DATA FILE EXTENT 1 ON CKD-151, SYS002, 6690, 2025
   1R49I  DATA FILE EXTENT 2 ON CKD-152, SYS003, 165, 900
```

Note: If the last data file extent entry reads (for example)

```
1R49I  DATA FILE EXTENT 3 ON CKD-152, SYS003, 1500, 900 (IN FORMAT)
```

data file extension during a warm start has been requested (see “Extending the Data File During Warm Start” on page 38). The additional extent is currently being formatted and its data block groups are not yet available for spooling.

```
3  1R49I  USED QUEUE RECORDS: 31, CRE-Q: 2, DEL-Q: 0
   1R49I  RDR-Q: 16, LST-Q: 10, PUN-Q: 1, XMT-Q: 2
```

The total spooling entries and their distribution over the queues shown as:

USED QUEUE RECORDS: 31
The total used spooling entries, as distributed over the physical(RDR/LST/PUN/XMT) and logical queues(CRE/DEL) in the following text.

The disk extent(s) of the data file with the same information as described for the queue file. Up to 32 extents can be presented. See also "Organization of the Data File" on page 28.

```
4 1R49I  DATA FILE 003% FULL - 1230 FREE DBLK GROUPS
```

The "FULL" amount means total number of DBLK groups of the Data File minus the free DBLK groups. It comprises DBLK groups:
- used by "in creation" queue entries (D CRE)
- used by entries in the RDR/LST/PUN/XMT queues
- used by "in deletion" queue entries (D DEL)
- "being freed" of a queue entry that has just left the DEL queue and has not yet completed this process
- lost due to I/O or logic error.

For exact numbers of "in deletion", "being freed" or lost DBLK groups see Figure 7 on page 46.

Format 8: PDISPLAY A

```
1 1R48I  LST,00E,T,2,,,  POWERM12 00008 T
  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=00004919 OF 00010018
         NODE=BOEVS03
  1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B   0 LINES SPOOLED
  1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
  1R48I  SAS,00006, SAS=GCMAPPL,GCMUSER, REQ=GCM
  1R48I  F3,FEE,,  VTAMSTRT,00163,A      19 LINES SPOOLED,QNUM=00004
      ...  ...  ...
```

Figure 39. Display of Currently Active VSE/POWER Tasks

1 `1R48I  LST,00E,T,2,,,  POWERM12 00008 T`

The first line on the display shows the following:

LST The type of the involved task (list writer in the example). Instead of LST, one of the following may show up:

BG,Fn,Qm
Execution tasks of static or dynamic partitions

DEV Device-service

C-RV Console receiver

C-TR Console transmitter

J-RVn Job receiver

J-TRn [1]
Job transmitter

L-OFF POFFLOAD SAVE|BACKUP|PICKUP

O-RVn [1]
Output receiver

O-TRn [1]
Output transmitter

PSP PUTSPOOL service

PUN Punch writer

RDR Reader

R-OFF POFFLOAD LOAD|SELECT

SAS Spool-access support

Note: [1] *n* refers to the number of transmitter or receiver tasks.

00E The address of the I/O 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.

2 The number of buffers 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.

T The job class assigned to the job.

```
   1R48I  LST,00E,T,2,,,  POWERM12 00008 T
2  1R48I  LST,00F,V,2,VM,,  VMOUTP12 00016 V
3  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
   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=00004919 OF 00010018
          NODE=BOEVS03
   1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B    0 LINES SPOOLED
   1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
   1R48I  SAS,00006, SAS=GCMAPPL,GCMUSER, REQ=GCM
   1R48I  F3,FEE,,  VTAMSTRT,00163,A      19 LINES SPOOLED,QNUM=00004
        ...  ...  ...
```

Figure 40. Display of Currently Active VSE/POWER Tasks (continued)

2 `1R48I  LST,00F,V,2,VM,,  VMOUTP12 00016 V`

The same as **1** of this example, except for:

VM This indicates that the output of the job VMOUTP12 is directed to a printer under control of VM.

P390 P390 may be displayed instead of VM. It indicates that the printer, for example at address 00F is emulated by the P390 Device Manager. For details see ″PSTART LSTP390″ on page 360.

The last position (shown as empty in this example) may present the “SEP/NOSEP/ISEP/ISEPJ/DLSEP” startup options.

3 `1R48I   F2,FEC,A,      INACTIVE,`

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 I/O 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 because no queue entry selectable for processing can be found at the moment.

(PENDING)
Instead of INACTIVE you may see PENDING. PENDING means that an attempt is being made to search a selectable queue entry, but the task waits for the DMB resource (locked by another task).

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:

PRT1 The printer device name.

DDSIM
The XPCC application ID of the owning device-driving system (DDS).

ABCD The output classes that the printer is started for (A,B,C,D).

PRI01 The job name.

00150 The job number (150).

A The job class of the accessed output entry.

Note: When the selected task terminates during collection of PRT1,... information, the display line may be completed by the text: 'TASK IN TERMINATION PROCESS'.

```
       1R48I  LST,00E,T,2,,,  POWERM12 00008 T
       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
7      1R48I  SAS,00005, SAS=SYSCICS1,TOOL1, REQ=PUT, REPORT  ,04711,A
              100 RECORDS SPOOLED
8      1R48I  C-RV,036,     AWAITING     NODE=BOEVS03
       1R48I  O-TR1,036,*,  WSKTNL02,00167,L  LEFT=00004919 OF 00010018
              NODE=BOEVS03
       1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B       0 LINES SPOOLED
       1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
       1R48I  SAS,00006, SAS=GCMAPPL,GCMUSER, REQ=GCM
       1R48I  F3,FEE,,  VTAMSTRT,00163,A      19 LINES SPOOLED,QNUM=00004
         ...  ...  ...
```

Figure 41. Display of Currently Active VSE/POWER Tasks (continued)

```
7  1R48I  SAS,00005, SAS=SYSCICS1,TOOL1, REQ=PUT, REPORT  ,04711,A
         100 RECORDS SPOOLED
```

If a spool-access support user task is involved, VSE/POWER displays:

00005 The connect-ID assigned by VSE/POWER.

SYSCICS1
The XPCC application ID.

TOOL1
The user ID.

PUT The function being performed.

REPORT ,04711,A
Job name (REPORT), job number (04711), and job class (A), if present.

100 The number of records spooled so far.

8 `1R48I  C-RV,036,     AWAITING     NODE=BOEVS03`

If a networking task is involved, VSE/POWER displays:

C-RV The type of the involved task; C-RV stands for console receiver

036 The address of the BSC line or CTCA address used for the network communication

AWAITING
The status of the task

BOEVS03
The node ID of the node with which communication is established

```
1R48I  LST,00E,T,2,,,  POWERM12 00008 T
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
IR48I  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=00004919 OF 00010018
       NODE=BOEVS03
1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B       0 LINES SPOOLED
1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
1R48I  SAS,00006, SAS=GCMAPPL,GCMUSER, REQ=GCM
1R48I  F3,FEE,,  VTAMSTRT,00163,A      19 LINES SPOOLED,QNUM=00004
  ...  ...  ...
```

Figure 42. Display of Currently Active VSE/POWER Tasks (continued)

9 `1R48I  O-TR1,036,*,  WSKTNL02,00167,L  LEFT=00004919 OF 00010018`
`NODE=BOEVS03`

If a networking task is involved, VSE/POWER displays:

O-TR1
The type of the involved task. O-TR1 stands for output transmitter 1.

036 The address of the BSC line or CTCA address used for the network communication

***** The * indicates that the transmitter task may transmit entries of all possible classes

WSKTNL02
The VSE/POWER-assigned job name

00167 The job number

L The class of the queue entry being transmitted

4919 The number of records that remain to be transmitted out of the total number of records to be transmitted (10018)

BOEVS03
The node ID of the node with which communication is established.

```
        1R48I  LST,00E,T,2,,,  POWERM12 00008 T
        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=00004919 OF 00010018
               NODE=BOEVS03
 10     1R48I  BG,FFF,TAPE=182,,  PRINTLOG,00265,B       0 LINES SPOOLED
 11     1R48I  LST,02E,TAPE=183,,  DFHFCTSP,00102,Q
 12     1R48I  SAS,00006, SAS=GCMAPPL,GCMUSER, REQ=GCM
 13     1R48I  F3,FEE,,  VTAMSTRT,00163,A      19 LINES SPOOLED,QNUM=00004
          ...  ...  ...
```

Figure 43. Display of Currently Active VSE/POWER Tasks (continued)

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'.

12 `1R48I  SAS,00006, SAS=GCMAPPL,GCMUSER, REQ=GCM`

If the Job Event Message Retrieval function of the spool-access support is active, VSE/POWER displays `0006,GCMAPPL,GCMUSER` as explained for message **7**.

GCM The immediate Retrieve Job Event Message function being performed

(GCM-OPEN WAIT)
The corresponding function being performed, but waiting for an event message to be queued

13 `1R48I  F3,FEE,,  VTAMSTRT,00163,A      19 LINES SPOOLED,QNUM=00004`

Display of an execution writer task of partition F3. The task intercepts and spools output for the printer device assigned to FEE. The output entry being created has jobname VTAMSTRT, jobnumber 163, and is intended for class A of the LST queue. To this point, 19 lines have been spooled to disk.

QNUM=00004 is the internal VSE/POWER queue record number in a 5-digit decimal format for use in the PALTER ...,SEGMENT= command.

Format 9: PDISPLAY PNET

```
        pdisplay pnet,all
1       1RB7I *****  NDT NAME = NDTBV01   *****
2       1RB7I NODE     ROUTE1   ROUTE2  AUTH BSIZE APPLID/IPADDR  IPPORT SPORT
        1RB7I NODEA    ----- LOCAL -----           TPNS              175  2252
        1RB7I BOEVS03  SNA LINK          NET  1500 ZURS
        1RB7I BOEVS04  B/C LINK BOEVS03  JOB  2500
        1RB7I HURSL01  BOEVS03           JOB
        1RB7I BOETP5   TCP LINK          NET  4096 200.200.10.11     175
        1RB7I BOETP6   TCP LINK          NET  4096 BOE6.DE.IBM.COM   175
        1RB7I BOETP7   TCP LINK          NOJ  8000 BOETP7.NJECOM..   175
        1RB7I POWER262 SSL LINK          NET 32000 009.164.155.137        2252
```

Figure 44. Display of All Entries in the Active Network Definition Table (NDT) Version '06.0'

Notes:

1. The internal and displayed formats of the NDT have changed with VSE/POWER 6.6. Passwords of '06.0' NDTs are only shown in single-node displays.
2. You can find the Assembler source code for Network Definition Table 'NDTBV01' at "Example 6: VSE/POWER with PNET,CTC and SNA, and TCP and SSL Support" on page 96.

1 `1RB7I *****  NDT NAME = NDTBV01   *****`

A status display header line. It gives the name, NDTBV01, by which your active network-definition table is cataloged.

2 `1RB7I NODE     ROUTE1   ROUTE2   AUTH  BSIZE APPLID/IPADDR   IPPORT SPORT`

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
Find any of the following:

- SNA LINK, direct SNA link exists to node in NODE column
- TCP LINK, direct TCP/IP link exists to node in NODE column
- SSL LINK, direct TCP/IP SSL (Secure Sockets Layer) link exists to node in NODE column
- B/C LINK, either direct BSC or CTC link exists to node in NODE column. The link type (BSC or CTC) is not known prior to issuing PSTART PNET,nodeid.
- The name of a node to 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 to which a direct link exists. This link is used for transmission of data to the node named in the NODE column if either:

- No ROUTE1 node is defined because a direct link exists, but this direct link is not active, or
- The link to the node named under ROUTE1 is not active

AUTH
The level of command authorization (JOB, NOJ, or NET) for the operator of the node named in the NODE column. For more information about see *VSE/POWER Networking*.

BSIZE The length (in bytes) of the transmission buffers. Note that this is the buffer size of the NDT, not necessarily the buffer size actually used in transmission.

APPLID/IPADDR
Find either of the following:

- the column APPLID/IPADDR for the *local* node shows
  - the VTAM applid, either as specified or its default (local node name)
  - IPADDR cannot be specified for the local node; therefore, this static information is not displayed. To see the dynamically acquired IPADDR of the local host, use the 'PINQUIRE NODE=local-node' command
- the column APPLID/IPADDR for *remote* nodes shows either
  - the VTAM applid (8 char), if 'SNA LINK', or
  - the dotted decimal IP address (see 'TCP LINK', BOETP5), or
  - the dotted decimal IP address (see 'SSL LINK', POWER262), or
  - the symbolic IP address (max. 15 characters: see 'TCP LINK', BOETP6), or
  - a longer symbolic IP address (13 characters ending with '..' (see 'TCP LINK', BOETP7) which can further be displayed by a single 'PDISPLAY PNET,BOETP7' command.

```
      pdisplay pnet,boetp7
      1RB7I *****  NDT NAME = NDTBV01   *****
      1RB7I NODE     ROUTE1   ROUTE2  AUTH BSIZE APPLID/IPADDR  IPPORT SPORT
      1RB7I BOETP7   TCP LINK          NOJ  8000 BOETP7.NJECOM..   175
3     1RB7I          PWD=BOET7PWD IPADDR=BOETP7.NJECOMMUNICATIONS.DEUTSCHLAN
      1RB7I          D.IBM.COM
```

- the display of column APPLID/IPADDR for remote TCP/SSL LINK nodes may be appended by an 'E' due to the following:
  - If the command PDISPLAY PNET,ALL is used and IPEXTRAD has been specified for a node, the character 'E' is displayed immediately after the ip-address (without a separating blank), for example:

    ```
    1RB7I node-id  SSL LINK route2   JOB  4096 009.155.155.125E
    ```

    'E' is an indication that an external-ip-addr exists, by which the local node is known to the remote node node-id. Use the PDISPLAY PNET,node-id command to obtain a full display of the IPEXTRAD.
  - If the command PDISPLAY PNET,node-id is used and IPEXTRAD has been specified for the node, the character 'E' is displayed immediately after the ip-address (without a separating blank), and IPEXTRAD=ip-address is displayed in a second line, for example:

    ```
    PDISPLAY PNET,POWER272
    1RB7I POWER272 TCP LINK route2   JOB  4096 009.155.155.125E
    1RB7I          IPEXTRAD=009.164.155.125
    ```
  - If the second line displays a password and a logical ip-address, IPEXTRAD is displayed in between, for example:

    ```
    PDISPLAY PNET,POWER275
    1RB7I POWER275 SSL LINK POWER272 JOB  4096 HUS275         E
    1RB7I          PWD=H275 IPEXTRAD=009.155.155.175 IPADDR=HUS275
    ```

IPPORT
For 'TCP' nodes, the port number of the

- (Local node only) Local TCP/IP host partition on which VSE/POWER listens for connection requests from remote nodes
- (Remote-node entries) Foreign TCP/IP host (partition) to which VSE/POWER should connect as a result of a PSTART PNET,node-id command.

SPORT

For 'SSL' nodes, the secure port number of the

- (Local node only) Local TCP/IP host partition on which VSE/POWER listens for secure socket connection requests from remote nodes
- (Remote-node entries) Foreign TCP/IP host (partition) to which VSE/POWER should establish a secure socket connection request as a result of a PSTART PNET,node-id command.

PWD=password of 3

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 console. Furthermore, passwords are not displayed for an 'ALL' display request. Instead, a single node must be addressed, such as 'PDISPLAY PNET,BOETP7'.

Note: In general, the NDT display shows only statically generated information. Dynamically evaluated information can be requested by the PINQUIRE NODE=node_id command, which presents

- for the local node: the IP address, acquired by the GETHOSTID request, as soon as at least one NDT entry is found of type 'TCP LINK' or 'SSL LINK' and the interface to TCP/IP has been established.
- for a remote node: the IP address, either as specified in the NDT in dotted decimal format or as resolved from the specified symbolic address, as soon as the PSTART PNET,node_id command has been entered.

Format 11: PDISPLAY queue,TAPE=cuu,OUT=CON

A display of a tape will appear as for displays of spool queues on disk, with the following exceptions:

- The display messages begin with the message number 1R41I.
- The queue entries may display a tape volume number VOL=nnn.

A tape volume number VOL=nnn indicates that the tape contains an incomplete queue entry split into multi-volume fragments (here called a **CONTINUED** queue entry) and the sequence number of the given queue entry tape fragment, beginning with VOL=001 up to VOL=126. A greater number of volumes for an individual entry is possible (in which case VOL=*** is displayed); but, VSE/POWER is unable to properly sequence the tapes. In this case, the operator must manually record the tape sequence number as tapes are created (see message 1QA2I). The final tape volume is displayed as VOL=nnn(LAST) or VOL=***(LAST).

The following examples are of displays of POFFLOAD tapes containing continued queue entry(s) (if the displays were of a job execution spool tapes the total page count for a continued multi-volume entry - here MYJOB8 - would increase with increasing VOL=nnn numbers).

```
1R41I    LIST QUEUE   P D C S   PAGES   CC FORM B
1R41I  MYJOB1   00001 3 D A 1     110
1R41I  MYJOB2   00002 3 K A 1     205
...
1R41I  MYJOB8   00008 3 D B 1    2523
1R41I  VOL=001
```

Figure 45. Display of Tape Ending with Continued Multi-Volume Queue Entry

```
1R41I    LIST QUEUE   P D C S   PAGES   CC FORM B
1R41I  MYJOB8   00008 3 D B 1    2523
1R41I  VOL=002
```

Figure 46. Display of Tape Beginning and Ending with Same Continued Multi-Volume Queue Entry

```
1R41I    LIST QUEUE   P D C S   PAGES   CC FORM B
1R41I  MYJOB8   00008 3 D B 1    2523
1R41I  VOL=003(LAST)
1R41I  MYJOB9   00009 3 K B 1     136
...
```

Figure 47. Tape Begins with Continued Multi-Volume Queue Entry

```
1R41I    LIST QUEUE   P D C S   PAGES   CC FORM B
1R41I  MYJOB8   00008 3 D B 1    2523
1R41I  VOL=003(LAST)
1R41I  MYJOB9   00009 3 K B 1     136
...
1R41I  MYJOB21  00005 3 D C 1     233
1R41I  VOL=001
```

Figure 48. Tape Beginning and Ending with Continued Multi-Volume Queue Entries

Format 12: PDISPLAY DYNC,ALL

```
1  1Q6AI  ******   ACTIVE DYNAMIC CLASS TABLE DTR$DYNC.Z    ******
2  1Q6AI  CLS STATE   ACT/MAX ALLOC    SIZE  SP-GETV  PROFILE LUBS
   1Q6AI   G  ENAB      6 32     1M    200K    128K   ASIPRCG   15
   1Q6AI   X  ENAB      0 15     2M    300K    200K   ASIX      15
   1Q6AI   P  ENAB      4  8     1M    150K    128K   ASIPROCP  15
   1Q6AI   Q  DISAB     0  5     1M    100K    128K   ASICQ     12
   1Q6AI   M  SUSPEND   8  8     1M    100K    168K   ASIPROCM  15
     ...  ...  ...
```

Figure 49. Display of All Entries in the Active Dynamic Class Table

1 `1Q6AI  ******   ACTIVE DYNAMIC CLASS TABLE DTR$DYNC.Z    ******`

A status display header displaying the name of the active arbitrary library member.

2 `1Q6AI  CLS STATE   ACT/MAX ALLOC    SIZE  SP-GETV  PROFILE LUBS`

Second status display header line providing headings for the display columns of each dynamic class:

CLS The dynamic class for which dynamic partitions may be allocated to process jobs of the same VSE/POWER job input class.

STATE
The current state of the dynamic class with respect to permission of allocating a dynamic partition for this class:

ENAB The class is enabled for allocation of dynamic partitions.

DISAB
The class is disabled for allocation of dynamic partitions, mainly due to user specification or to message 1Q6DA or 1Q6FA. The class may be enabled by the PVARY DYNC command.

SUSPEND
The class is actually in the enabled state, but VSE/POWER has temporarily suspended this class for allocation because of resource shortage (see message 1Q3FI for detailed explanation). VSE/POWER enables the class automatically as soon as the resource is made available again. The class may be disabled by the PVARY DYNC command. Enabling by command has no effect, unless the PLOAD DYNC command is used again.

ACT/MAX
The maximum number of dynamic partitions ever allowed to be allocated for this class at one time versus the active or currently allocated number of dynamic partitions.

ALLOC
The total amount of virtual storage which every dynamic partition allocated for this class occupies. This includes both the partition allocation area and the Space Getvis area.

SIZE Defines the size of the program area within the partition allocation area for each dynamic partition of this class. This denotes the beginning of the partition Getvis area.

SP-GETV
Defines the size of the Space Getvis area within the ALLOC amount of

every dynamic partition of this class. The Space Getvis area is used as a work area for internal VSE services. For details on ALLOC, SIZE, or SP-GETV, refer to "Storage Management" in *z/VSE Guide to System Functions*.

PROFILE
Defines the .PROC member name of a IJSYSRS.SYSLIB library member which contains a user-defined sequence of job control statements. This sequence is executed automatically whenever any dynamic partition of this class is started, before execution of a VSE/POWER job in the partition.

LUBS Defines the maximum number of programmer logical units which may be used by any dynamic partition of this class.

Note: For additional display samples of the verified and loaded/active dynamic class table, see also "Dynamic Partition Support - Operation" on page 135.

Format 13: PDISPLAY EXIT

```
1 1R4AI  EXITTYPE STATE    NAME     WA-SIZE ADDRESS  EXITSIZE  WU
  1R4AI  JOBEXIT  ENABLED  RDREX1   00050   001F0200 01234     NP
  1R4AI  OUTEXIT  ENABLED  PRTEX5   00000   001E0000 00910     PA
  1R4AI  NETEXIT  ENABLED  NETEXIT1 00040   00100200 00502     NP
  1R4AI  XMTEXIT  DISABLED XMTEX1   00112   00120000 00300     PA
```

Figure 50. Display of Information on Loaded User Exit Routines

```
1 1R4AI  EXITTYPE STATE    NAME     WA-SIZE ADDRESS  EXITSIZE
```

The first line displays the header line and presents the display columns.

1R4AI The message number

EXITTYPE
Identifies the type of the exit routine

STATE
Displays the state of the exit and can be one of the following: enabled or disabled. If 'DISABLED' is indicated, the operator disabled the exit by means of the PVARY command

NAME
Displays the name of the exit

WA-SIZE
Displays the size of associated work area in bytes (decimal)

ADDRESS
Displays the location of exit in storage (hexadecimal)

EXITSIZE
Displays the size of the exit in bytes (decimal).

WU Displays the type of work unit for which the exit should be given control, either for

NP non parallel work unit, or

PA parallel work unit.

For details refer to "Passing Control to VSE/POWER User Exits" on page 130

Format 14: PDISPLAY AUSTMT

```
  1R48I  001 ***  BEGIN OF AUTOSTART INFORMATION  ***
1 1R48I  002 SET NORUN=YES
1 1R48I  003 SET SJECL=YES
1 1R48I  004 DEFINE L,CICSDATA,3F00,1,128,*
2 1R48I  005 FORMAT=NO
3 1R48I  006 PSTART BG,A0I
4 1R48I  007 READER=FEC
4 1R48I  008 PRINTERS=FEE,01E,02E,06E,05E,04E,03E,00E
4 1R48I  009 PUNCHES=FED,02D
  1R48I  010 PSTART F3
  1R48I      ..........
  1R48I      ..........
  1R48I  032 ***   END OF AUTOSTART INFORMATION   ***
```

Figure 51. Display of Information on Used Autostart Statements

Any autostart statement is displayed as is, regardless of the validity of the statement or if it has been processed. To make sure that it processed successfully, check the messages issued during VSE/POWER start up time, for example message 1Q13I or 1Q06I. The example is used to discuss possible failures:

```
1  1R48I  002 SET NORUN=YES
   1R48I  003 SET SJECL=YES
   1R48I  004 DEFINE L,CICSDATA,3F00,1,128,*
```

If SET or DEFINE statements were specified incorrectly, and were flagged each by message 1Q13I or 1Q09I, they are prefixed in the display by the string '>>ERROR:'.

If SET and DEFINE statements are placed **after** the FORMAT statement, they are ignored and message 1Q06I is displayed at startup time and they are prefixed in the display by the string ">>IGN'D:".

```
2  1R48I  005 FORMAT=NO
```

If the autostart statement 'FORMAT' was invalid or missing, the message '1Q11D FORMAT QUEUES=' has been displayed on the system console asking for an operator reply. When the reply was NO, the PDISPLAY AUSTMT records this event by:

```
1R48I nnn REPLY TO MSG '1Q11D FORMAT QUEUES=': NO
```

When the operator presses the 'ENTER' key, the reply event is recorded by:

```
1R48I nnn REPLY TO MSG '1Q11D FORMAT QUEUES=': (=NO AS DEFAULT)
```

```
3  1R48I  006 PSTART BG,A0I
```

After the PSTART statement, the statements READER, PRINTERS, PUNCHES must follow in the shown sequence. If they are omitted and other valid (or invalid) autostart statements follow, the others are not processed but messages 1R79I, 1R86I, and 1R50D are displayed on the system console.

```
4  1R48I  007 READER=FEC
   1R48I  008 PRINTERS=FEE,01E,02E,06E,05E,04E,03E,00E
   1R48I  009 PUNCHES=FED,02D
```

In order to make sure that all the specified devices for READER, PRINTERS, and PUNCHES have been successfully processed, you may use the PDISPLAY SPDEV command, which presents the currently defined spooled devices.

Format 15-1: PDISPLAY SPDEV

```
1R48I  001 ** BEGIN OF DISPLAYING SPOOLED DEVICES **
1R48I  002 PARTITION,DEV-CLASS: CUU,CUU,...
1R48I  003 BG,RDR: FEC
1R48I  004 BG,PRT: FEE,01E,02E,06E,05E,04E,03E,00E
1R48I  005 BG,PUN: FED,02D,00D,01D
1R48I  006 F3,RDR: 00C
1R48I      .......
1R48I      .......
1R48I  015 **   END OF DISPLAYING SPOOLED DEVICES **
```

Figure 52. Display of Information on Spooled Devices

The devices are displayed in the user-specified sequence, **not** in 'hexadecimal' order. This means, whenever a job executes in a certain partition and a * $$ LST or * $$ PUN statement is used without a specific LST= or PUN= operand, the * $$ LST/PUN statement applies to the **first** of the displayed 'PRT/PUN' devices of that partition. The following exceptional display conditions may appear:

For a Writer-Only Partition

```
1R48I  nnn F2,RDR: NONE (WRITER ONLY PARTITION)
```

if no device for READER has been specified, but writers.

For Reader-Only Partition

```
1R48I  nnn F2,PRT: NONE
1R48I  nnn F2,PUN: NONE
```

if no device for PRINTERS and PUNCHES has been specified, but a reader.

For a Static Partition not PSTART'ed under VSE/POWER

```
1R48I  PARTITION F1 NOT CONTROLLED BY VSE/POWER
```

if the command specifies a partition, which is not controlled by VSE/POWER, for example, D SPDEV,PART,F1.

For Static or Dynamic Partition not Allocated

```
1R48I  PARTITION(S) NOT ALLOCATED
```

if the command specifies a specific partition which is not allocated, for example:

```
D SPDEV,PART,FB
D SPDEV,PART,Y3
D SPDEV,PART,Y
D SPDEV,DPART
```

In the last example, at the moment the command is issued, no dynamic partition is allocated at all.

After Loading a New Dynamic Class Table

If a new dynamic class table has been re-loaded, those spooled devices of a dynamic partition are displayed which were defined by the dynamic class table being loaded at the time, the partition was started. Note, the spooled device definitions for the same class might be different in the new table.

Format 15-2: PDISPLAY SPDEVT

```
1R48I  001 ** BEGIN OF DISPLAYING SPOOLED DEVICE WITH DEVICE TYPE **
1R48I  002 PARTITION,DEV-CLASS: CUU(PUB-CODE,DEV-TYPE),CUU(...),...
1R48I  003 BG,RDR: FEC(11,2540R)
1R48I  004 BG,PRT: FEE(40,1403),01E(40,1403),02E(40,1403),
1R48I  005 BG,PRT: 06E(43,PRT1),05E(43,PRT1),04E(46,4248),
1R48I  006 BG,PRT: 03E(46,4248),00E(46,4248)
1R48I  007 BG,PUN: FED(21,2540P),02D(21,2540P),00D(21,2540P),
1R48I  008 BG,PUN: 01D(21,2540P)
1R48I  009 F3,RDR: 00C(11,2540R)
1R48I      .......
1R48I      .......
1R48I  022 **   END OF DISPLAYING SPOOLED DEVICE WITH DEVICE TYPE **
```

Figure 53. Display of Information on Spooled Devices including Device Types

```
1R48I  002 PARTITION,DEV-CLASS: CUU(PUB-CODE,DEV-TYPE),CUU(...),...
```

This line presents column descriptions to interpret the display lines. For possible inaccuracy of device type(s), take the following into account:

The displayed device type code may differ from the device type used in the ADD statement. This is because after the system has completed IPL, the device type code is saved in the internal VSE PUB table. This saved code is sometimes the same for different device types specified in the ADD statement. For example: Code 43 for PRT1 and 3211. In this case, the most likely device type is displayed, namely PRT1 for 43.

For a cross reference of device type code to actual device type, refer also to explanation of message 1Q41I.

Format 16-1: PDISPLAY TASKS

```
1R48I  ***   BEGIN OF DISPLAYING VSE/POWER TCB'S       ***
1R48I  TID ,CUU,TCBADR,T,PHASE(ADDR),REG12 ,STATE(DETAIL)
1R48I  LLDR,   ,459AE0, ,LD (4EA000),4EA304,C(R01=459B14)
1R48I  YTES,   ,452CC0, ,TV (4DC000),4DC0E0,C(R01=452CF4)
1R48I  O CP,   ,4512A0, ,CM (470000),470010,R(----------) (D TASKS)
1R48I  XMAS,   ,458820, ,XM (4CF900),4CFC32,M(R01=4589B8)
1R48I  XDEV,   ,45B600, ,XT (4D0200),4D039C,M(R01=45B798)
1R48I  XSAS,   ,45B920, ,XTS(4D6000),4D60D8,M(R01=45BAB8)
1R48I  NRV ,400,45B000, ,BS (4FB900),4FB9E6,B(NCB=45A000) (JEC3,CON)
1R48I  NTR1,400,45BC40, ,NC (4F7700),4F79F0,B(NCB=45A000) (JEC3,JOB)
1R48I  NTR1,400,45C000, ,NC (4F7700),4F79F0,B(NCB=45A000) (JEC3,OUT)
1R48I  DPST,   ,4571C0, ,DP (4B6200),4B62EE,M(R01=457020)
1R48I  WLST,01E,459000, ,LW (488700),489FC2,O(MSG=4DB0F9)
1R48I  E F3,00C,459540, ,XRE(491F00),492FB0,Q(R01=459560)
1R48I  E F2,009,458000, ,XRE(491F00),493AB0,S(R01=458034)
1R48I  RRDR,00C,4597C0, ,PR (483C00),483EFC,I(----------)
1R48I  I IT,   ,451020, ,T1 (46A700),46A75C,I(----------)
1R48I  ***      END OF DISPLAYING VSE/POWER TCB'S      ***
```

Figure 54. Display Information about ALL Active VSE/POWER Tasks

The display presents tasks according to their internal dispatching priority and shows the following per VSE/POWER task:

```
1R48I  TID ,CUU,TCBADR,T,PHASE(ADDR),REG12 ,STATE(DETAIL)
```

TID Task-id, four characters. For example: E F3 where E stands for 'execution task' and F3 for the partition-id.

CUU Channel-unit, three characters. For example: 00C for SYSIN. 400 for a BSC line or a CTCA. Three blanks if none is available.

TCBADR
Address of VSE/POWER's task control block (TCB)

T Stop code. For example:
S immediate stop
E stop at end-of-job
' ' (blank)
if task is not to be stopped

PHASE
Phase, three characters identifying a VSE/POWER phase to which register 12 (saved within the TCB) is pointing

(ADDR)
Start address of PHASE

REG12
Contents of register 12. This is the address from where the task would continue when its wait condition has been resolved.

STATE
Task state, one character identifying the task state. For example:
'C' for wait on ECB
'I' for inactive, wait for internal posting.

(DETAIL)
Shows detailed information about the state.

I,D state
DASHES (--------)

B state
Address of NCB for a PNET task, address of LCB for RJE/BSC tasks.

L state Register 3 (R03)- pointer to locked resource.

O state
Message address from TCMW, task is 'operator bound'.

R state
DASHES (--------) the current operating address cannot be determined, because the command was entered using the Attention Routine Interface.

All others
Register 1 (R01).

In addition, the following information might be displayed in parentheses, if applicable:

nodename
To identify the node of a PNET task, for example (JEC3)
CON To identify a console task for a PNET node
OUT To identify a task which transmits/receives output (LST/PUN entry)
JOB To identify a task which transmits/receives a job (RDR entry)

lu-name
To identify the lu-name of an RJE/SNA task

connect-id
To identify a task using the spool-access support, for example (00001)

devname
To identify the device name of a task using the external device support, for example (HUS).

command
To identify the currently processed operator command of the active permanent command processor task.

resource,LCK=lockword
To identify, for STATE=L, the 3-byte resource eyecatcher, followed by the contents of the resource lockword at offset X'1C' within the resource control block. Usually, the lockword identifies the address of the task (TCB) that currently owns the locked resource.

Notes:

1. A STATE of 'O' indicates this task is waiting for an operator reply. In this case, please check the messages issued by VSE/POWER and issue replies as many as required to allow VSE/POWER to continue.
2. If 009 is displayed for cuu, this partition has been started as a 'writer only partition'. In this case, a state of 'S' is displayed although no stop-condition for this partition has arisen.
3. A state of 'R' indicates this task is running. Only one task may be in the state 'R' and this should be a command processor task (task-id = 'O CP') at the moment this command is processed.
4. This layout may be changed due to the requirements of IBM.

Format 16-2: MSG F1,DATA=D TASKS

```
   1R48I  ***   BEGIN OF DISPLAYING VSE/POWER TCB'S       ***
 1 1R48I  ***   NO TASK ACTIVE, VSE/POWER IN WAIT STATE   ***
   1R48I  TID ,CUU,TCBADR,T,PHASE(ADDR),REG12 ,STATE(DETAIL)
   1R48I  YTES,   ,2BBA60, ,TV (33E600),33E6E0,C(R01=2BBA94)
   1R48I  JSPM,   ,2BD820, ,SM (33EA00),33EC08,Q(R01=1A00B8)
   1R48I  O CP,   ,2B12A0, ,CM (2CF000),2CF2FA,E(R01=2B12D4)
   1R48I  XMAS,   ,2BCCA0, ,XM (331B00),331E26,M(R01=2BCE38)
   1R48I  DPST,   ,2BC000, ,DP (317700),3177EE,M(R01=2BBD00)
   1R48I  I IT,   ,2B1020, ,T1 (2CA700),2CA75C,I(----------)
   1R48I  ***      END OF DISPLAYING VSE/POWER TCB'S      ***
```

Figure 55. ALL Active VSE/POWER Tasks, No Tasks Running, OC-exit Interface Request

```
   1R48I  ***   BEGIN OF DISPLAYING VSE/POWER TCB'S       ***
   1R48I  TID ,CUU,TCBADR,T,PHASE(ADDR),REG12 ,STATE(DETAIL)
   1R48I  YTES,   ,2BBA60, ,TV (33E600),33E6E0,C(R01=2BBA94)
   1R48I  JSPM,   ,2BD820, ,SM (33EA00),33EC08,Q(R01=1A00B8)
   1R48I  O CP,   ,2B12A0, ,CM (2CF000),2CF010,E(R01=2B12D4) (D A)
   1R48I  XMAS,   ,2BCCA0, ,XM (331B00),331E26,M(R01=2BCE38)
 2 1R48I  P PS,   ,2BC380, ,PS (306E00),306E10,R(PSW=306EE8)
   1R48I  DPST,   ,2BC000, ,DP (317700),3177EE,M(R01=2BBD00)
   1R48I  I IT,   ,2B1020, ,T1 (2CA700),2CA75C,I(----------)
   1R48I  ***      END OF DISPLAYING VSE/POWER TCB'S      ***
```

Figure 56. ALL Active VSE/POWER Tasks, One Task Running, OC-exit Interface Request

This command sample assumes VSE/POWER to reside in partition F1. When you pass the PDISPLAY TASKS command to the Operator Communication (OC)- exit of VSE/POWER, you get a snapshot of VSE/POWER tasks - all undisturbed by the PDISPLAY TASKS command itself. For Operator Communication support, refer also to “No VSE/POWER Command Accepted” on page 518. You may find the VSE/POWER partition either

(**1**) waiting, as displayed by the second headline, or

(**2**) active, as displayed by one ’R’-running task.

For an explanation of the information shown, refer to “Format 16-1: PDISPLAY TASKS” on page 286 (command passed through the standard Attention Routine Interface). In addition, for the ’R’-running task, this may be shown:

PHASE
identifies the VSE/POWER phase the task is currently operating in. If the task operates in a phase outside the VSE/POWER partition, the constant ’NO PHASE FD’ is displayed.

(DETAIL)
identifies the PSW-address of the running task when being interrupted for this OC-exit display.

PDRAIN: Deactivating (Draining) a Transmitter or Receiver Task

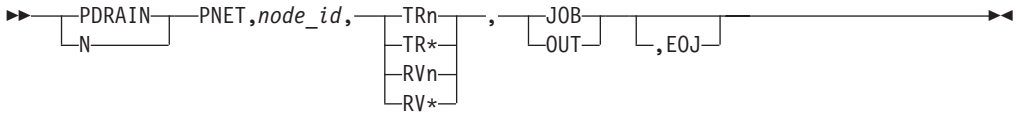
This command applies only if the networking function is initialized; it deactivates a transmitter or receiver task.

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



PNET
Code this operand as shown.

node_id
For *node_id*, specify the name of the node with which the affected receiver or transmitter task is communicating.

TRn | TR*
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 | RV*
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: Ending Operation Under Control of VSE/POWER

The command ends VSE/POWER operation after all jobs in dynamic and static partitions have come to end-of-job or after formatting (due to extending the data file: see "Extending the Data File During Warm Start" on page 38) has been completed. It causes all static partitions under control of VSE/POWER to be released and the VSE/POWER partition to be restored to normal VSE operation.

The processing statistics of the past VSE/POWER session are displayed on the operator console, as shown in Figure 7 on page 46.

All I/O 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 or IMM are 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 and 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.

The PEND IMM command stops immediately all activities of the VSE/POWER tasks. In contrast to the PEND command, queue entries are not processed until end-of-job, but terminated immediately. Just processed queue entries are kept or lost depending on their disposition and on the actions of the corresponding commands listed under the IMM operand.

In contrast to the PEND FORCE command, no dump is produced and the output so far produced by running partitions is kept and not lost. Using cuu as a second operand, a status report may be produced.

Once PEND or PEND cuu has been entered, PEND IMM still can be entered to expedite the shutdown of VSE/POWER. Once PEND or PEND IMM has been entered, PEND FORCE can still be entered.

Note: To avoid inadvertent VSE/POWER termination, you may request confirmation of the PEND command by providing the SET CONFIRM=PEND autostart statement. For further explanation, see "SET: Setting VSE/POWER Startup Control Values" on page 473.

Format of the Command

If you issue the command without an operand, VSE/POWER performs a normal shut-down operation with a statistics status report displayed on the operator console.

uraddr
indicates that you want VSE/POWER to provide an additional 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, see Figure 7 on page 46. Examples of queue displays are shown under "PDISPLAY Output Examples" on page 249.

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.

IMM

specifies that all activities are stopped immediately. Thus, the PEND IMM has the same effect as the following commands:

1. PFLUSH BG|Fn|dyn.-part.-id and PSTOP BG|Fn for all partitions. Unlike the normal PFLUSH partition command, a partition is flushed in any case which is without any prompting message (1QZ1D) if an IBM subsystem is running in the partition. No dumps are produced, the output produced so far is not lost but saved on spool, and additional JCL messages are added at the end to explain the termination.

   Note that a writer-only partition cannot be flushed and, therefore, this partition is not stopped until the end of the current running VSE/POWER job.
2. PSTOP uraddr
3. PSTOP tapeaddr
4. PSTOP lineaddr,FORCE
5. PSTOP RJE,SNA
6. PSTOP PNET,nodeid
7. PSTOP DEV,devname,FORCE
8. PSTOP SAS,ALL
9. PCANCEL STATUS to terminate the display initiated by the various types of the PDISPLAY command.
10. PSTOP TCPIP
11. PSTOP TCPSSL

In addition, PACCOUNT processing or formatting (due to extending the data file, see "Extending the Data File During Warm Start" on page 38) is canceled, which cannot be triggered by any other VSE/POWER command.

Note: If VTAM is running in a partition controlled by VSE/POWER, you should first stop VTAM. Otherwise, it may happen that VTAM cannot successfully stop all its connections. This may result, for example, in a hanging subtask in the VSE/POWER partition, which means VSE/POWER cannot be terminated.

FORCE

Specify FORCE if VSE/POWER operation is to end at once. VSE/POWER writes a formatted partition dump to the assigned dump sublibrary. When the SET 1Q30D=YES autostart statement has been specified, the operator is first prompted by message 1Q30D whether a dump is desired or not. Then all VSE/POWER controlled partitions are canceled, and currently active reader and writer tasks are terminated as for a PSTOP command.

Any subsequent VSE/POWER startup (or takeover of control by another shared system) will enter a full recovery of the queue file, so that potentially ″broken″ class chains are rebuilt and ″lost″ queue entries re-collected.

Examples of the PEND Command

PEND Normal end of VSE/POWER operation with a status report on the operator console.

PEND 00E
Normal end of VSE/POWER operation with a status report on the central operator console and an additional status report on output device 00E.

PEND FORCE
Abrupt end of VSE/POWER operation with prompting whether a dump is required.

PFLUSH: Canceling the Processing for a Queue Entry

The command causes the processing of a VSE/POWER queue entry by a VSE/POWER task to be canceled. The task then processes the next available VSE/POWER queue entry.

When *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 VSE/POWER job stops. If this job's disposition is:

D VSE/POWER deletes the entry, except if HOLD is specified. In this case, the entry is retained with a disposition of H.

K VSE/POWER retains the entry with its disposition changed to L.

For terminating VSE/POWER jobs that have established an ON $CANCEL condition, see "Interaction with z/VSE Conditional Job Control Language" on page 108.

When *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 this 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.

The formats of the PFLUSH command are as follows:

- Format 1: Flushing a Reader or Writer Task
- Format 2: Flushing the Processing of a Job
- Format 3: Flushing the Processing of Output Controlled by a Device Driving System
- Format 4: Flushing Network Receiving or Transmitting

Format 1: Flushing a Reader or Writer Task

Use format 1 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 cuu address of the I/O device associated with the reader or writer task which 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 cuu 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, POFFLOAD LOAD, or SELECT operation, the currently processed queue entry is terminated and not added to the queue. 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 requeued according to normal processing rules like for SAVE or BACKUP.

Format 2: Flushing the Processing of a Job

Use format 2 to cancel further processing of a VSE/POWER job in a VSE/POWER-controlled partition.

partition
The static or dynamic partition id of the partition which is to be cancelled. Thus, the VSE job running in the partition is canceled and no dump is taken, regardless of any dump options which may have been specified.

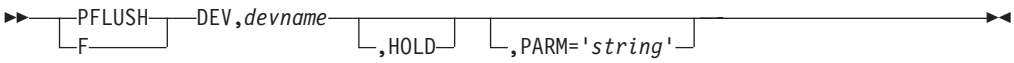
If an IBM subsystem (such as CICS or VSE/ICCF) is running in this partition, the operator is requested to confirm the PFLUSH command.

For terminating VSE/POWER jobs that have established an ON $CANCEL condition, see "Interaction with z/VSE Conditional Job Control Language" on page 108.

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).

The command is not accepted for a writer-only partition. When SLI is in process during flushing, message 1QC1I RC=0005 may be issued in addition.

Format 3: Flushing the Processing of Output Controlled by a Device Driving System



Use format 3 to indicate to a subsystem, for example PSF, 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.

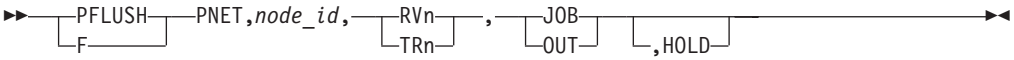
devname
For *devname*, specify the name of the output device to be used. This name must be the one by which the device-owning subsystem or application program knows the device.

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', 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 ('').

Format 4: Flushing Network Receiving or Transmitting



Use format 4 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.

node_id
For *node_id*, 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-queuing 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 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).

HOLD is meaningless with RVn.

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 queue entry.

PFLUSH BG

Terminates the VSE/POWER job that is processing 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 processing 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: Reactivating 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.

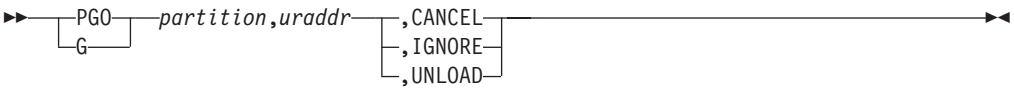
Notes:

1. Do not use PGO ...,CANCEL after the message 1Q28I. If the command is a PGO with a partition ID, 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).

The formats of the PGO command are as follows:

- Format 1: Reactivating a Tape Spooling Task in 1QB9A Wait State
- Format 2: Reactivating a Tape Reader or Offload Task in 1QB9A Wait State
- Format 3: Reactivating a Writer Task in 1Q40A Wait State
- Format 4: Reactivating a Writer Task in 1Q61A Wait State
- Format 5: Reactivating a Tape Spoling Task in 1Q55A Wait State
- Format 6: Reactivating a Tape Writer Task in 1Q55A Wait State
- Format 7: Reactivating a Device Controlled by a Device Driving System

Format 1: Reactivating a Tape Spooling Task in 1QB9A Wait State



This set of operands applies to a partition whose output is being spooled to tape.

partition
The static or dynamic partition id.

uraddr
The address of the 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 processing 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.

Format 2: Reactivating a Tape Reader or Offload Task in 1QB9A Wait State

This set of operands applies to the tape-SYSIN reader and the POFFLOAD functions of VSE/POWER:

tapeaddr
For *tapeaddr*, specify the cuu address of the tape drive used.

CANCEL
Specify CANCEL if the affected tape function is to be stopped immediately without any further action (see also Note 1 on page 300).

IGNORE
Specify IGNORE if VSE/POWER is to continue processing using the mounted POFFLOAD tape. When writing to tape, the tape labels are destroyed.

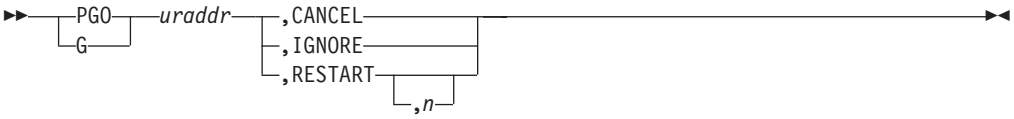
UNLOAD
Specify UNLOAD to have the affected tape unloaded. VSE/POWER requests the operator to mount a new, valid tape.

Format 3: Reactivating a Writer Task in 1Q40A Wait State

Use this command format to reactivate a writer task associated with an I/O device.

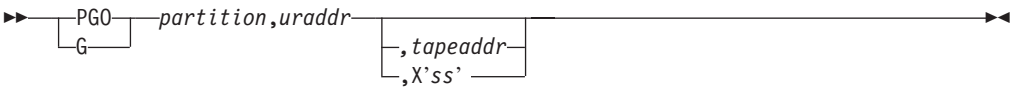
For *uraddr*, specify the cuu address of the I/O device whose associated writer task is to be reactivated.

Format 4: Reactivating a Writer Task in 1Q61A Wait State



Use this command format to reactivate a list/punch writer task that has encountered an unrecoverable I/O error. Specify the PGO command as prompted by message 1Q61A. For the meaning of the CANCEL, IGNORE, and RESTART operands, refer to the explanation of message 1Q61A.

Format 5: Reactivating a Tape Spooling Task in 1Q55A Wait State



This set of operands applies to a partition whose output is to be spooled to tape.

partition
The static or dynamic partition id as named in message 1Q55A.

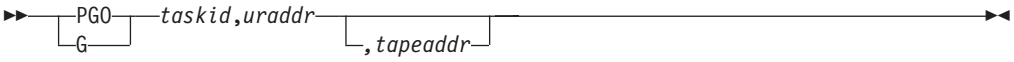
uraddr
For *uraddr*, specify the cuu address of the spooled writer device, with which VSE/POWER spools to output to tape.

tapeaddr
For *tapeaddr*, specify the cuu address of the tape drive to be used for tape spooling.

Note: If *tapeaddr* is not specified, VSE/POWER switches to disk spooling and keeps the output with DISP=H. See also the explanation of message 1Q55A.

X'ss'
Specify the desired recording density(mode) of the tape drive. A list of valid mode specifications can be found in the discussion of the ASSGN statement in *z/VSE System Control Statements*.

Format 6: Reactivating a Tape WriterTask in 1Q55A Wait State



This set of operands applies to a tape writer task that obtains its input from a VSE/POWER spool tape or POFFLOAD tape.

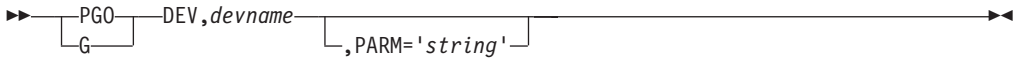
taskid
The writer task named in message 1Q55A, either LST or PUN.

uraddr
For *uraddr*, specify the cuu address of the device named in message 1Q55A.

tapeaddr
For *tapeaddr*, specify the cuu address of the tape drive (to be used as input for the writer task), on which the VSE/POWER spool tape or POFFLOAD tape is mounted.

Note: If *tapeaddr* is not specified VSE/POWER stops the tape writer task. See also the explanation of message 1Q55A.

Format 7: Reactivating a Device Controlled by a Device Driving System



Use this set of operands to tell a subsystem, for example PSF, 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', 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: Placing Entries of Physical Queues into the Hold or Leave State

The command puts one or more VSE/POWER entries of a queue into the hold or leave state. It causes an entry with a disposition of:
- D (dispatchable) − to be changed to disposition H (hold).
- K (keep) − to be changed to disposition L (leave).

Note: A central operator PHOLD command is confirmed on the console by

```
1R88I  OK : 6 ENTRIES PROCESSED BY PHOLD,LST,*SSL
```

for example, thereby indicating the number (6) of held queue entries together with the corresponding operator command.

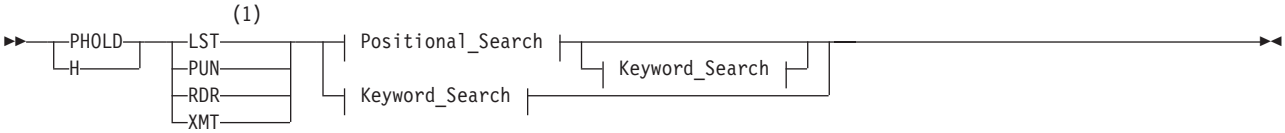
VSE/POWER ignores the command if it is given for a queue entry with a disposition other than D or K. It informs the operator about this by a message.

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 terminal, 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.

For VSE/POWER queues addressable by the PHOLD command, refer to Table 9 on page 225.

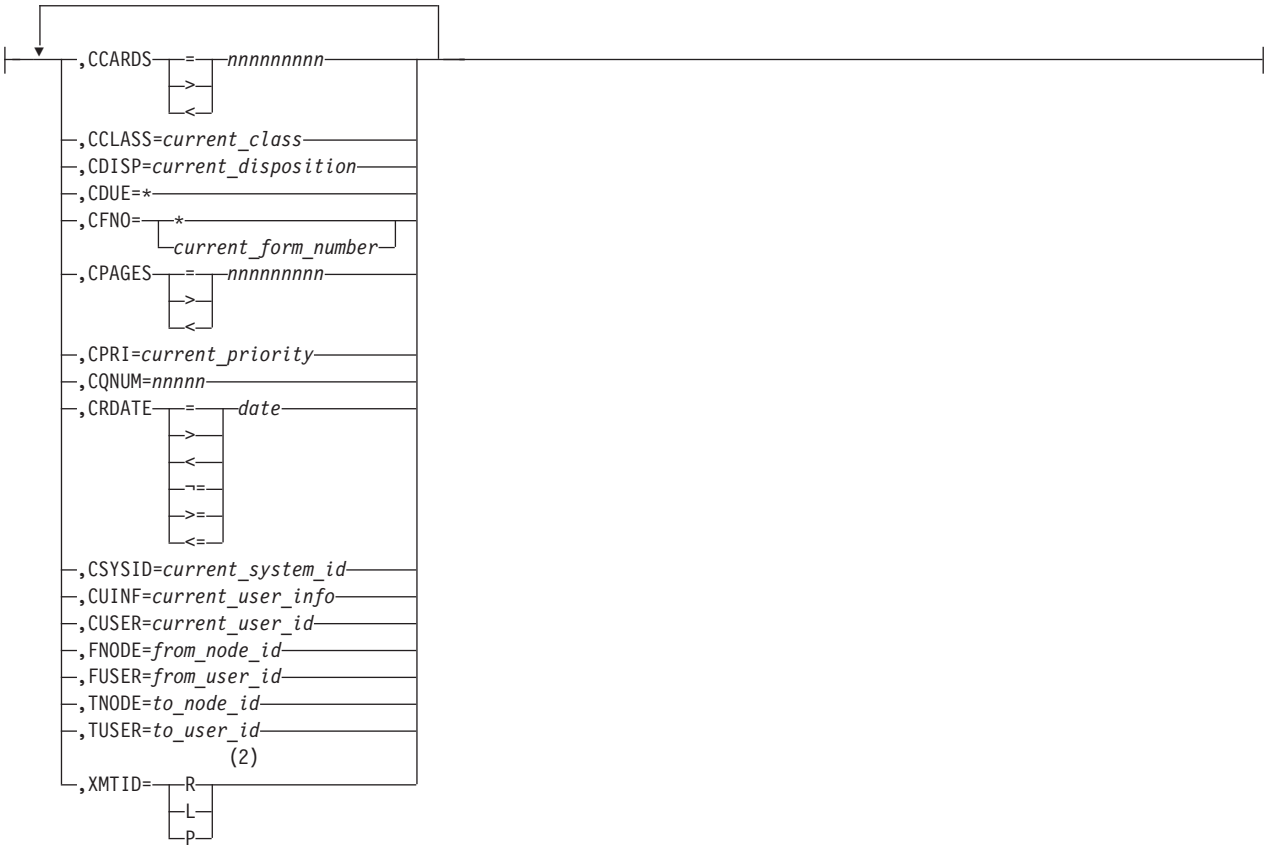
VSE/POWER accepts a PHOLD command with up to 14 operands.

Format of the Command



Positional_Search:

Keyword_Search:



Notes:

1 RDR/LST/PUN/XMT queues are called physical queues in contrast to the logical CRE/DEL queues.

2 XMTID is meaningful only if XMT is specified.

Specify the queue for which the command is to be executed, in the form:

LST
to address the list queue.

PUN
to address the punch queue.

RDR
to address the reader queue.

XMT
to address the transmission queue.

Positional Search Operands

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 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
specifies the 1 to 8 character job name by which the entry is known to VSE/POWER.

Specifying PHOLD with 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.

jobnumber
specifies the number assigned to the job by VSE/POWER. 1 to 5 digits may be specified. 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 (specified as 1 to 3 digits) only together with the VSE/POWER assigned job number. To get the correct job suffix S=nnn, issue a PDISPLAY command for the output queue entry.

jobn*
Specifies those queue entries whose name begin with the specified characters. As search argument, you can specify for *jobn* one to eight characters.

Keyword Search Operands

CCARDS=|<|>nnnnnnnnn
indicates that only reader and punch queue entries are to be addressed whose current total card count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified card nnnnnnnnn value, which must be greater than zero. A card count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P). When CCARDS is specified for the LST queue or for list type entries of the XMT queue (I=L), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

   ```
   1R52I LIST QUEUE NOTHING TO HOLD or
   1R52I NOTHING TO HOLD
   ```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For reader or punch type entries in the XMT queue the 'LINES' column actually presents statements, that means cards. This is the value against the CCARDS operand is compared.
4. For active entries (DISP=*) in the PUN or XMT queue, 'cards left to be processed' are displayed, although the entries were selected according to their total card count, which is not displayed then.

CCLASS=current_class
requests an affected queue entry (or entries) to be placed into the hold or leave state if the entry's current class matches the specified class. This operand will override a possible specification of the positional 'class' operand. Specification of either 'class' or the 'CCLASS=' operand will expedite the access path to the VSE/POWER queue file.

CDISP=current_disposition
requests an affected queue entry (or entries) to be placed into the hold or leave state if the entry's current disposition matches the specified disposition.

CDUE=*
indicates that all jobs are to be held for which time event scheduling operands have been specified. Such jobs can either be in the RDR or XMT queue.

CFNO=*|current_form_number
requests an affected queue entry (or entries) 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. One can specify '*' to process those queue entries which have no form number, or the four-character form number of the queue entries to be processed.

CPAGES=|<|>nnnnnnnnn
indicates that only list queue entries are to be addressed whose current total page count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified page nnnnnnnnn value, which must be greater than zero. A page count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the LST queue or for list type entries of the XMT queue (I=L). When CPAGES is specified for the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

```
1R52I READER QUEUE NOTHING TO HOLD or
1R52I NOTHING TO HOLD
```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For list type entries in the XMT queue the standard display presents 'LINES'. Use the 'P=n...n' page count of the FULL=YES display, when you select entries by the CPAGES operand.
4. For active entries (DISP=*) in the LST or XMT queue, 'pages left to be processed' are displayed, although the entries were selected according to their total page count, which is not displayed then.

CPRI=current_priority
requests an affected queue entry (or entries) to be placed into the hold or leave state if the priority currently assigned to the entry matches the specified priority.

CQNUM=nnnnn
indicates that the requested change is to be done if the queue entry's internal queue number matches the specified `nnnnn` value.

This unique number assigned by VSE/POWER can be made visible by the PDISPLAY command QNUM=nnnnn display field. See also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257. Use the CQNUM search operand if more than one queue entry has the same jobname and jobnumber and the entries otherwise cannot be distinguished by other attributes for command selection. You can specify 1 to 5 digits.

Note: Only numbers within the range indicated by "TOTAL NUMBER OF QUEUE RECORDS" (see Figure 7 on page 46) are accepted for searching. Only one queue entry can satisfy the search criterion.

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 in relation to the specified date:

```
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

```
1) mm/dd/yy or dd/mm/yy
2) mm/dd/yyyy or dd/mm/yyyy
```

Do not specify mm greater than 12, dd greater than 31, and yy or yyyy greater than the current year.

Notes:

1. If you specify a 2 digit yy-year, VSE/POWER expands it for internal processing and comparison to a 4 digit year according to the 'fix-88-window' rule:

```
if yy is greater than (>) 88, then 19yy
if yy is less than or equal (<=) 88, then 20yy
```

2. The format of the system date must not be changed after VSE/POWER has been initialized or between warm starts of VSE/POWER (see "Functions Not Expecting Date Format Changes" on page 128).
3. In a shared spooling environment, all systems must have the same date format.

CSYSID=current_system_id
requests an affected queue entry to be placed into the hold or leave state if the system ID currently assigned to the entry matches the specified ID.

CUINF=current_user_info
requests an affected queue entry to be placed in the hold or leave state if the queue entry's user information matches the specified current_user_info value. User information is provided to a queue entry by the * $$ JOB/LST/PUN JECL statements or by the PALTER command, and can be made visible by a PDISPLAY ...,FULL=YES request, which presents this information left-justified with trailing blanks, all enclosed in quotes via the U='...' display field (see also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257). When user information contains

1. neither embedded blanks nor commas and is displayed, for example, as

   U='PETER ', it can be searched for using
   CUINF=PETER, or
   CUINF='PETER' or
   CUINF='PETER '
2. an embedded blank (or comma) and is displayed, for example, as

   U='PETER BOY ', it can be searched for using
   CUINF='PETER BOY ', or
   CUINF='PETER BOY'
3. an embedded blank (or comma) and a single quote, shown for example as

   U='PETER'S BOOK ', it can be searched for using
   CUINF='PETER''S BOOK'

The CUINF specification (when enclosed in single quotes) can contain trailing blanks up to the field limit of 16 characters.

CUSER=current_user_id
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 ID.

FNODE=from_node_id
requests an affected queue entry to be placed into the hold or leave state if the entry's "from node" name matches the node name specified by *from_node_id*. However, VSE/POWER does not verify that the specified node name is defined in the network-definition table.

FUSER=from_user_id
requests an affected queue entry to be placed into the hold or leave state if the entry's "from user" ID matches the user ID specified by *from_user_id*.

Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user ID of LOCAL; it does not cause the status of local-user originated queue entries to be changed.

TNODE=to_node_id
requests an affected queue entry to be placed into the hold or leave state if the

entry's "to node" name matches the node name specified by *to_node_id*. This operand is valid only if your specification for queue is XMT.

TUSER=to_user_id
requests an affected queue entry to be placed into the hold or leave state if the entry's "to user" ID matches the user ID specified by *to_user_id*.

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 ID of LOCAL; it does not cause a status change of entries destined for local users.

XMTID=R|L|P
This operand is effective only if your command applies to the XMT queue; for other queue types the operand is ignored. For XMT queue, you can address one of the three transmission subqueues presented as R=Reader, L=List, P=Punch by the 'I' identification column of a PDISPLAY XMT command display line.

The operand indicates that the entry is to be placed into hold or leave state if the queue entry's 'I'-identification matches the XMTID value given in this command. For a PDISPLAY XMT example, see page 261.

Examples of the PHOLD Command

PHOLD RDR,ALL
Changes to H or L the disposition of those entries in the reader queue which have a disposition of D or K, respectively.

H LST,C
Changes to H or L the disposition of entries in the list queue with job characteristics as follows:

- A disposition of D or K.
- Output class C.

H LST,CRDATE<05/15/05
Changes to H or L the disposition of those entries in the list queue that were created before May 15, 2005. Or for 4-digit years:

H LST,CRDATE<05/15/2005
Changes to H or L the disposition of those entries in the list queue that were created before May 15, 2005.

H XMT,TNODE=NODEB
Changes to H or L the disposition of those entries in the transmission queue which have a disposition of D or K, and are to be routed to the node NODEB.

PINQUIRE: Requesting Remote Status Information

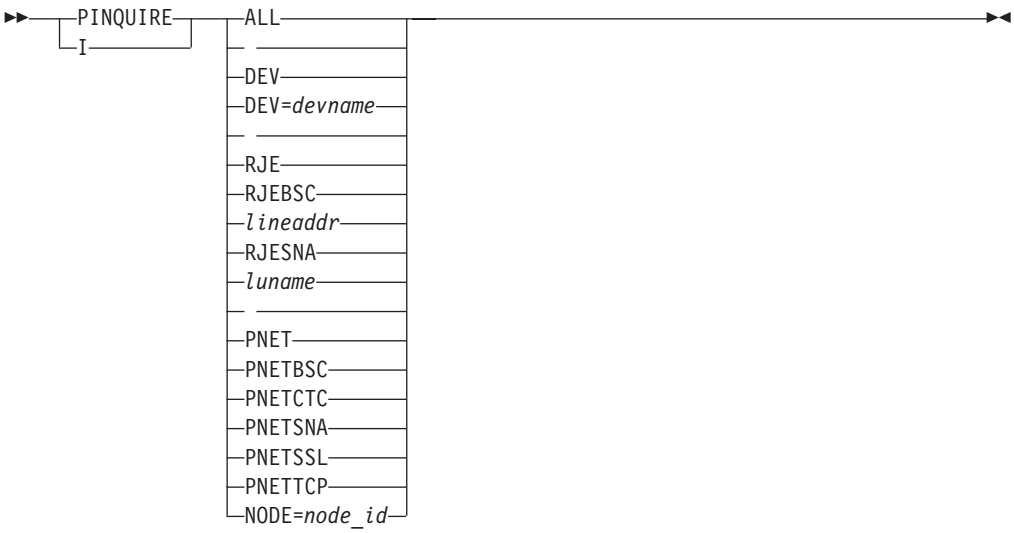
The command can be used to request status information about one or all of the following:

- An external output device
- A BSC line
- A PNET remote node
- The PNET local node and interface to TCP/IP
- An RJE/SNA logical unit

Note: If VSE/POWER runs on a virtual machine under VM and this machine happens to enter CP mode, you cannot use the command's short form (I). This would cause the VM control program (CP) to attempt IPL.

A discussion of VSE/POWER's display in response to the command is given under "Examples of Status Information Displayed" on page 313.

Format of the Command



ALL
to display status information about all telecommunication and device driving connections.

DEV
to display status information about all device driving connections.

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.

RJE
to display status information on all RJE (both BSC and SNA) telecommunication connections.

RJEBSC
to display status information on all RJE/BSC connections.

lineaddr
For *lineaddr*, supply the address of the BSC line used with RJE/BSC (or even PNET/BSC), for which status information is requested. Specify the address in the form `cuu`.

RJESNA
to display status information on all RJE/SNA connections.

luname
For *luname*, give the name of the RJE/SNA logical unit for which the status information is desired.

PNET
to display status information on all VSE/POWER NJE networking connections to any other node.

PNETBSC
to display status information on all nodes linked via a PNET/BSC connection.

PNETCTC
to display status information on all nodes linked via a PNET/CTC connection.

PNETSNA
to display status information on all nodes linked via a PNET/SNA connection.

PNETSSL
to display status information on all nodes linked via a PNET/SSL connection.

PNETTCP
to display status information on all nodes linked via a PNET/TCP connection.

NODE=node_id
For *node_id*, specify the name of the networking node for which status information is desired:

- any non-local (remote) node, or
- the local node, to get a summary display of
  - the number of PSTARTed nodes (per connection type)
  - the state of the interface of VSE/POWER to the TCP/IP partition for standard communication
  - the state of the interface of VSE/POWER to the TCP/IP partition for Secure Sockets Layer (SSL) communication

  For details, see also “Example 5: I NODE=own-node” on page 317.

Examples of Status Information Displayed

VSE/POWER displays status information in response to a PINQUIRE command in the form of messages.

Example 1: PINQUIRE ALL

```
1R56I  030  PROCESSING 22
1R56I  031  PROCESSING 150
1R56I  032  NOT INITIATED
 ... ... ...
1R56I  036  INACTIVE
1R56I  056  PROCESSING NODE=NODEA BSIZE=01000 BSC
1R56I  RT74 LOGGED ON R025,1024/1024=IN/OUT RUS1ZE
1R78I  DEVICE PLOT1 ACTIVE
1R78I  DEVICE PLOT2 SETUP IN PROGRESS
1R56I  TCP  NODE=NODEB SESSION PENDING IPADDR=009.164.155.068
1R56I  SNA  NODE=NODEC SESSION PENDING APPLID=MYCAPP
```

Figure 57. Display of Status Information Issuing PINQUIRE ALL

There is one line of status information for every resource whose status is displayed. The line is either message 1R56I (related to remote job entry or networking resources) or message 1R78I (related to external device control resources).

Every display line includes a status descriptor. A list of possible status descriptors follows. The list gives a short explanation of these descriptors.

Status Descriptor
Explanation for Message 1R56I

INACTIVE
The indicated BSC line has been started by the 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 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 ID.

- For RJE,SNA: The session is logged on. A processor is either sending or receiving.

For networking: a session is in progress with the named node, negotiated buffer size, and identified connection type

- CTC or BSC
- IPADDR=www.xxx.yyy.zzz for TCP or SSL
- APPLID=xxxxx for SNA

If you inquire about a specific node, 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 the named node.

Status Descriptor
Explanation for 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 sub- system control.

Example 2: I NODE=NODEA

```
1R56I  056  PROCESSING NODE=NODEA BSIZE=01000 BSC
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
```

Figure 58. Display of Status Information Issuing I NODE=NODEA For Non-Local Node

VSE/POWER displays, as a header message, the one it displays for the node in response to a PINQUIRE ALL or PNET 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.

Example 3: I DEV=PLOT1

```
   1R78I  DEVICE PLOT1 ACTIVE
1  1R78I  CLASSES: ABCD - STATUS: HALTING
2  1R78I  DESTINATION: DEPT3244 DEPT3245
```

Figure 59. Display of I DEV=PLOT1 (Active)

```
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 IDs (DEPT3244 and DEPT3245) of the users
to which the output is dedicated.
```

Example 4: I DEV=PLOT1

```
1 1R78I  DEVICE PLOT1 WAITING FOR WORK
2 1R78I  CLASSES: ABCD - QUEUE:  L
  1R78I  DESTINATION: DEPT3244 DEPT3245
```

Figure 60. Display of I DEV=PLOT1 (Waiting for Work)

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 (could also be P for PUN queue).

Example 5: I NODE=own-node

```
1 1R56I  NO. STARTED NODES: BSC=000, CTC=000, SNA=000, TCP=001, SSL=001
2 1R56I  TCP/IP: DRIVER SUBTASK ATTACHED
2 1R56I  TCP/IP: INTERFACE TO IP PARTITION AVAILABLE
2 1R56I  TCP/IP: IPADDR=009.164.155.068 IPPORT=00175
3 1R56I  TCP SSL: DRIVER SUBTASK ATTACHED
3 1R56I  TCP SSL: INTERFACE TO IP PARTITION AVAILABLE
3 1R56I  TCP SSL: IPADDR=009.164.155.068 SPORT=02252
4 1R56I  CIPHERS: STRONG=X'0A62' NORMAL='09' WEAK='080102'
```

Figure 61. Display of I NODE=own-node

```
1 1R56I  NO. STARTED NODES: BSC=000, CTC=000, SNA=000, TCP=001, SSL=001
```

Number of PSTARTed remote nodes per connection type.

```
2 1R56I  TCP/IP: DRIVER SUBTASK ATTACHED
2 1R56I  TCP/IP: INTERFACE TO IP PARTITION AVAILABLE
2 1R56I  TCP/IP: IPADDR=009.164.155.068 IPPORT=00175
```

Status of VSE/POWER's interface (represented by the TD Subtask) to the TCP/IP for VSE/ESA partition for standard communication, enabled as soon as the loaded Network Definition Table (NDT) contains at least one TCP node, otherwise "SUBTASK NOT ATTACHED". IPADDR denotes the dynamically acquired TCP/IP dotted decimal address of the TCP/IP for VSE/ESA partition. IPPORT denotes the port (for standard communication) on which the TD Subtask listens for incoming CONNECT requests.

```
3 1R56I  TCP SSL: DRIVER SUBTASK ATTACHED
3 1R56I  TCP SSL: INTERFACE TO IP PARTITION AVAILABLE
3 1R56I  TCP SSL: IPADDR=009.164.155.068 SPORT=02252
```

Status of VSE/POWER's interface (represented by the SD Subtask) to the TCP/IP for VSE/ESA partition for secure sockets layer (SSL) communication, enabled as soon as the loaded Network Definition Table (NDT) contains at last one SSL node, otherwise "SUBTASK NOT ATTACHED". SPORT denotes the secure port (for SSL communication) on which the SD Subtask listens for incoming CONNECT requests.

```
4 1R56I  CIPHERS: STRONG=X'0A62' NORMAL='09' WEAK='080102
```

The command displays the ciphers which are used for the corresponding encryption.

Example 6: I NODE=remote-node with SSL

```
   1R56I  SSL PROCESSING NODE=POWER263 BSIZE=32000 IPADDR=009.164.155.138
1  1R56I  LOCAL NODE IS ACTING AS CLIENT,ENCRYPT=NORMAL,CIPHER=X'09'
   1R56I  JOB-TRANSMITTER 1=I 2=D 3=D 4=D 5=D 6=D 7=D
   1R56I  OUT-TRANSMITTER 1=A 2=D 3=D 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=I 2=I 3=I 4=I 5=I 6=I 7=I
```

Figure 62. Display of I NODE=remote-node with SSL(as client)

or

```
   1R56I  SSL PROCESSING NODE=POWER263 BSIZE=32000 IPADDR=009.164.155.138
1  1R56I  LOCAL NODE IS ACTING AS SERVER,ENCRYPT=NORMAL,CIPHER=X'09'
   1R56I  JOB-TRANSMITTER 1=I 2=D 3=D 4=D 5=D 6=D 7=D
   1R56I  OUT-TRANSMITTER 1=A 2=D 3=D 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=I 2=I 3=I 4=I 5=I 6=I 7=I
```

Figure 63. Display of I NODE=remote-node with SSL(as server)

```
1  1R56I  LOCAL NODE IS ACTING AS CLIENT|SERVER,ENCRYPT=NORMAL,CIPHER=X'09'
```

The command displays the used cipher (*ENCRYPT=WEAK|NORMAL|STRONG*) and the encryption value (*CIPHER=),*

PLOAD: Loading an NDT, an Exit Routine, or a Dynamic Class Table

The formats of the PLOAD command are as follows:

- Format 1: Loading an NDT
- Format 2: Loading an Exit Routine
- Format 3: Loading a Dynamic Class Table

Loading an NDT (Format 1)

The PNET parameter of this command applies only to the networking function. It dynamically loads a new network definition table (NDT) while VSE/POWER is 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.

   Changing the node name can only be requested during a VSE/POWER warm start via an NDT named in the PNET= operand of the POWER macro generation, or named by the SET PNET autostart statement.
2. When an NDT is loaded that contains at least one TCP node, an attempt is made to attach the TCP/IP TD subtask in the VSE/POWER partition (unless already activated) in order to be prepared to accept TCP/IP connect requests from other TCP NJE nodes. The TD subtask is terminated during VSE/POWER PEND processing when all active TCP nodes have been stopped. For premature termination of the TCP/IP interface, see the PSTOP command "Format 13: Stopping the TCP/IP Interface Before PEND" on page 387.
3. When an NDT is loaded that contains at least one SSL node, an attempt is made to attach the TCP/IP SSL SD subtask in the VSE/POWER partition (unless already activated) in order to be prepared to accept TCP/IP Secure Sockets Layer connect requests from other TCP NJE nodes. The SD subtask is terminated during VSE/POWER PEND processing when all active SSL nodes have been stopped. For premature termination of the TCP/IP SSL interface, see the PSTOP command "Format 14: Stopping the TCP/IP SSL Interface Before PEND" on page 388.
4. If you issue too many PLOAD commands for large tables and exits, you may run out of Getvis space and have to restart VSE/POWER.

Loading an Exit Routine (Format 2)

The exit parameters (JOBEXIT, OUTEXIT, NETEXIT, XMTEXIT) load new exit routines. A just loaded routine always is enabled. If a new exit routine is loaded while another exit routine of the same type currently is enabled, the new routine takes effect at once. This means that the next record is passed to the new exit routine.

VSE/POWER does **not** change the work area size for entries currently printed, punched, received or transmitted via the network or accessed by a device service

task (DST). The new work area size takes effect when starting a new queue entry. For a logical reader task (spooling input to disk), the new work area takes only effect when the logical reader task is started again by the PSTART command. Therefore, loading of exit routines dynamically only should be done when the task which should use the exit is not running.

For safety reasons, exit routines should verify at entry that the passed workarea size (passed in the first fullword) is sufficient. For a sample verification code refer to OUTEXIT and JOBEXIT routines in *VSE/POWER Application Programming*, or to the receiver and transmitter exit example in *VSE/POWER Networking*.

An exit routine loaded by the PLOAD command is stored into the Getvis area of the VSE/POWER partition. If a new routine is loaded the Getvis area occupied by the current one never can be released since other tasks could currently use this exit. This means that if the operator frequently issues the PLOAD command, the available Getvis space may be exhausted.

Note: If you try to load an exit phase which currently resides in the SVA, VSE/POWER issues message 1Q15I and disables a corresponding previously loaded exit.

Loading a Dynamic Class Table (Format 3)

The DYNC parameter of this command applies to dynamic partition scheduling only. It loads the z/VSE library member DTR$DYN*x*.Z into the VSE/POWER area for verification and optionally creates an active Dynamic Class Table with classes enabled for dynamic scheduling of jobs.

All command combinations of this format are terminated by a status display of the addressed Dynamic Class Table.

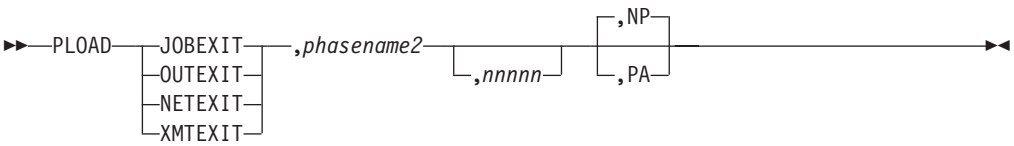
For further information refer to “Activate Dynamic Partition Support” on page 135.

Format 1: Loading an NDT

PNET
Code this operand as shown.

phasename1
Specify the name of the NDT 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.

Format 2: Loading an Exit Routine



JOBEXIT
Specify this operand to load a new job/reader exit.

OUTEXIT
Specify this operand to load a new output exit.

NETEXIT
Specify this operand to load a new PNET receiver exit.

XMTEXIT
Specify this operand to load a new PNET transmitter exit.

phasename2
Specify the name of a user-written exit routine to be loaded into the Getvis area of the VSE/POWER partition.

nnnnn
For *nnnnn* specify the number of bytes VSE/POWER reserves as work area for every task that enters the exit. If you omit nnnnn, no work area is reserved. The maximum is 65,535 bytes. The work area is initialized with X'00'. The first four bytes of the work area contain its length. The length is refreshed every time before passing control to the exit routine. The work area is obtained from the Getvis area of the VSE/POWER partition.

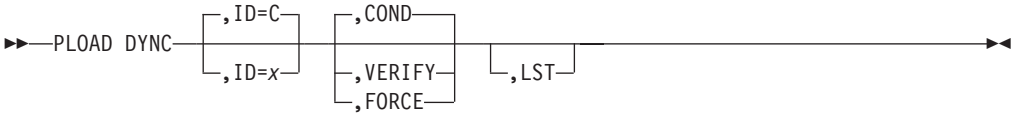
NP|PA
Specify the type of workunit for which the exit should be given control:

NP as non-parallel workunit (default)

PA as parallel workunit. The PA option takes effect only when the multiprocessor support of VSE/POWER has been enabled.

For details on processing workunits refer to "Passing Control to VSE/POWER User Exits" on page 130.

Format 3: Loading a Dynamic Class Table



DYNC
Specify this operand to check and/or activate the Dynamic Class Table.

ID=C|*x*
Specify this operand to address a specific member DTR$DYN*x*.Z of the z/VSE IJSYSRS.SYSLIB for loading. *x* may be a character from 0-9 or A-Z. If this operand is omitted, the default member DTR$DYNC.Z will be loaded.

COND
This operand activates your z/VSE library member DTR$DYN*x*.Z as the active Dynamic Class Table conditionally, provided that all dynamic classes are specified correctly. For further information, please refer to "Loading a Dynamic Class Table Conditionally" on page 135.

VERIFY
Specify this operand to load your z/VSE library member DTR$DYN*x*.Z for the verification of dynamic class characteristics only, without creating an active Dynamic Class Table. For further information refer to "Loading a Dynamic Class Table for Verification" on page 135.

FORCE
Specify this operand to activate your z/VSE library member DTR$DYN*x*.Z as the active Dynamic Class Table unconditionally, even if invalid classes were found. For further information, see "Loading a Dynamic Class Table Unconditionally" on page 136.

LST
Specify this operand if all status display messages of the Dynamic Class Table should not be displayed on the console, but should be collected in a list queue entry with the following attributes:

- $DYDxxxx as job name, where xxxx are the last four digits of the VSE/POWER-assigned job number.
- Priority 9
- Disposition H
- Class A

Message 1Q6BI additionally appears on the console to show the result of the PLOAD DYNC command. It is followed by message 1Q6BI (message version 5) which identifies the name of the created list queue entry.

Examples 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.

PLOAD OUTEXIT,OWNEX,600
Loads the user output exit routine OWNEX and causes a work area of 600 bytes to be given to every output task which gives control to OWNEX.

PLOAD DYNC,VERIFY,LST
Loads z/VSE library member DTR$DYNC.Z into the VSE/POWER area and checks all dynamic class specifications for correctness. The verification results are collected in the list queue entry named '$DYDxxxx', where 'xxxx' is shown in message 1Q6BI.

PLOAD DYNC
Loads z/VSE library member DTR$DYNC.Z into the VSE/POWER area, checks all dynamic classes and, if all classes are valid, activates your member as the Dynamic Class Table with all classes enabled (provided that they do not specify 'initially disabled').

PLOAD DYNC,ID=2,FORCE
Loads the z/VSE IJSYSRS.SYSLIB library member DTR$DYN2.Z, checks and verifies the member, and activates it as the Dynamic Class Table under all circumstances, even if the verification step finds an invalid class.

POFFLOAD: Saving Queue Entries on Tape and Restoring

The command enables you to save or restore queue entries of the various queues, using a tape as intermediate storage. For further information on this command, please refer to “Tape Processing with VSE/POWER” on page 151.

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 BACKUP/PICKUP/SAVE function can be transferred from one VSE/POWER-controlled system to another that has the same or higher version/release level.

A tape created with the POFFLOAD BACKUPxx/PICKUPxx/SAVExx function can be transferred from the current VSE/POWER-controlled system to the corresponding lower xx-version/release level VSE/POWER system. According to “Queue Entries on Spool Tape” on page 528, lower level systems can only interpret queue records on tape with their native queue record length, but no greater length.

Output queue entries saved on tape may be printed or punched directly from that tape using PSTART Format 2 “Format 2: Processing Output Stored on Tape” on page 364.

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 every queue entry on the tape.

Notes:

1. For handling of the end-of-volume condition in case of BACKUP, PICKUP, or SAVE for very large entries, refer to the SAVE function under “Processing for the Offload-Queue Function” on page 155. If processing unlabeled tape, your tape reel should be big enough to hold even your largest queue entry completely.
2. POFFLOAD does not check on the validity of the queue records or data blocks when saving or backing them up to tape. However, when loading them back from tape, validity checking is done for all data records contained in the data blocks (DBLKs) and message *1Q6MI* is issued at failure.
3. When requesting for example PDISPLAY LST for queue entries being processed, (DISP=*) by POFFLOAD, PICKUP, or SAVE, only ’total’ number of pages are displayed. For ’remaining’ number of pages, refer to “Format 1-3: PDISPLAY LST” on page 253.

The functional differences of the three save-to-tape functions are listed in detail in the description of each BACKUP/PICKUP/SAVE operand. The following figure helps you select the function for your need:

Table 10. Characteristics of BACKUP, SAVE, and PICKUP Functions of the VSE/POWER POFFLOAD Command

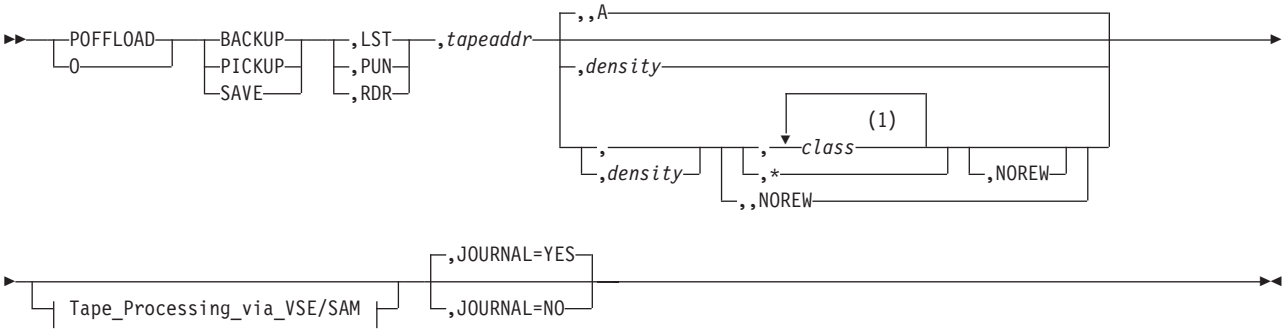
Description	BACKUP	SAVE	PICKUP
Scope of Locking:	Locks entire system while running	Locks individual spool entry only	Same as SAVE function
Type of offload:	Snapshot of spool queue(s)	No snapshot; unlocked system permits spool queue changes	Same as SAVE function
Selection of Spool Entries:	Saves all entries regardless of disposition	Saves dispatchables only: • no active entries (DISP=*) • no non-dispatchables • no time-event scheduling jobs	Same as BACKUP function
Disposition Handling:	No change of entry	Removes entries with DISP=D. Changes DISP=K to DISP=L.	Same as BACKUP function
Parallelism:	• Only one BACKUP allowed, including shared systems • BACKUP and PICKUP are mutually exclusive	Multiple SAVE tasks can run.	Same as BACKUP function

The term *snapshot* means a true point-in-time copy of the spool queue.

The formats of the POFFLOAD command are as follows:

- Format 1: Backup, Pickup, or Save of a Selected Local Queue
- Format 2: Backup, Pickup, or Save of the Transmission Queue or All Queues
- Format 3: Loading All or Selected Entries from Tape
- Format 4: Backup, Pickup, or Save for a Previous Release of VSE/POWER

Format 1: Backup, Pickup, or Save of a Selected Local Queue

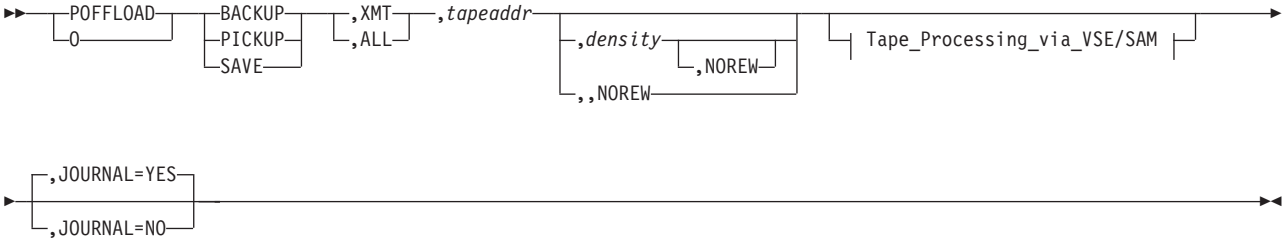


Tape_Processing_via_VSE/SAM:

Notes:

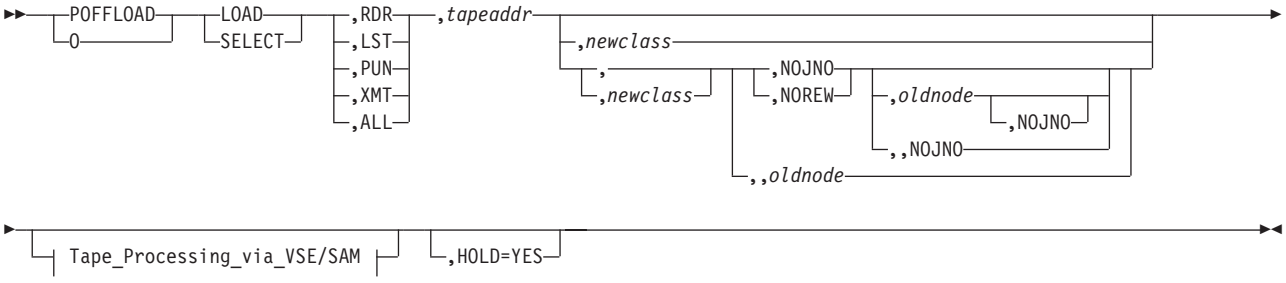
1 You may specify up to four classes. If no class is specified, class A is the default.

Format 2: Backup, Pickup, or Save of the Transmission Queue or All Queues



Tape_Processing_via_VSE/SAM:

Format 3: Loading All or Selected Entries from Tape



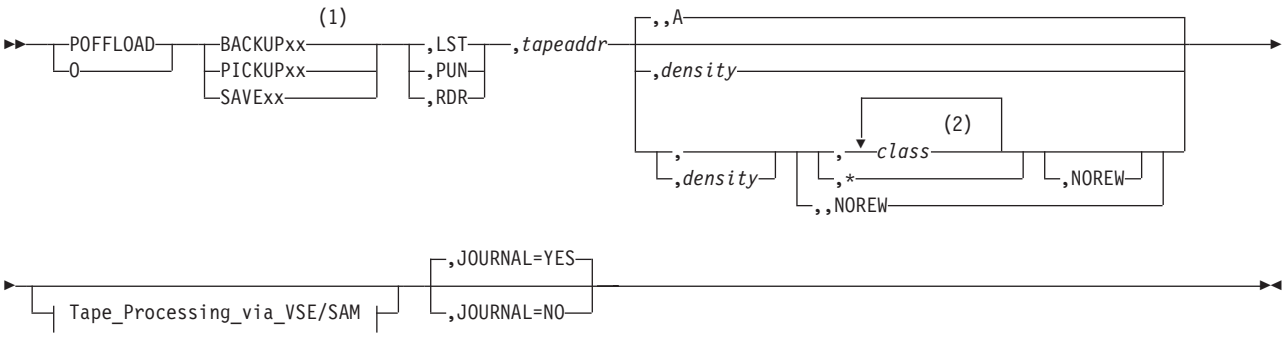
Tape_Processing_via_VSE/SAM:

Format 4: Backup, Pickup, or Save for a Previous Release of VSE/POWER

When producing an Offload tape for a previous-release VSE/POWER system, different Data Block (DBLK) and Data Block Group (DBLKGP) sizes remain transparent to the user, but different sizes of the internal VSE/POWER queue record are not accepted by the target system. According to "Queue Entries on Spool Tape" on page 528, the current queue record length of 368 bytes is used in VSE/POWER 6.4 (VSE/ESA 2.4) and in all higher releases, but this length is different in VSE/POWER 6.3 and older releases. Therefore, when the target system is

- a *6.4 or higher* release of VSE/POWER, you can use the POFFLOAD *Format 1 and Format 2 commands*.
- a *6.3 or lower* (older) release of VSE/POWER, *the following command formats* provide specification of the target release, so that the target queue record size can be honored.

The following format shows how to backup or save a selected local queue for a previous VSE/POWER release (6.3 or older).



Tape_Processing_via_VSE/SAM:

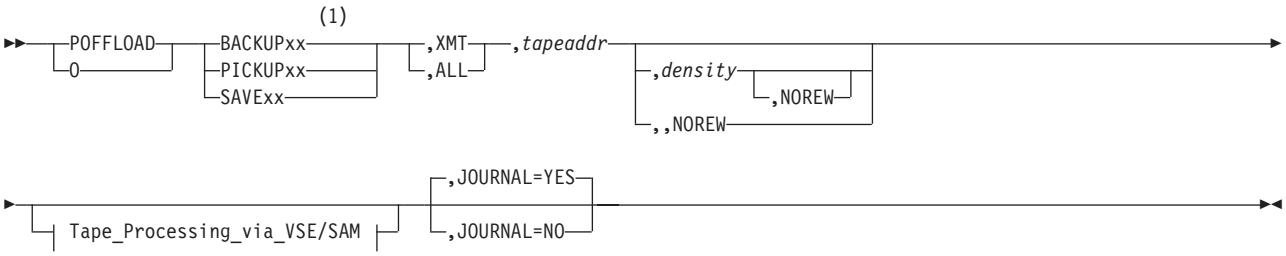
Notes:

1 xx = 63 | 61 | 52 | 51 | 41, representing the previous VSE/POWER target release

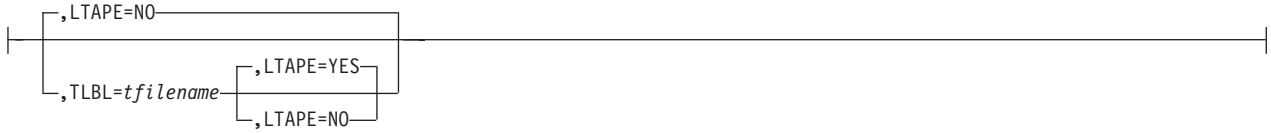
2 You may specify up to four classes. If no class is specified, class A is the default.

The following format shows how to backup or save the transmission queue or all queues for a previous VSE/POWER release (6.3 or older).

POFFLOAD



Tape_Processing_via_VSE/SAM:



Notes:

1 xx = 63 | 61 | 52 | 51 | 41, representing the previous VSE/POWER target release

POFFLOAD Types

BACKUP

You can use the command with this operand to save all entries of a queue, including those which are processing, by creating a backup tape.

While processing a POFFLOAD BACKUP command, VSE/POWER:

1. *Locks the entire queue file* for exclusive use.
2. *Suspends* the processing of any queue entries and of further output spooling.
3. Does *not* allow any of the following commands to be processed: PALTER, PDELETE, PHOLD, PRELEASE, or PRESET.

In other words, **no** other VSE/POWER task can access the queue file. In a shared spooling environment, the command also causes all other sharing systems to wait. Queue entries so saved remain in their queue unchanged.

For a comparison of save-to-tape characteristics, refer also to Table 10 on page 324.

Only one POFFLOAD BACKUP command can be processed by VSE/POWER at a time, and the BACKUP and PICKUP functions are mutually exclusive at any one time.

If your output tape becomes full while the command is processing, then VSE/POWER requests the operator to mount a new tape volume. Again, the queue file remains locked during this time, and 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, or that you use the PICKUP function instead.

BACKUP63 | BACKUP61 | BACKUP52 | BACKUP51 | BACKUP41

Format 4 is similar to Formats 1 and 2.

The suffix of the BACKUPxx operand specifies a previous VSE/POWER target release.

BACKUP63
Backup entries for VSE/POWER 6.3 as contained in VSE/ESA 2.3.

BACKUP61
Backup entries for VSE/POWER 6.1.0/6.1.1/6.1.2 as contained in VSE/ESA 2.1.0/2.1.2/2.2.0.

BACKUP52
Backup entries for VSE/POWER 5.2 as contained in VSE/ESA 1.3

BACKUP51
Backup entries of VSE/POWER 5.1 as contained in VSE/ESA 1.1 and 1.2

BACKUP41
Backup entries for VSE/POWER 4.1 as contained in VSE/SP Version 4.

Depending on the suffix level specified, a POFFLOAD tape with a queue record length of the target release is created. This tape can be used for the POFFLOAD LOAD or POFFLOAD SELECT function on the previous release.

The DBLK size on the current system should not exceed 12,888 bytes to allow for the POFFLOAD migration to a previous release of Version 5 or 4. If the DLBK size is larger, the command is rejected by message *1RF0I*.

Note: Using BACKUPxx limits the functionality and capability of queue entry attributes to that of the target VSE/POWER release.

TLBL= alone, or TLBL= and LTAPE=YES should not be used for VSE/POWER releases that do not support VSE/POWER tape processing via VSE/SAM.

PICKUP
You can use the command with this operand to save all entries of a queue, including those which are processing (DISP=*), by creating an offload tape. The disposition of an entry on the queue will not be altered (as in case of the POFFLOAD SAVE command).

Influence on other system processing:

In contrast to the BACKUP function, the POFFLOAD PICKUP command has the following characteristics:

1. It generally allows queue entries to be saved to tape *without locking the entire queue file* for exclusive use. The exception occurs when a processing entry (DISP=*) or an entry residing in the 'wait for run' subqueue, is being saved to tape - only then are the entire queues blocked from access by other tasks.
2. If a spool entry is not being processed at the time of saving to tape, then only that entry is locked and other spool entries are available as before.
3. All VSE/POWER commands that manipulate queue entries (PALTER, PDELETE, for example) are permitted during PICKUP processing, apart from the exceptional situations when the entire queue file is locked.

Therefore the new PICKUP function combines the advantages of both the BACKUP and SAVE commands, but avoids their inherent operational drawbacks. See also Table 10 on page 324.

If the output tape becomes full while processing the command, the operator will be prompted to mount another tape until all entries indicated have been processed. Similar to the BACKUP function, if during the pause between unmounting the full tape and mounting an empty tape, either an active queue entry (DISP=*) or a 'wait for run' subqueue entry is being saved and continued to the next tape, then ALL queues will be locked during this period.

Only one POFFLOAD PICKUP command is permitted by VSE/POWER at a time, and POFFLOAD PICKUP and POFFLOAD BACKUP are mutually exclusive at any one time. This is also the case across shared spooling systems.

Selection of entries:

When the POFFLOAD PICKUP command is issued, only those queue entries that can be displayed at that time by the operator (PDISPLAY) are eligible for saving to tape. As with the BACKUP function, spool entries ″in creation″ will not be saved (e.g. print output being produced by a running job that has not segmented), even if checkpointed.

You should also note that possibly not all spool entries displayable at the time the PICKUP command is issued and specified to be selected (by queue and class specification) will end up saved on tape. Since generally only individual queue entries are locked during saving to tape, other eligible entries can experience normal spool processing and thereby may disappear from the spool file (DISP=D processing), or the operator may have changed the spool entry status with a VSE/POWER command (PDELETE, PALTER, for example).

Progress recording:

The PICKUP function will begin by flagging all eligible spool entries it finds as ″scheduled″ for offloading, and will notify the operator of the number of entries found by:

```
1Q6NI POFFLOAD PICKUP HAS SCHEDULED nnnnn SPOOL ENTRIES FOR OFFLOADING
      ON cuu AT mm/dd/yyyy
```

It then begins saving those spool entries to tape. During processing the operator will be notified of the function progress every three minutes:

```
1Q6PI POFFLOAD PICKUP PROCEEDING WITH bbbbb OUT OF ccccc SPOOL
      ENTRIES STORED TO TAPE ON cuu
```

PICKUP63 | PICKUP61 | PICKUP52 | PICKUP51 | PICKUP41
Format 4 is similar to Formats 1 and 2.

The suffix of the PICKUPxx operand specifies a previous VSE/POWER target release.

PICKUP63
Save entries for VSE/POWER 6.3 as contained in VSE/ESA 2.3.

PICKUP61
Save entries for VSE/POWER 6.1.0/6.1.1/6.1.2 as contained in VSE/ESA 2.1.0/2.1.2/2.2.0.

PICKUP52
Save entries for VSE/POWER 5.2 as contained in VSE/ESA 1.3

PICKUP51
Save entries of VSE/POWER 5.1 as contained in VSE/ESA 1.1 and 1.2

PICKUP41
Save entries for VSE/POWER 4.1 as contained in VSE/SP Version 4.

Depending on the suffix level specified, a POFFLOAD tape with a queue record length of the target release is created. This tape can be used for the POFFLOAD LOAD or POFFLOAD SELECT function on the previous release.

The DBLK size on the current system should not exceed 12,888 bytes to allow for the POFFLOAD migration to a previous release of Version 5 or 4. If the DBLK size is larger, the command is rejected by message *1RF0I*.

Note: Using PICKUPxx limits the functionality and capability of queue entry attributes to that of the target VSE/POWER release.

TLBL= only, or TLBL= and LTAPE=YES should not be used for VSE/POWER releases that do not support VSE/POWER tape processing via VSE/SAM.

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, and, 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.

Note: Dispatchable jobs of the reader queue with time event scheduling operands are **not** copied to tape. Use the BACKUP or PICKUP operand instead.

SAVE63 | SAVE61 | SAVE52 | SAVE51 | SAVE41
Format 4 is similar to Formats 1 and 2.

The suffix of the SAVExx operand specifies a previous VSE/POWER target release.

SAVE63
Save entries for VSE/POWER 6.3 as contained in VSE/ESA 2.3

SAVE61
Save entries for VSE/POWER 6.1.0/6.1.1/6.1.2 as contained in VSE/ESA 2.1.0/2.1.2/2.2.0.

SAVE52
Save entries for VSE/POWER 5.2 contained in VSE/ESA 1.3

SAVE51
Save entries of VSE/POWER 5.1 contained in VSE/ESA 1.1 and 1.2

SAVE41
Save entries for VSE/POWER 4.1 contained in VSE/SP Version 4.

Depending on the suffix level specified, a POFFLOAD tape with a queue record length of the target release is created. This tape can be used for the POFFLOAD LOAD or POFFLOAD SELECT function on the previous release.

The DBLK size on the current system should not exceed 12,888 bytes to allow for the POFFLOAD migration to a previous release of Version 5 or 4. If the DLBK size is larger, the command is rejected by message *1RF0I*.

Note: Using SAVExx limits the functionality and capability of queue entry attributes to that of the target VSE/POWER release.

TLBL= only, or TLBL= and LTAPE=YES should not be used for VSE/POWER releases that do not support VSE/POWER tape processing via VSE/SAM.

LOAD

A POFFLOAD LOAD command causes queue entries previously saved on tape (by the POFFLOAD BACKUP, PICKUP, or SAVE command) to be reloaded into the VSE/POWER queues. You also can 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 65,024 bytes.

If the queue entries to be reloaded are contained on two or more unlabeled tape volumes, you must provide one POFFLOAD LOAD command for every volume, although the message OFFLOAD SUCCESSFULLY COMPLETED appears after the first volume has been reloaded.

VSE/POWER assigns a new job number and a new creation date to each of the reloaded queue entries and also for each of the segments. Specify NOJNO if this is not desirable.

Note: The following processing rules apply for LOAD and SELECT:

For *PNET systems*, the own node name is determined from the PNODE LOCAL=YES entry. For *non-PNET* systems however, the own node name is (8 bytes) blank.

If the own node name does not change between BACKUP|SAVE and LOAD|SELECT operations, all entries appear after the reloading in the queue display with their original origin and target node ID. This corresponds to the own node name for local (RDR, LST, PUN) queue entries and is, therefore, **not** displayed.

However, when during reloading of local queue entries the own node name changes, then:

- The queue entry's target node ID is *replaced* automatically by the new own local node (hence original RDR/LST/PUN queue entries appear again in the same local queue), and
- The target user Id is *nullified*.
- The original node Id and original user ID remain unchanged (hence, after job creation, output of reloaded jobs is routed to the point of origin and will be added to the XMT queue).

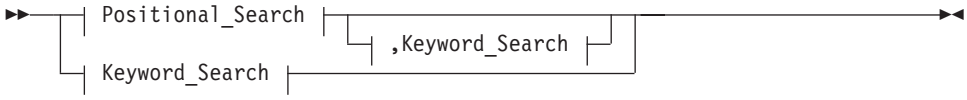
You may influence the latter behavior by specifying the *oldnode* operand (see page 336), which identifies the own node name of the system from which BACKUP|PICKUP|SAVE was done.

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 by message

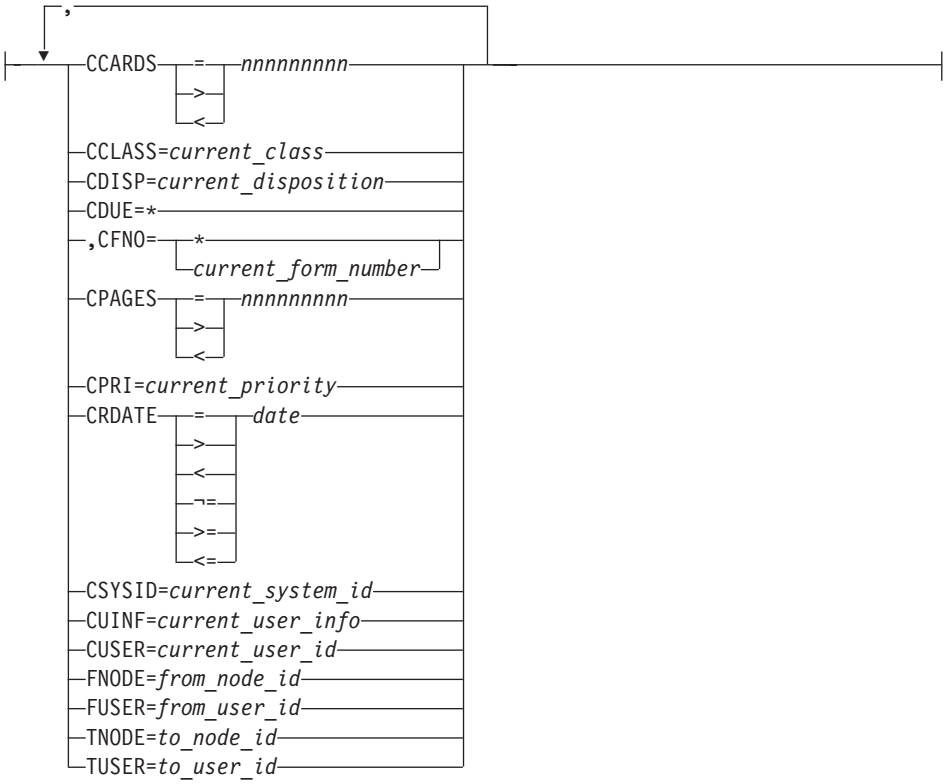
1R41D SPECIFY TAPE SELECT CRITERIA OR PRESS ENTER TO QUIT

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:



Positional_Search:

Keyword_Search:



The selection operands are the same as those of the queue display command. For a description of these operands, see the PDISPLAY command.

POFFLOAD Operands

In any form of the POFFLOAD command, specify the queue whose entries are to be saved to tape or loaded from tape in the form:

LST
to address the list queue.

PUN
to address the punch queue.

RDR
to address the reader queue.

XMT
to address the transmission queue.

ALL
If all entries of all physical VSE/POWER queues (RDR, LST, PUN, XMT) are to be written onto tape (in accordance with the disposition restrictions described for BACKUP, PICKUP, or SAVE) or to be loaded from tape.

Note: If ALL has been used with POFFLOAD BACKUP, PICKUP, or SAVE, you *cannot* reload an individual queue with the LOAD function from the produced tape. Instead, you may specify the SELECT function to address a single queue on the produced tape, as prompted by message 1Q77I.

tapeaddr
For *tapeaddr*, specify the device address of the tape drive that is to be used. You specify this address in the form cuu.

density
Specify the *density* (or tape mode) applicable to the tape drive you are using. See the description of the ASSGN statement in *z/VSE System Control Statements*. Default is the mode defined for the tape drive during system startup.

At completion of the POFFLOAD function with user specified density, VSE/POWER resets the density specification to the standard one defined at IPL or by a later permanent ASSGN statement.

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,PICKUP, 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 and a new creation date 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 every queue entry loaded from tape. This also holds true for segments which are loaded.

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, PICKUP, or SAVE, this operand allows multiple queue off-loads to be saved on one tape. See example below.

If a BACKUP, PICKUP, or SAVE function is started on a tape no longer positioned at its load point, any specification of density is invalid. Instead, the density used previously remains effective.

Note:

1. NOREW is only accepted for non labelled (LTAPE=NO) offloading.
2. With NOREW specified, more than one POFFLOAD BACKUP/PICKUP/SAVE command output can be written to the same tape. Each command produces two trailing tape marks — see “Layout of VSE/POWER Tapes” on page 527. When specified, the NOREW operand causes the final processing to backspace over the final trailing tape mark, so that when the next POFFLOAD BACKUP/PICKUP/SAVE,..., NOREW command is issued, the leading tape mark will overwrite the final tape mark causing each POFFLOAD output to be separated by two tape marks.
3. Such a tape can be addressed with the NOREW option specified with the POFFLOAD LOAD/SELECT commands for reloading of queue entries placed to tape by BACKUP/SAVE/PICKUP commands. Each LOAD/SELECT command terminates when two tapes marks have been read and remains positioned after the second trailing tape mark. The subsequent LOAD/SELECT command will cause the tape to backspace between the two trailing tape marks, then starts reading the first queue entry of the POFFLOAD data as in the normal case when the tape is positioned at the beginning of tape.

   Therefore when manually positioning the tape to LOAD/SELECT queue entries of the 2nd, 3rd, ... BACKUP/PICKUP/SAVE NOREW command output, the tape has to be positioned following the double trailing marks (for example, using the MTC command).
4. Such a tape can also be addressed with the REW=NO option specified with the PDISPLAY entry TAPE=cuu command for displaying of queue entries outputted to tape by BACKUP/PICKUP/SAVE NOREW commands. Positioning of the tape is similar to that described above for LOAD/SELECT.

   This means that a PDISPLAY TAPE=cuu,REW=NO command may be used to position a tape that contains output of multiple POFFLOAD BACKUP/PICKUP/SAVE,..., NOREW comands.
   a. following a PDISPLAY TAPE=cuu,REW=NO command the tape is positioned for either another PDISPLAY TAPE=cuu,REW=NO command or for a POFFLOAD LOAD/SELECT...,NOREW command
   b. following a PDISPLAY TAPE=cuu,REW=NO command the tape may be positioned with a MTC BSF,cuu,1 command for another POFFLOAD BACKUP/PICKUP/SAVE,..., NOREWcommand.

```
d rdr,1
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R46I  READER QUEUE   P D C S  CARDS B
F1 0001 1R46I  PAUSEF1  00010 3 L 1        4   FROM=(SYSA)
F1 0001 1R46I  PAUSEF1  00106 3 L 1        4   FROM=(SYSA)
d lst,s
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R46I    LIST QUEUE   P D C S  PAGES  CC FORM B
F1 0001 1R46I  PAUSEBG  00086 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
F1 0001 1R46I  PAUSEF3  00012 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
mtc rew,490
AR 0015 1I40I  READY
o backup,rdr,490,,1,norew
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON 490
o backup,lst,490,,s,norew
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON 490
mtc rew,490
AR 0015 1I40I  READY
d all,tape=490,out=con,rew=no
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R41I  READER QUEUE   P D C S  CARDS
F1 0001 1R41I  PAUSEF1  00010 3 L 1        4   FROM=(SYSA)
F1 0001 1R41I  PAUSEF1  00106 3 L 1        4   FROM=(SYSA)
F1 0001 1R41I  LIST QUEUE NOTHING TO DISPLAY
F1 0001 1R41I  PUNCH QUEUE NOTHING TO DISPLAY
F1 0001 1R41I  XMIT QUEUE NOTHING TO DISPLAY
d all,tape=490,out=con,rew=no
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R41I    LIST QUEUE   P D C S  PAGES  CC FORM
F1 0001 1R41I  PAUSEBG  00086 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
F1 0001 1R41I  PAUSEF3  00012 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
F1 0001 1R41I  READER QUEUE NOTHING TO DISPLAY
F1 0001 1R41I  PUNCH QUEUE NOTHING TO DISPLAY
F1 0001 1R41I  XMIT QUEUE NOTHING TO DISPLAY
mtc rew,490
AR 0015 1I40I  READY
o load,rdr,490,,norew
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON 490
F1 0001 1Q33I  STOPPED R-OFF,490
o load,lst,490,,norew
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON 490
F1 0001 1Q33I  STOPPED R-OFF,490
d rdr,1
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R46I  READER QUEUE   P D C S  CARDS B
F1 0001 1R46I  PAUSEF1  00010 3 L 1        4   FROM=(SYSA)
F1 0001 1R46I  PAUSEF1  00106 3 L 1        4   FROM=(SYSA)
F1 0001 1R46I  PAUSEF1  00146 3 L 1        4   FROM=(SYSA)
F1 0001 1R46I  PAUSEF1  00147 3 L 1        4   FROM=(SYSA)
d lst,s
AR 0015 1C39I COMMAND PASSED TO VSE/POWER
F1 0001 1R46I    LIST QUEUE   P D C S  PAGES  CC FORM B
F1 0001 1R46I  PAUSEBG  00086 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
F1 0001 1R46I  PAUSEF3  00012 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
F1 0001 1R46I  PAUSEBG  00148 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
F1 0001 1R46I  PAUSEF3  00149 3 D S        1   1        TO=(SYSA) FROM=(SYSA)
```

Figure 64. Example of POFFLOAD NOREW and PDISPLAY REW=NO

oldnode
This operand applies if you performed first the BACKUP, PICKUP, or SAVE operation on one system and then the LOAD or SELECT operation on another, or if the own node name was changed after creation of the tape.

For *oldnode*, specify the node name of the system where the entry was created. If specified, the origin nodeid and/or the target nodeid are updated to the name of the new local nodeid if the old value matches the **oldnode** value. Origin and target user Ids are not changed at all (for details, see the LOAD operand of POFFLOAD at 332).

TLBL=tfilename
For filename specify one to seven characters of a tape label `// TLBL filename...`. It is assumed that the `// TLBL` statement is available to the VSE/POWER partition. For details refer to "Labeled Tape Support" on page 152. For the syntax of the TLBL statement, see *z/VSE System Control Statements*. If specified and the LTAPE operand is not specified, then LTAPE=YES is assumed.

For more information about using the TLBL statement see the description of LTAPE.

LTAPE=YES|NO
If specified, VSE/SAM is used during OPEN, CLOSE and end-of-volume (EOV) tape processing.

The NOREW operand is not allowed for LTAPE=YES.

YES
Indicates VSE/SAM labeled tape processing.

NO
Indicates VSE/SAM unlabeled tape processing.

If TLBL= is specified in addition to LTAPE=NO, an unlabeled tape will still be processed using VSE/SAM. However, VSE/POWER will pass the tfilename value to VSE/SAM as for labeled tape processing. This operand value is normally needed only by Tape Management Systems which might use the `// TLBL` statement to indicate tape information, such as tape density. The TLBL= value then applies to all unlabeled tapes produced.

Note: Unlabeled tape processing may result in more than one output tape being produced; however, VSE/POWER processes only a single unlabeled input tape.

HOLD=YES
Specify this operand if you want to avoid that re-loaded dispatchable queue entries are processed unintentionally. Instead, during re-loading, they will be added to their queue with their disposition set to the corresponding hold DISP (D->H, K->L).

JOURNAL=YES|NO
indicates to simultaneously produce a POFFLOAD journal of the tape(s) contents produced by a POFFLOAD BACKUP/SAVE/PICKUP command. The journal is a written itemization of all of the entries in the order written to tape by the POFFLOAD BACKUP/SAVE/PICKUP function, similar to that displayed by the PDISPLAY ...,TAPE=cuu,OUT=LST command. It is placed in a LST queue entry named $OFJ*nnnn* (where 'nnnn' is the last four digits of the VSE/POWER job number) containing the POFFLOAD BACKUP/SAVE/PICKUP journal contents (see "POFFLOAD Journaling" on page 339for details). The journaling output LST entry has the following attributes:
- it is announced by the message 1Q2AI or 1Q5LI (or 1Q5NI in an error case)
- Priority 9
- Disposition H

- Class A.

VSE/POWER assumes JOURNAL=YES by default. Specify JOURNAL=NO to cause the journal not to be produced. The POFFLOAD BACKUP/SAVE/PICKUP command did not produce a journal prior to VSE/POWER Version 7.

Examples of the POFFLOAD Command

POFFLOAD SAVE,LST,280,,AM
Saves dispatchable list queue entries of classes A and M on the tape mounted on the drive at the address 280 and creates a POFFLOAD journal LST entry. VSE/POWER uses the default density. It continues processing queue entries in parallel.

O BACKUP,ALL,280
Produces a backup of all queues and creates a POFFLOAD journal LST entry. 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 and produces a ″true snapshot″.

O PICKUP,XMT,280,JOURNAL=NO
Saves all entries (job and output type) of the transmission queue on tape drive 280 and produces no POFFLOAD journal entry. It continues processing queue entries in parallel.

O BACKUP,RDR,280,,*
Produces a backup of all entries of the reader queue, regardless of their class assignments and creates a POFFLOAD journal LST entry.

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.

POFFLOAD BACKUP63,LST,280
Produces a backup copy of all class A entries of the list queue on the tape mounted on drive 280 and creates a POFFLOAD journal LST entry. Due to the tailored 6.3 queue record length on tape (256 bytes), this tape should only be used for POFFLOAD LOAD/SELECT on the target VSE/POWER 6.3 system. It will be accepted on any system higher than VSE/POWER 6.3, however.

POFFLOAD BACKUP,LST,280,,AH,TLBL=TLABOUT,JOURNAL=NO
Produces a backup copy of all class A and class H entries of the list queue for a labeled tape mounted on drive 280 and doesn’t create a POFFLOAD journal.

POFFLOAD Journaling

The POFFLOAD JOURNAL= operand allows an operator issuing a POFFLOAD BACKUP/SAVE/PICKUP command writing VSE/POWER spool entries to tape, to further indicate whether a recorded journal of the saved entries should be created. This eliminates the reoccurring necessity of printing the tape(s) contents afterwards for control purposes:

- having the operator to individually remount each POFFLOAD tape produced and.
- create the journal report with the PDISPLAY ...,TAPE=cuu,OUT= command for each separate POFFLOAD tape produced.
- furthermore, most important console messages accompanying the offload process are also recorded in the journal, relieving the operator's effort to search the hardcopy in case problems are noticed later

Because the POFFLOAD function may be requested at a time of system emergency shutdown or high z/VSE system stress (low resources) the journaling (but not POFFLOAD) will be discontinued if system resources for the journaling are not immediately available, or the journaling output as sensed by a timer wait interrupt is slow or hanging. This permits the POFFLOAD function to complete without hindrance, and the operator can later obtain a listing of the tape(s) contents using the PDISPLAY ...,TAPE=cuu,OUT=LST command if needed. The maximum wait time allowed until the journaling function is cancelled, is specified by a field in storage with the eye catcher ">>>POFFLOAD JOURNAL MAX TIMEOUT VALUE 4 BYTES UNIT 0.1 SEC HERE=" (the present value is X'00000258'=60.0 seconds).

If the POFFLOAD command is prematurely ended by the PSTOP cuu or PGO cuu,CANCEL commands, then the existing journal tape will still be placed on the LST queue.

Journal Contents

The journal is a written itemization of all of the entries in the order written to tape by the POFFLOAD command, similar to that displayed by the PDISPLAY ...,TAPE=cuu,OUT=LST command. In addition, the report contains further informational messages (message number "1R4CI"), journal prolog messages (e.g. tape labels VOL1 and HDR1 if any), tape begin time, date and volume sequence number journal epilog messages (e.g. begin and end times, total tapes) and furthermore: most of the important POFFLOADing messages (e.g. 1Q7EA)

In addition, the individual spool entry journal message is modified beginning in column 89 with the decimal sequence number of the spool entry on tape in the format of '*qnnnnnn*' where '*q*' is the queue identifier ('R', 'L', 'P' or 'X') and '*nnnnnn*' is the sequence number. See the figure below for an illustration of the typical POFFLOAD journal content layout.

Table 11. POFFLOAD Typical Journal Layout

```
1R4CI                  POFFLOAD JOURNAL BEGIN
1R4CI
1R4CI  JOURNAL LST ID= $OFJnnnn nnnnn
1R4CI  INPUT COMMAND=POFFLOAD type,queue,cuu,...
1R4CI  TAPE VOL1 LABEL=
1R4CI  (NONE) or label-values
1R4CI  TAPE HRD1 LABEL=
1R4CI  (NONE) or label-values
1R4CI
1R4CI  DATE BEGIN=aa/bb/cc,TIME BEGIN=hh:mm:ss,TIME NOW=hh:mm:ss,VOL=0001
1R41I  READER QUEUE   P D C S  CARDS-------------------------------------------
1R41I  entry1   nnnnn p d c        m                                        R000001
       D=mm/dd /yyyy DBGP=hhhhhh ...
1R41I  entry2   nnnnn p d c        m                                        R000002
       D=mm/dd /yyyy DBGP=hhhhhh ...                          n
. . .
1R41I    LIST QUEUE   P D C S  PAGES  CC FORM-----------------------------------
1R41I  entryn   nnnnn p d c        m   n                                    Lnnnnnn
       D=mm/dd /yyyy DBGP=hhhhhh ...
. . .
1R4CI
1R41I   PUNCH QUEUE   P D C S  CARDS  CC FORM-----------------------------------
1R41I  entrym   mmmmm p d c        m   n                                    Pnnnnnn
       D=mm/dd /yyyy DBGP=hhhhhh ...
. . .
1R41I    XMIT QUEUE   P D C I  LINES-------------------------------------------
1R41I  entrym   mmmmm p d c x      m                                        Xnnnnnn
       D=mm/dd /yyyy DBGP=hhhhhh ...
. . .
1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON cuu, JOURNAL LST ENTRY
       $OFJnnnn CREATED
1R4CI
1R4CI  DATE BEGIN=aa/bb/cc,TIME BEGIN=hh:mm:ss,TIME NOW=hh:mm:ss,VOL=nnnn(TOTAL)
1R4CI
1R4CI                      POFFLOAD JOURNAL END
```

Journaling Console Messages

When the POFFLOAD command terminates normally then the following message occurs on the console and in the journal:

```
1Q2AI   OFFLOADING SUCCESSFULLY COMPLETED ON cuu,  JOURNAL LST
        ENTRY $OFJnnnn CREATED
```

When the POFFLOAD command is terminated by the operator (e.g. via PSTOP cuu or PGO cuu,CANCEL) then if any spool entry has been written to tape, the message occurs:

```
1Q5LI   VSE/POWER OFFLOAD TERMINATED FOR UNIT cuu, JOURNAL LST
        ENTRY $OFJnnnn CREATED
```

If an error occurs causing the offload task to be terminated (e.g. disk I/O) then an attempt will be made to close the journal and place it on the LST queue documentated by the message:

```
1Q5NI  OFFLOADING ERROR ON task,cuu, JOURNAL LST ENTRY $OFJnnnn CREATED
```

If an error occurs causing the abnormal termination of the journaling function then the following message occurs.

```
1Q5MI  OFFLOAD {BACKUP/SAVE/PICKUP} JOURNALING ON tapecuu TERMINATED, RC=nnnn, task TRACE=ccccc
```

Journal Examples

The following examples illustrate two POFFLOAD BACKUP,ALL journals, one for a single offload tape volume and another for a multivolume offload.

Table 12. Example of POFFLOAD Journal Single Volume

```
1R4CI                          POFFLOAD JOURNAL BEGIN
1R4CI
1R4CI  JOURNAL LST ID=$OFJ0055 00055
1R4CI  INPUT COMMAND=BACKUP,ALL,490,TLBL=TLBL1
1R4CI  TAPE VOL1 LABEL=
1R4CI  VOL1VOL001
1R4CI  TAPE HDR1 LABEL=
1R4CI  HDR1LABEL1 RETAIN=365VOL00100010001      00500500600500000000IBMDOSVS
1R4CI
1R4CI  DATE BEGIN=01/05/05,TIME BEGIN=21:49:32,TIME NOW=21:49:44,VOL=0001
1R41I  READER QUEUE   P D C S  CARDS-----------------------------------------
1R41I  PRTDUMPA 00019 3 L 0        7   FROM=(SYSA)                                    R000001
       D=12/17/2004  DBGP=000001
1R41I  PRTDUMPB 00020 3 L 0        7   FROM=(SYSA)                                    R000002
       D=12/17/2004  DBGP=000001
1R41I  PAUSEBG  00030 3 L 0        4   FROM=(SYSA)                                    R000003
       D=12/17/2004  DBGP=000001
. . .
1R41I  PAUSEFB  00009 3 L T        4   FROM=(SYSA)                                    R000021
       D=12/17/2004  DBGP=000001
1R41I    LIST QUEUE   P D C S  PAGES  CC FORM-------------------------------------
1R41I  TLBL     00032 3 D A        1   1                                          L000022
       D=01/05/2005  DBGP=000001 L=00000011
1R41I  CATAL    00033 3 K A        3   6        FROM=POWER271                     L000023
       D=01/05/2005  DBGP=000001 L=00000009
1R41I  CATAL    00034 3 K A        3   6        FROM=POWER271                     L000024
       D=01/05/2005  DBGP=000001 L=00000009
1R41I  AUTONAME 00035 3 D A        2   1        FROM=POWER271                     L000025
       D=01/05/2005  DBGP=000001 L=00000003
. . .
1R41I  AUTONAME 00043 3 H A        2   1        FROM=POWER271                     L000034
       D=01/05/2005  DBGP=000001 L=00000003
1R41I   PUNCH QUEUE   P D C S  CARDS  CC FORM-------------------------------------
1R41I  PACCOUNT 00048 1 L 0      210   1        FROM=POWER271                     P000035
       D=01/05/2005  DBGP=000001
1R41I  PACCOUNT 00031 1 L A      170   1                                          P000036
       D=01/05/2005  DBGP=000001
1R41I    XMIT QUEUE   P D C I  LINES-----------------------------------------
1R41I  TLBL     00049 3 D A R     12   TO=POWER272 FROM=POWER271                  X000037
       D=01/05/2005  DBGP=000001
1R41I  CATAL    00027 3 D A L      9   TO=POWER282                                X000038
       D=12/17/2004  DBGP=000001 P=00000003
1R41I  IPW$$NU  00050 3 K 0 L     23   TO=POWER272(ALBSPA) FROM=POWER271          X000039
       D=01/05/2005  DBGP=000001 P=00000004
. . .
1R41I  ARCHPTF  00053 3 D A L     14   TO=POWER272 FROM=POWER271                  X000042
       D=01/05/2005  DBGP=000001 P=00000003
1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON 490, JOURNAL LST ENTRY
       $OFJ0055 CREATED
1R4CI
1R4CI  DATE BEGIN=01/05/05,TIME BEGIN=21:49:32,TIME NOW=21:49:57,VOL=0001(TOTAL)
1R4CI
1R4CI                          POFFLOAD JOURNAL END
```

Table 13. Example of POFFLOAD Journal Multivolume

```
1R4CI                          POFFLOAD JOURNAL BEGIN
1R4CI
1R4CI  JOURNAL LST ID=$OFJ0062 00062
1R4CI  INPUT COMMAND=BACKUP,ALL,491,TLBL=TLBL1
1R4CI  TAPE VOL1 LABEL=
1R4CI  VOL1VOL001
1R4CI  TAPE HDR1 LABEL=
1R4CI  HDR1LABEL1 RETAIN=365VOL00100010001      00500600600600000000IBMDOSVS
1R4CI
1R4CI  DATE BEGIN=01/06/05,TIME BEGIN=17:05:27,TIME NOW=17:05:36,VOL=0001
1R41I  READER QUEUE   P D C S  CARDS-------------------------------------------
1R41I  PRTDUMPA 00019 3 L 0        7   FROM=(SYSA)                              R000001
       D=12/17/2004  DBGP=000001
. . .
1R41I  PAUSEFB  00009 3 L T        4   FROM=(SYSA)                              R000021
       D=12/17/2004  DBGP=000001
1R41I    LIST QUEUE   P D C S  PAGES  CC FORM-------------------------------------
1R41I  TLBL     00032 3 D A        1   1                                        L000022
       D=01/05/2005  DBGP=000001 L=00000011
1R41I  CATAL    00033 3 K A        3   6        FROM=POWER271                   L000023
       D=01/05/2005  DBGP=000001 L=00000009
. . .
1R41I  $OFJ0060 00060 9 H A        2   1                                        L000032
       D=01/06/2005  DBGP=000001 L=00000112
1R41I  $QFL0054 00054 9 H A      353   1                                        L000033
       D=01/05/2005  DBGP=000040 L=00018689
       VOL=001
1QA2I  VSE/POWER MULTI-VOLUME TAPE COMPLETE FOR $QFL0054 00054 LST
       VOLUME=001 ON 491 FOR L-OFF,491
1R4CI
1R4CI  DATE BEGIN=01/06/05,TIME BEGIN=17:05:27,TIME NOW=17:05:56,VOL=0002
1R41I  $QFL0054 00054 9 H A      353   1                                        L000033
       D=01/05/2005  DBGP=000040 L=00018689
       VOL=002
1QA2I  VSE/POWER MULTI-VOLUME TAPE COMPLETE FOR $QFL0054 00054 LST
       VOLUME=002 ON 491 FOR L-OFF,491
1R4CI
1R4CI  DATE BEGIN=01/06/05,TIME BEGIN=17:05:27,TIME NOW=17:06:04,VOL=0003
1R41I  $QFL0054 00054 9 H A      353   1                                        L000033
       D=01/05/2005  DBGP=000040 L=00018689
       VOL=003(LAST)
1QA2I  VSE/POWER MULTI-VOLUME TAPE COMPLETE FOR $QFL0054 00054 LST
       VOLUME=003(LAST) ON 491 FOR L-OFF,491
1R41I  DITESYSA 00040 3 L A        4   1        FROM=POWER271(SYSA)             L000034
       D=01/05/2005  DBGP=000001 L=00000022
. . .
1R41I   PUNCH QUEUE   P D C S  CARDS  CC FORM-------------------------------------
1R41I  PACCOUNT 00048 1 L 0      210   1        FROM=POWER271                   P000040
       D=01/05/2005  DBGP=000001
1R41I    XMIT QUEUE   P D C I  LINES-------------------------------------------
1R41I  TLBL     00049 3 D A R     12   TO=POWER272 FROM=POWER271                X000041
       D=01/05/2005  DBGP=000001
1R41I  CATAL    00027 3 D A L      9   TO=POWER282                              X000042
       D=12/17/2004  DBGP=000001 P=00000003
. . .
1R41I  PACCOUNT 00031 1 D A P    170   TO=POWER272                              X000047
       D=01/05/2005  DBGP=000001
1Q2AI  OFFLOADING SUCCESSFULLY COMPLETED ON 491, JOURNAL LST ENTRY
       $OFJ0062 CREATED
1R4CI
1R4CI  DATE BEGIN=01/06/05,TIME BEGIN=17:05:27,TIME NOW=17:06:05,VOL=0003(TOTAL)
1R4CI
1R4CI                          POFFLOAD JOURNAL END
```

PRELEASE: Removing Entries of Physical Queues from the Hold or Leave State

The command takes one or more VSE/POWER queue entries 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 14 operands.

Note: A central operator PRELEASE command is confirmed on the console by

```
1R88I  OK : 6 ENTRIES PROCESSED BY PRELEASE,LST,*SSL
```

for example, thereby indicating the number (6) of released queue entries together with the corresponding operator command.

Using the command changes the sequence of the queue entries in the specified queue. They are processed in accordance with the classes and priorities defined for them. If an affected queue entry is in the reader (RDR) queue and was executed previously, VSE/POWER **gives the job a new jobnumber**.

After the command has been processed, the disposition of an affected queue entry is changed as follows:

An original disposition of L is changed to K.
An original disposition of H is changed to D.

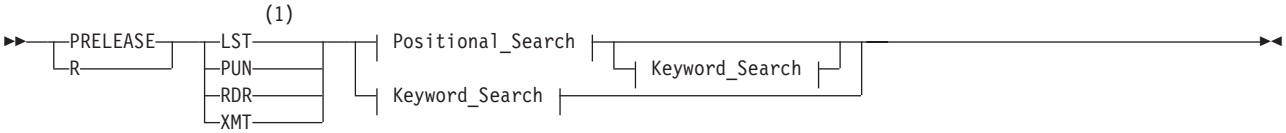
Note: Queue entries with a temporary disposition of A, X, or Y are not changed.

A job placed into the reader queue from a workstation 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 terminal user.

For VSE/POWER queues addressable by the PRELEASE command, refer to Table 9 on page 225.

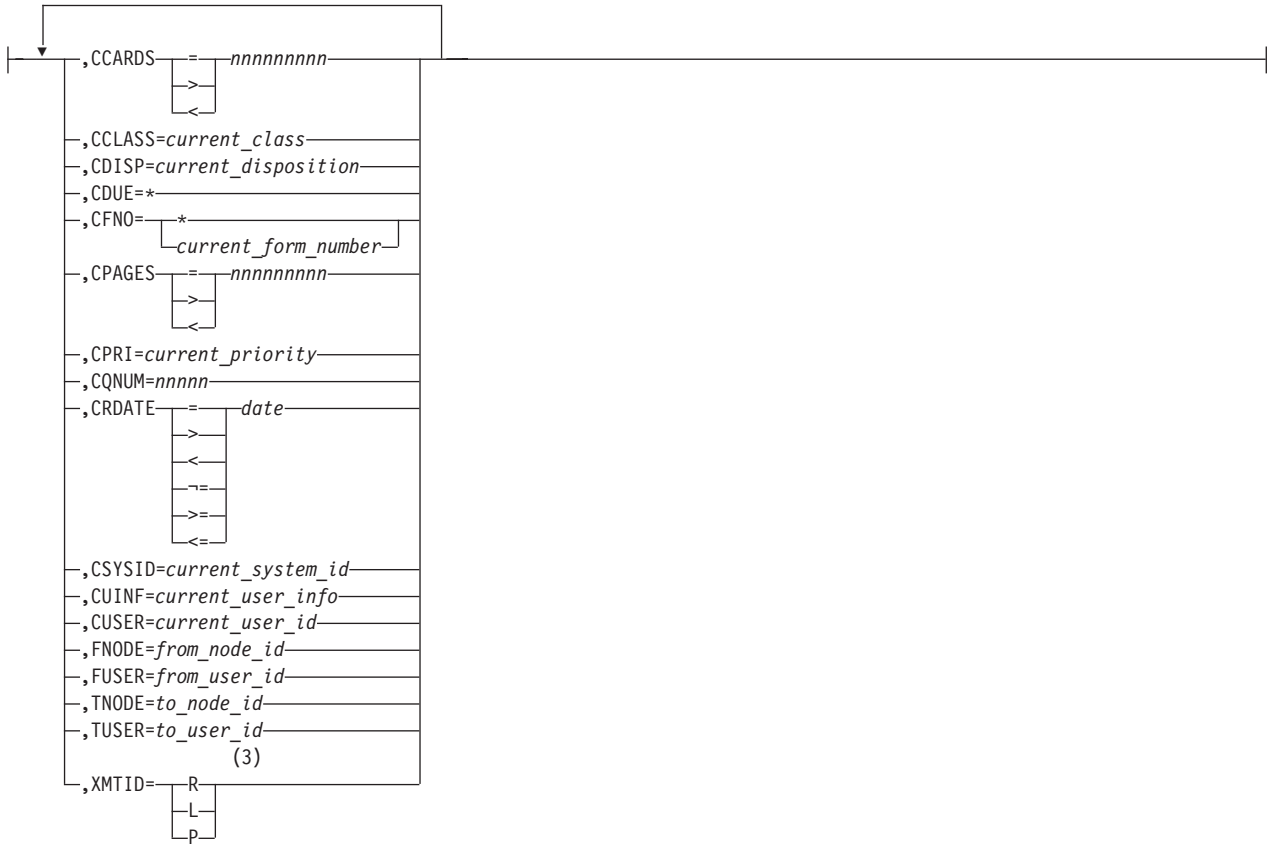
PRELEASE

Format of the Command



Positional_Search:

Keyword_Search:



Notes:

1 RDR/LST/PUN/XMT queues are called physical queues in contrast to the logical CRE/DEL queues.

2 If only 'ALL' or 'class' is specified, queue entries in the 'L' state are not released.

3 XMTID is meaningful only if XMT is specified.

Specify the queue for which the command is to be executed, in the form:

LST
to address the list queue.

PUN
to address the punch queue.

RDR
to address the reader queue.

XMT
to address the transmission queue.

Positional Search Operands

ALL
specifies that all VSE/POWER jobs in the specified queue are to be released,**except** those with **disposition L**. ALL is the default if you specify a queue with one or more keyword search operands in order to have a certain group of entries in the queue to be released.

Note: Using the SET CONFIRM=PRELEASE,QUEUE,ALL autostart statement you can protect your system against inadvertent PRELEASE'ing of DISP=H jobs. For details, see "SET: Setting VSE/POWER Startup Control Values" on page 473.

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 L**.

jobname
specifies the name by which the specific job is known to VSE/POWER.

PRELEASE with 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 (RBS) 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.

jobnumber
specifies the number assigned to the specific job by VSE/POWER. 1 to 5 digits may be specified. 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
specifies the VSE/POWER-assigned segment number if only a certain output segment of the named job is to be released. 1 to 3 digits may be specified. You can use this operand only together with the VSE/POWER assigned job number. To get the correct job suffix S=nnn, issue a PDISPLAY command for the job.

jobn*
Specifies those queue entries whose name begin with the specified characters. As search argument, you can specify for *jobn* one to eight characters.

Keyword Search Operands

CCARDS=|<|>nnnnnnnnn
indicates that only reader and punch queue entries are to be addressed whose current total card count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified card nnnnnnnnn value, which must be greater than zero. A card count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P). When CCARDS is specified for the LST queue or for list type entries of the XMT queue (I=L), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

   ```
   1R52I LIST QUEUE NOTHING TO RELEASE or
   1R52I NOTHING TO RELEASE
   ```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For reader or punch type entries in the XMT queue the 'LINES' column actually presents statements, that means cards. This is the value against the CCARDS operand is compared.
4. For active entries (DISP=*) in the PUN or XMT queue, 'cards left to be processed' are displayed, although the entries were selected according to their total card count, which is not displayed then.

CCLASS=current_class
requests an affected queue entry (or entries) to be released if the class currently assigned to the entry matches the specified class. This operand will overwrite a possible specification of the positional 'class' operand. Specification of either 'class' or the 'CCLASS=' operand will expedite the access path to the VSE/POWER queue file.

CDISP=current_disposition
requests an affected queue entry (or entries) to be released if the disposition currently assigned to the entry matches the specified disposition.

Note: queue entries with a temporary disposition(DISP=) of *A*, *X* or *Y* are not addressed for releasing.

CDUE=*
indicates that all jobs are to be released for which time event scheduling operands have been specified. These jobs can be either in the RDR or the XMT queue.

Note that a job in the reader queue does not start processing immediately. First, the time event scheduling information is checked to determine the next due date.

For a job in the transmission queue, however, the time event scheduling information is ignored and the transmission of the job starts as soon as possible.

CFNO=*|current_form_number
requests an affected queue entry (or entries) to be released if the form number defined for the output of the entry matches the specified form number. One can specify '*' to process those queue entries which have no form number, or the four-character form number of the queue entries to be processed.

CPAGES=|<|>nnnnnnnnn
indicates that only list queue entries are to be addressed whose current total page count is:

```
equal to     (=)
less than    (<)
greater than (>)
```

the specified page nnnnnnnnn value, which must be greater than zero. A page count of 1-9 digits may be specified, with or without leading zeros.

Note:

1. This operand is effective only for entries of the LST queue or for list type entries of the XMT queue (I=L). When CPAGES is specified for the RDR or PUN queue or for reader/punch type entries of the XMT queue (I=R/P), then these queue entries are simply ignored during selection by the requested command, resulting, for example, in the following command response

   ```
   1R52I READER QUEUE NOTHING TO DISPLAY, or
   1R52I NOTHING TO ALTER/DELETE/...
   ```

2. The operands CCARDS and CPAGES are mutually exclusive. When both are specified, the command is rejected by message 1R52I.
3. For list type entries in the XMT queue the standard display presents 'LINES'. Use the 'P=n...n' page count of the FULL=YES display, when you select entries by the CPAGES operand.
4. For active entries (DISP=*) in the LST or XMT queue, 'pages left to be processed' are displayed, although the entries were selected according to their total page count, which is not displayed then.

CPRI=current_priority
requests an affected queue entry (or entries) to be released if the priority currently assigned to the entry matches the specified priority.

CQNUM=nnnnn
indicates that the requested release is to be done if the queue entry's internal queue number matches the specified `nnnnn` value.

This unique number assigned by VSE/POWER can be made visible by the PDISPLAY command QNUM=nnnnn display field. See also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257. Use the CQNUM search operand if more than one queue entry has the same jobname and jobnumber and the entries otherwise cannot be distinguished by other attributes for command selection. You can specify 1 to 5 digits.

Note: Only numbers within the range indicated by `TOTAL NUMBER OF QUEUE RECORDS` (see Figure 7 on page 46) are accepted for searching. Only one queue entry can satisfy the search criterion.

CRDATE=|>|<|¬=|>=|<=date
In this string:

CRDATE
indicates that those queue entries are to be released whose creation date is one of the following in relation to the specified date:

```
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

```
1) mm/dd/yy or dd/mm/yy
2) mm/dd/yyyy or dd/mm/yyyy
```

Do not specify mm greater than 12, dd greater than 31, and yy or yyyy greater than the current year.

Notes:

1. If you specify a 2 digit yy-year, VSE/POWER expands it for internal processing and comparison to a 4 digit year according to the 'fix-88-window' rule:

   ```
   if yy is greater than (>) 88, then 19yy
   if yy is less than or equal (<=) 88, then 20yy
   ```

2. The format of the system date must not be changed after VSE/POWER has been initialized or between warm starts of VSE/POWER (see "Functions Not Expecting Date Format Changes" on page 128).
3. In a shared spooling environment, all systems must have the same date format.

CSYSID=current_system_id
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.

CUINF=current_user_info
requests an affected queue entry to be released if the queue entry's user information matches the specified current_user_info value. User information is provided to a queue entry by the * $$ JOB/LST/PUN JECL statements or by the PALTER command, and can be made visible by a PDISPLAY ...,FULL=YES request, which presents this information left-justified with trailing blanks, all enclosed in quotes via the U='...' display field (see also "Format 1-4: PDISPLAY LST,FULL=YES" on page 257). When user information contains

1. neither embedded blanks nor commas and is displayed, for example, as

   U='PETER ', it can be searched for using
   CUINF=PETER, or
   CUINF='PETER' or
   CUINF='PETER '

2. an embedded blank (or comma) and is displayed, for example, as

   U='PETER BOY ', it can be searched for using
   CUINF='PETER BOY ', or
   CUINF='PETER BOY'

3. an embedded blank (or comma) and a single quote, shown for example as

U='PETER'S BOOK ', it can be searched for using CUINF='PETER''S BOOK'

The CUINF specification (when enclosed in single quotes) can contain trailing blanks up to the field limit of 16 characters.

CUSER=current_user_id
requests an affected queue entry to be released if the queue entry's "from user" or "to user" matches the specified user ID.

FNODE=from_node_id
requests an affected queue entry to be released if the entry's "from node" name matches the node name specified by *from_node_id*. However, VSE/POWER does not verify that the specified node name is defined in the network-definition table.

FUSER=from_user_id
requests an affected queue entry to be released if the entry's "from user" ID matches the user ID specified by *from_user_id*.

Specifying FUSER=LOCAL causes VSE/POWER to search for job entries with an explicit user ID of LOCAL. It does not cause the release of entries originated by local users.

TNODE=to_node_id
requests an affected queue entry to be released if the entry's "to node" name matches the node name specified by *to_node_id*. This operand is valid only if your specification for queue is XMT.

TUSER=to_user_id
requests an affected queue entry to be released if the entry's "to user" ID matches the user ID specified by *to_user_id*.

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 ID of LOCAL. It does not cause a release of entries destined for local users.

XMTID=R|L|P
This operand is effective only if your command applies to the XMT queue; for other queue types the operand is ignored. For XMT queue, you can address one of the three transmission subqueues presented as R=Reader, L=List, P=Punch by the 'I' identification column of a PDISPLAY XMT command display line.

The operand indicates that the queue entry is to be released if the queue entry's 'I'-identification matches the XMTID value given in this command. For a PDISPLAY XMT example, see page 261.

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<=05/15/05
Releases all entries in the reader queue whose creation date is less than or equal to May 15, 2005. Or, for 4-digit years:

PRELEASE RDR,CRDATE<=05/15/2005
Releases all entries in the reader queue whose creation date is less than or equal to May 15, 2005.

R LST,CFNO=SPEC
Releases all entries in the LST queue which require forms SPEC.

PRESET: Resetting 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 the next VSE/POWER startup.

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. This leaves the assigned security zone inoperable for a certain period.
- Queue entries which were processed are required urgently. This refers to:
  - Jobs in execution
  - Output being printed/ punched
  - Jobs/output being transmitted
  - Jobs/output in access by the spool-access support GET Service.
- No other sharing VSE/POWER system has received message 1QB7I since the abnormal termination of the failing system.

The formats of the PRESET command are as follows:

- Format 1: Resetting Queue Entries Currently in Execution State
- Format 2: Removing SECNODE Security Zone Entry of a Terminated Sysid

Format 1: Resetting Queue Entries Currently in Execution State

Notes:

1 You can specify up to nine IDs.

Format 2: Removing SECNODE Security Zone Entry of a Terminated Sysid

Notes:

1 You can specify up to nine IDs.

system_id
specifies the ID of the VSE/POWER controlled system that terminated abnormally.

For *system_id*, specify the ID, the one-digit number which VSE/POWER displays with message 1Q12I during startup. You can specify up to nine IDs.

VSE/POWER verifies that none of the specified IDs is that of your own system. It requests the operator to confirm the IDs that are specified in the PRESET command.

A specification error in the command can lead to unexpected results. If, for example, you enter the ID of a running system that is not your own, then VSE/POWER cannot determine that the ID is wrong and sets all jobs for this system into the inactive state. This could cause data integrity problems later on.

SECNODE
This operand resets the VSE/POWER SECNODE start-up value for the given sysid(s) in the shared system secnode table which affects job dispatching in the shared environment. This means that after execution of the command, no job will wait for execution on the shared system(s) specified by the ″system_id″ of the command.

If after execution of the command no remaining shared system has the SECNODE values as specified by some job waiting on execution, the job may be executed on any shared system (if the SYSID= operand was not specified), but without security authorization.

Example of the PRESET Command

PRESET 2,4
Initiates queue-file recovery for the sharing systems with the IDs 2 and 4.

PRESET SECNODE,2,4
Resets the SECNODE values for the sharing systems with IDs 2 and 4.

PRESTART: Restarting 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.

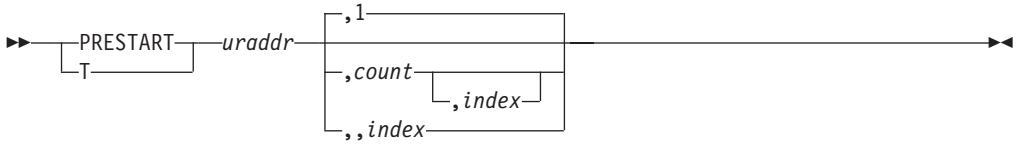
Restarting printed output is not possible for page numbers higher than 999,999. For page/card count, see also page 253.

If you enter a PRESTART command while VSE/POWER is printing separator pages, these pages are printed a second time.

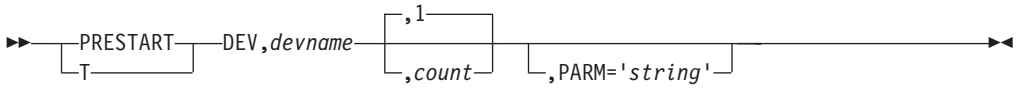
The formats of the PRESTART command are as follows:

- Format 1: Restarting a Local Print/Punch Output
- Format 2: Requesting Restart for a Printer Controlled by a Device Driving System

Format 1: Restarting a Local Print/Punch Output



Format 2: Requesting Restart for a Printer Controlled by a Device Driving System



DEV
Code this operand as shown. It indicates to a subsystem, for example PSF, 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 cuu address of the 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 999999 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, this is 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 processing 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 and displays 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.
- A CPDS defined page (see also "Page Counting for CPDS Queue Entries" on page 146).

Note: Any lines spooled *before* the completion of the first page will not increment the VSE/POWER page count and can therefore not be addressed by a restart page count of 1 (use 0 instead).

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 also 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', 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.

PSEGMENT: Segmenting Output Spooled by Execution Writer Tasks

Use this command to segment output of active jobs before the job reaches its intended end. The output thereby changes its life-cycle state from 'in creation' to 'queued', and the output queue entry is made available for further processing in the LST/PUN/XMT queue as if it had been created by count-driven (RBS) segmentation (see also "Output Segmentation" on page 117). The same segmentation function can also be requested by the PALTER...,SEGMENT command (see "Format 3: Altering to Segment Job Output In Creation" on page 209).

partition
The static or dynamic partition ID that spools output to be segmented.

uraddr
The cuu address of the VSE/POWER spooled output device of the named partition.

PAGE
Requests segmentation on the next page boundary (default for LST output).

CARD
Requests segmentation on the next data statement (default for PUN output).

IMM
Requests immediate segmentation at the current spooling position.

Notes:

1. 'PAGE' and 'CARD' are used synonymously.
2. The 'PSEGMENT ...,IMM' command produces output with a record count of zero if the command is requested again before the execution writer task has spooled a record in the meantime.
3. The command may be issued by a local operator or a SAS program, or be XMITed from another node, to initiate segmentation, which is then processed by the execution writer task.

Returned messages and codes:

- When the command has executed successfully, message 1R88I OK is returned to the local operator and RC/FDBK=00/01 to the SAS program.
- When the segment request has been finally processed (e.g., page boundary reached and segment created) by the execution writer task, the task issued message:

  ```
  1053I OUTPUT SEGMENT FOR jobname jobnumber suffix partition-id, cuu
  ```

- When the specified queue entry cannot be found due to incorrect specification or because the queue entry has meanwhile been completed, message

  ```
  1R88I NOTHING TO SEGMENT
  ```

  is returned to the local operator and to the SAS program.
- When the queue entry is spooled with DISP=I|T, the command is rejected with

  ```
  1R9BI commandcode SEGMENT REQUEST IGNORED FOR DISP=I|T
  ```

- When no DBLK group is available at all, meaning that the DBLK group cushion is used up, the command is rejected with

  ```
  1R9BI commandcode SEGMENT REQUEST IGNORED DUE TO EMPTY DBLKGP CUSHION
  ```

Interaction with Count-Driven (RBS) Segmentation:

If RBS segmentation was previously requested by the '* $$ LST|PUN RBS=nnnn' statement for output addressed by the PSEGMENT command, the specified output entry will be segmented not on the desired RBS boundary but rather at the next page or card, or even immediately. This depends on the option of the PSEGMENT command. Therefore, the created segment will not show the RBS number of pages or cards. The next segment will be created using the original RBS value.

Collect Specification Values for Command Example

The command initiates segmentation for output spooled for the specified device in the named partition. Segmentation is triggered either immediately or, if PAGE is specified, as soon as the top of the next page is reached, such as by spooling a Skip to Channel 1. For these execution writer tasks, the 'PDISPLAY CRE,PART' or 'PDISPLAY A,PART' display lines or also message 1Q38A present all information needed to specify the requested command operands as shown below:

```
pdisplay cre,part,f2
F1 0001 1R4BI   CREATE QUEUE  C I  LINES B DBGP   QNUM  TASK    OWNER
FI 0001 1R4BI  CICSICCF 01855 A L   1436   000003 00519  F2 FEE JOB=CICSICCF

pdisplay a,part,f2
F1 0001 1R4BI   F2,FEC,L2,  CICSICCF,0855,2
FI 0001 1R4BI   F2,FEE,,  CICSICCF,01855,A     1436 LINES SPOOLED,QNUM=00519
---
F1 0001 1Q38A  NO DASD SPACE AVAILABLE FOR F2, FEE
---
PSEGMENT F2,FEE,IMM
```

PSETUP: Checking 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, to complete the still pending message 1Q40A, the PGO command is later issued to resume printing, VSE/POWER prints those pages again with their current contents. The command allows you to check and adjust forms alignment.

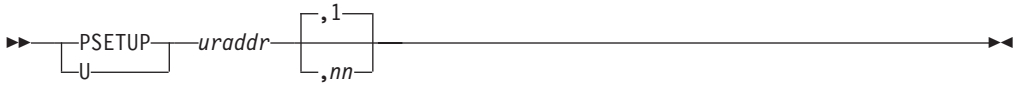
Necessary adjustments are to be made manually while VSE/POWER prints the setup pages.

Use the PSETUP command only after having received message 1Q40A or 1QA5A.

The formats of the PSETUP command are as follows:

- Format 1: Checking the Form Alignment on a Physical Printer
- Format 2: Checking the Form Alignment on a Printer Controlled by a Device Driving System

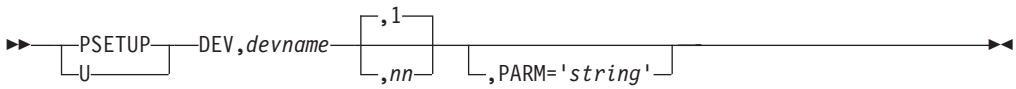
Format 1: Checking the Form Alignment on a Physical Printer



uraddr
For *uraddr*, specify the cuu 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.

Format 2: Checking the Form Alignment on a Printer Controlled by a Device Driving System



DEV
Code this operand as shown. It indicates to a subsystem, for example PSF, 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.

1|nn
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', 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
```

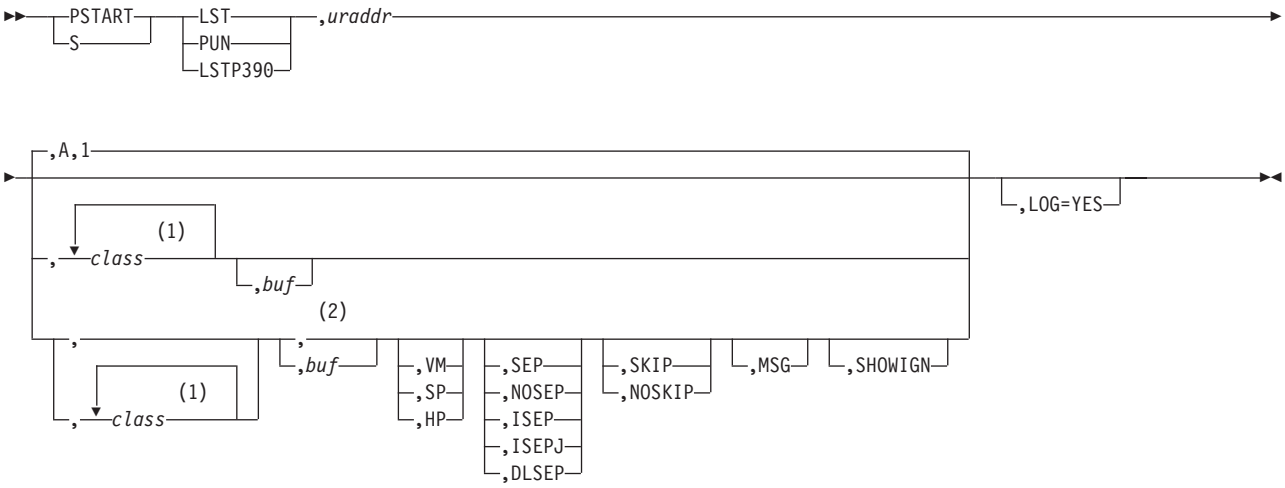
PSTART: Starting a Task or a Partition

The command can be used to bring a VSE/POWER task or a static partition under control of VSE/POWER or to start a task dispatching trace.

The formats of the PSTART command are as follows:

- Format 1: Processing Disk-Spooled Output
- Format 2: Processing Output Stored on Tape
- Format 3: Spooling Input from Card Reader to Disk
- Format 4: Spooling Input from SYSIN Tape to Disk
- Format 5: Switching Spooling between Card Reader and Diskette
- Format 6: Spooling Input from Diskette to Disk
- Format 7: Starting a Device Controlled by a Device Driving System
- Format 8: Putting a Static Partition Under VSE/POWER Control
- Format 9: Starting the Task Dispatch Trace
- Formats 10 and 11: Activating the Remote Job Entry Function
- Formats 12 and 13: Activating the PNET Function
- Format 14: Starting the PNET Console Trace on the Console or Printer
- Format 15: Enabling Dumping of the PNET or RJE Line Trace Area

Format 1: Processing Disk-Spooled Output



Notes:

1 You may specify up to four classes. If nothing is specified, the default class is A.

2 You must specify at least one additional operand.

Use this format to process output that has been spooled to disk:

- Start processing of spooled output under control of a VSE system running under VM. In this case, specify the 'VM' operand. The following list provides an overview of the VSE/POWER- VM interface:

  Given command: PSTART LST/PUN, cuu

  – Addresses only queue entries without a specific TO-userid

- When processing of a queue entry begins, **no** 'CP SPOOL cuu ...' command is passed to VM. Thus, a previous CP SPOOL command entered by the operator remains in effect.

Given command: PSTART LST/PUN,cuu,,,VM

- Either addresses queue entries without a specific TO-userid and passes the 'CP SPOOL cuu OFF' command to VM at begin of the queue entry resetting any previous CP SPOOL cuu operands that may have been set by an operator. Hence, any operator-provided CP SPOOL command is nullified.
- Or addresses queue entries with a specific To-userid (except remote user ID R001-R250) and passes the 'CP SPOOL cuu TO USERID ...' command to VM at begin of the queue entry. Likewise, any operator provided CP SPOOL command is nullified.
- Passes the 'CP CLOSE cuu ...' command to VM at end of every queue entry.

- Specify if you do or don't want separator pages on all output, regardless of the JSEP value associated with the queue entry.

Notes:

1. The processing of special record formats has changed since VSE/POWER Version 2.2.. Output queue entries with a record format of SCS, BMS, CPDS, 3270 and ESC are not eligible for local processing as offered by this command format. Such entries are to be printed by device driving systems which use the spool-access support to retrieve these entries, for example the IBM Print Support Facilities (PSF). To identify special format queue entries, issue a PDISPLAY command with option FULL=YES . The resulting display shows the record format, for example 'RF=CPDS'.

   Instead only queue entries with standard machine code format ('RF' not displayed) or entries with special record format ASA ('RF=ASA') are eligible for local printing.
2. For VM or VM/ESA-mode supervisors available before ESA Version 2, the 'CP CLOSE cuu ...' command is passed to VM at end of every queue entry.

LST

Code this if spooled list output is to be processed.

LSTP390

Code this if spooled list output is to be processed on a printer which is emulated by a P390 Device Manager.

The 'P390' suffix enables the device manager to recognize the boundary of list queue entries: a CCW with OPCODE X'FF' and the jobname and jobnumber as data are appended to all printed output.

The following CCW is used to pass the jobname and jobnumber:

```
** X'FF', data address,SLI, X'FF',13 **
The length comprises 8 bytes for the jobname and 5 bytes for
the jobnumber in decimal.
```

If the I/O with channel command X'FF' fails, message 1RF1A is issued and the task is terminated. The failing LST queue entry remains in the queue with its original disposition.

The VM and SP operands will be rejected with message 1R52I PSTART OPERANDS ARE INCONSISTENT. All other operands of the PSTART command can be used without any restriction.

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 cuu address of the output device which VSE/POWER is to use.

Under VM, uraddr is the address of the virtual output device that is to be used.

A | class
For *class* specify the desired class or classes if the default class cannot be used.

VSE/POWER accepts up to four letters of the alphabet, or numerals 0 through 9. Every letter or numeral represent a class of output that VSE/POWER is to write to the named device.

If both a VM and a local writer task (with and without VM option) run in parallel on your system, they must not be started for the same output class.

1 | buf
The operand is valid only for a list task (PSTART LST,...).

For *buf*, specify the number of output and input buffers that the task is to use. You can specify:

```
1 = one output buffer and one input buffer, or
2 = two output buffers and one input buffer, or
D = two output buffers and two input buffers
```

Default is one output buffer.

Note: The buffer storage is used by a task only while the task is active.

VM
Specify VM to tell VSE/POWER that the LST or PUN device 'uraddr' is a virtual device owned by VM. This causes VSE/POWER to generate VM-CP commands as follows:

```
SP cuu TO user_id|SYSTEM CLASS c COPY nnn FORM ffff
CL cuu NAME jobname jobnumber
```

or

```
CL cuu DIST distcode NAME jobname jobnumber
```

if DIST was specified in an * $$ LST/PUN statement.

Note: For entries with a copy count greater than 1, only one physical copy is passed to VM.

When a TO user ID is specified, but the user ID is not defined in the VM-CP directory, VSE/POWER informs you by message *1QAAI* and requeues the entry in HOLD disposition (H or L).

If VM is specified, local and non-local entries (with blank and non-blank To user-ID) are selected. If VM is not specified, only local entries are selected for printing or punching.

See also the SET CCW01=NO option on page 478, which is applicable only for list tasks.

SP
Specify SP to tell VSE/POWER that the LST or PUN virtual device is owned by VM and used in spooled mode. This option lowers the task's dispatching priority versus execution processor tasks and will avoid that printing of a large output into VM monopolizes the VSE system and halts job execution.

See also the SET CCW01=NO option on page 478, which is applicable only for list tasks.

HP
Specify HP to force the LST or PUN task to the same dispatching priority as a spool-access support task. This will increase the performance of the physical writer tasks and should only be used if a significant performance degradation of the physical printers or punches has been detected due to many active spool-access users, or due to intensive networking activities.

SEP
Code this operand if you want separator pages or cards for every 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 JSEP=1.

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.

ISEP
Code this operand to have 'identical' separator pages for printing list output. 'Identical' means that the last separator page is identical to all previous ones which contain detailed job information subsequent to the jobname/jobnumber printed in block letters. For standard layout of separator pages, in particular the last one, refer to "Separator Pages - Layout and Control" on page 181. This operand is accepted for PUN tasks but has no effect.

ISEPJ
Code this operand to have 'identical' separator pages (see description of the ISEP operand) whose number is chosen according to the JSEP= value specified in the * $$ LST statement thereby overwriting the JSEP=value of the POWER macro. This operand is accepted for PUN tasks but has no effect.

DLSEP
Code this operand if you want to drop the 'last-one' separator page for list output. In addition, via the autostart statement SET **LINE, you can define the size of the separator page by specifying how many perforation lines (normally, when DLSEP is not specified, 8 identical lines, starting with '****') are printed. If SET **LINE=n is not specified, a default of 4 perforation lines are printed. Refer to "Drop 'Last-One' and Print Shorter Separator Page" on page 184 and to "SET: Setting VSE/POWER Startup Control Values" on page 473. VSE/POWER takes the number of separator pages 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 without an additional 'last-one'.

This operand is accepted for PUN tasks but has no effect.

SKIP
Code this operand if you want VSE/POWER to insert a skip-to-channel 1 preceding the printed job output. VSE/POWER inserts a skip-to-channel 1 if

- A write command occurs and no skip-to-channel 1 was found earlier, while control commands preceding the first write are simply ignored
- 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 statement SET, if present.

If you omit the SKIP|NOSKIP operand, VSE/POWER takes the SKIP=YES option from the autostart statement SET (if present) as default.

MSG
Code this operand if you want VSE/POWER to display FORM RQ ffff on the display panel of a printer (for example, 4248).

SHOWIGN
When a writer task is started with this option, any usually ignored CCW op-code is printed or punched as hexadecimal data and is followed by up to 25 characters of original user data. This provides additional debugging help when trying to identify ignored records in the original data stream as presented by an IPW$$DD dump.

For locating these records easily within the output data stream, every line is marked with 10 heading and 10 trailing '-' characters.

Example:

```
---------- IGNORED: X'0P' ABCDEFGHIJKLMNOPQRSTUVWXY ----------
```

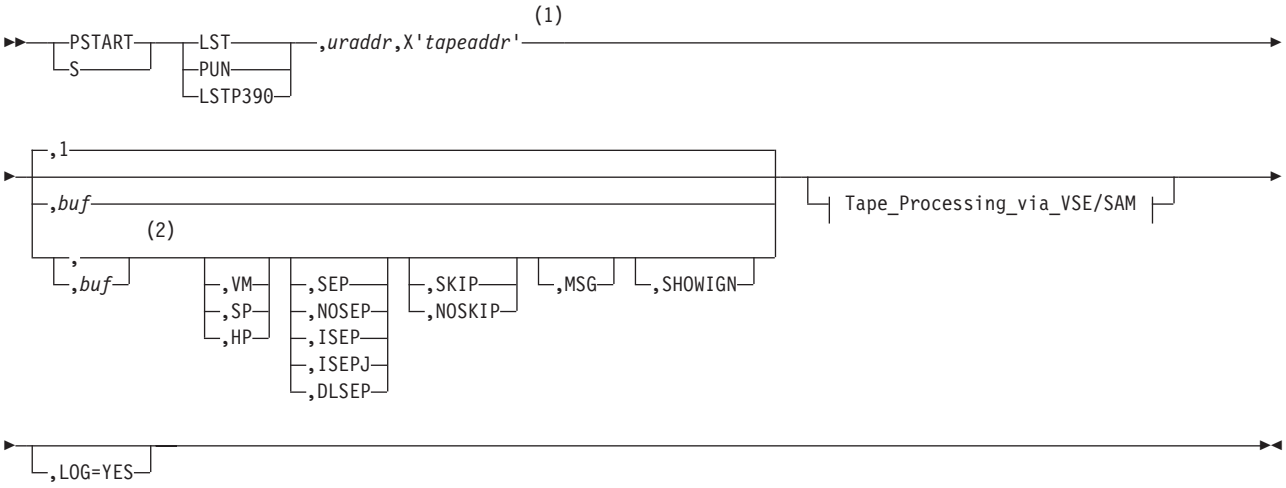
For more details on ignored records, refer also to the SET IGNREC autostart statement on page 476 or to "If Output Records are Ignored" on page 115.

LOG=YES
Specify this operand if every processed queue entry should be logged on the console by message 1Q8BKI (see example below). If you do not want this message to appear on the console, but be available in the hardcopy file for control purposes, then use the PVARY MSG,1Q8KI,NOCONS command.

```
1Q8KI  OUTPUT myjob nnnnn PASSED TO PRINTER/PUNCH DEVICE 00E FOR VM USER 'userid'
```

Format 2: Processing Output Stored on Tape

For further information on this command, please refer to "Tape Processing with VSE/POWER" on page 151.



Tape_Processing_via_VSE/SAM:

Notes:

1 Output cannot be printed/punched from an IBM 9346 or 3592 tape. Any attempt is rejected by message 1Q7FI.

2 You must specify at least one additional operand.

Use this format to process output that has been spooled to tape:

- Start processing of spooled output under control of a VSE system running under VM. In this case, specify the 'VM' operand.
- Specify if you do or don't want separator pages on all output, regardless of the JSEP value associated with the queue entry.

LST
Code this if spooled list output is to be processed.

LSTP390
Code this if spooled list output is to be processed on a printer which is emulated by a P390 Device Manager.

The 'P390' suffix enables the Device Manager to recognize the boundary of list queue entries: a CCW with OPCODE X'FF' and the jobname and jobnumber as data are appended to all printed output.

The following CCW is used to pass the jobname and jobnumber:

```
** X'FF', data address,SLI, X'FF',13 **
The length comprises 8 bytes for the jobname and 5 bytes for
the jobnumber in decimal.
```

If the I/O with channel command X'FF' fails, message 1RF1A is issued and the task is terminated. The failing LST queue entry remains in the queue with its original disposition.

The VM and SP operands will be rejected with message 1R52I PSTART OPERANDS ARE INCONSISTENT. All other operands of the PSTART command can be used without any restriction.

PUN
Code this if spooled punch output is to be processed.

uraddr
For *uraddr*, specify the cuu address of the output device which VSE/POWER is to use.

tapeaddr
For *tapeaddr*, specify the cuu address of the tape drive on which the spool or offloaded tape is mounted. Here the X specification is required.

1|buf
The operand is valid only for a list task (PSTART LST,...).

For *buf*, specify the number of output buffers the task is to use. Specify

```
1 =  one output buffer
2 =  two output buffers.
```

Default is one output buffer.

Note: The buffer storage is used by a task only while the task is active.

VM
Specify VM to tell VSE/POWER that the LST or PUN device 'uraddr' is a virtual device owned by VM. This causes VSE/POWER to generate VM-CP commands as follows:

```
SP cuu TO user_id|SYSTEM CLASS c COPY nnn FORM ffff
CL cuu NAME jobname jobnumber
```

or

```
CL cuu DIST distcode NAME jobname jobnumber
```

if DIST was specified in an * $$ LST/PUN statement.

When a TO user ID is specified but the user ID is not defined in the VM-CP directory, VSE/POWER informs you by message *1QAAI* and requeues the entry in HOLD disposition (H or L).

See also the SET CCW01=NO option on page 478, which is applicable on for list tasks.

SP
Specify SP to tell VSE/POWER that the LST or PUN virtual device is owned by VM and used in spooled mode. This option lowers the task's dispatching priority versus execution processor tasks and will avoid that printing of a large output into VM monopolizes the VSE system and halts job execution.

See also the SET CCW01=NO option on page 478, which is applicable on for list tasks.

HP
Specify HP to force the LST or PUN task to the same dispatching priority as a

spool-access support task. This will increase the performance of the physical writer tasks and should only be used if a significant performance degradation of the physical printers or punches has been detected due to many active spool-access users, or due to intensive networking activities.

SEP
Code this operand if you want separator pages or cards for every 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 JSEP=1.

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 JSEP value from the associated * $$ LST or * $$ PUN statement is used.

ISEP
Code this operand to have 'identical' separator pages for printing list output. 'Identical' means that the last separator page is identical to all previous ones which contain detailed job information subsequent to the jobname/jobnumber printed in block letters. For standard layout of separator pages, in particular the last one, refer to "Separator Pages - Layout and Control" on page 181. This operand is accepted for PUN tasks but has no effect.

ISEPJ
Code this operand to have 'identical' separator pages (see description of the ISEP operand) whose number is chosen according to the JSEP= value specified in the * $$ LST statement thereby overwriting the JSEP=value of the POWER macro. This operand is accepted for PUN tasks but has no effect.

DLSEP
Code this operand if you want to drop the 'last-one' separator page for list output. In addition, via the autostart statement SET **LINE, you can define the size of the separator page by specifying how many perforation lines (normally, when DLSEP is not specified, 8 identical lines, starting with '****') are printed. If SET **LINE=n is not specified, a default of 4 perforation lines is printed. Refer to "Drop 'Last-One' and Print Shorter Separator Page" on page 184 and to "SET: Setting VSE/POWER Startup Control Values" on page 473. VSE/POWER takes the number of separator pages 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 without an additional 'last-one'. This operand is accepted for PUN tasks but has no effect.

SKIP
Code this operand if you want VSE/POWER to insert a skip-to-channel 1 preceding the printed job output. VSE/POWER inserts a skip-to-channel 1 if

- A write command occurs and no skip-to-channel 1 was found earlier while control commands preceding the first write are simply ignored
- 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. It overrides a SKIP=YES specification in the autostart statement SET. If you omit the SKIP|NOSKIP operand, VSE/POWER takes the SKIP=YES option from the autostart statement SET (if present) as default.

MSG
Code this operand if you want VSE/POWER to display FORM RQ ffff on the display panel of a printer (for example, 4248).

SHOWIGN
When a writer task is started with this option, any usually ignored CCW op-code is printed or punched as hexadecimal data and is followed by up to 25 characters of original user data. This provides additional debugging help when trying to identify ignored records in the original data stream as presented by an IPW$$DD dump.

For locating these records easily within the output data stream, every line is marked with 10 heading and 10 trailing '-' characters.

Example:

```
---------- IGNORED: X'0P' ABCDEFGHIJKLMNOPQRSTUVWXY ----------
```

For more details on ignored records, refer also to the SET IGNREC autostart statement on page 481 or to "If Output Records are Ignored" on page 115.

TLBL=tfilename
For tfilename specify one to seven characters of a tape label `// TLBL filename...`. It is assumed that the `// TLBL` statement is available to the VSE/POWER partition. For the syntax of the TLBL statement, see *z/VSE System Control Statements*.

If specified and the LTAPE= operand is not specified, then LTAPE=YES is assumed.

For more information about using the TLBL statement see the description of LTAPE.

LTAPE=YES|NO
If specified, VSE/SAM is used during OPEN, CLOSE and end-of-volume (EOV) tape processing.

YES
Indicates VSE/SAM labeled tape processing.

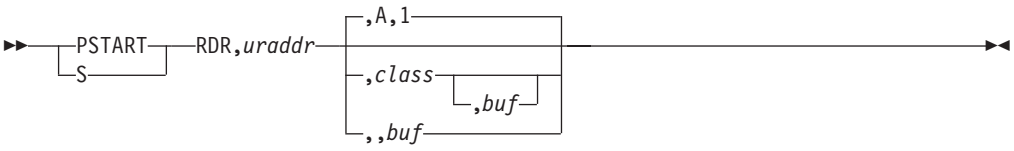
NO
Indicates VSE/SAM unlabeled tape processing.

If TLBL= is specified in addition to LTAPE=NO, an unlabeled tape will still be processed using VSE/SAM. However, VSE/POWER will pass the tfilename value to VSE/SAM as for labeled tape processing. This operand value is normally needed only by Tape Management Systems which might use the `// TLBL` statement to indicate tape information, such as tape density. The TLBL= value then applies to all unlabeled tapes produced.

LOG=YES
Specify this operand if every processed queue entry should be logged on the console by message 1QBKI (see example below). If you do not want this message to appear on the console, but be available in the hardcopy file for control purposes, then use the PVARY MSG,1QBKI,NOCONS command..

```
1Q8KI  OUTPUT myjob nnnnn PASSED TO PRINTER 00E FOR VM USER userid
```

Format 3: Spooling Input from Card Reader to Disk



RDR
Code this operand as shown.

uraddr
For *uraddr*, specify the cuu address of the device 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 = BG, 1 = F1, 2 = F2, and so on).

VSE/POWER assigns the specified class to all jobs submitted without a class specification in their * $$ JOB statement, except when an * $$ CTL statement is in effect.

1 | buf
For *buf*, specify the number of 4KB input buffers that the task is to use. You can specify one of the following:

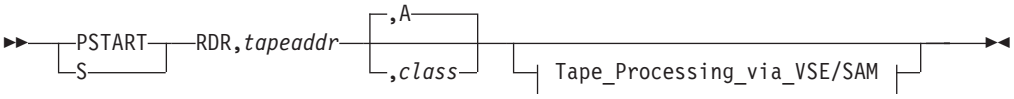
```
1 = one input buffer
2 = two input buffers
```

If you omit the operand, VSE/POWER starts the task with one input buffer available.

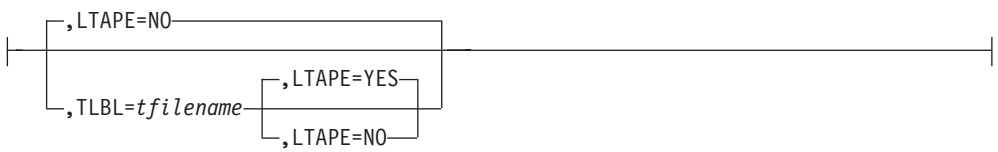
Note: Storage for the buffers is used by a task only as long as the task is "active", that means not in 1Q34I or 1Q35A state.

Format 4: Spooling Input from SYSIN Tape to Disk

For further information on this command, please refer to "Tape Processing with VSE/POWER" on page 151.



Tape_Processing_via_VSE/SAM:



RDR
Code this operand as shown.

tapeaddr
For *tapeaddr*, specify the cuu address of the drive where the input volume is mounted.

A | class
For *class*, the rules as given above for the RDR operand set apply.

TLBL=tfilename
For tfilename specify one to seven characters of a tape label `// TLBL filename...`. It is assumed that the `// TLBL` statement is available to the VSE/POWER partition. For the syntax of the TLBL statement, see *z/VSE System Control Statements*.

If specified and the LTAPE= operand is not specified, then LTAPE=YES is assumed.

For more information about using the TLBL statement see the description of LTAPE.

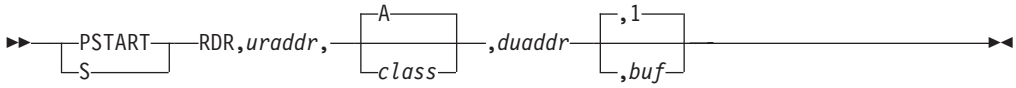
LTAPE=YES | NO
If specified, VSE/SAM is used during OPEN, CLOSE and end-of-volume (EOV) tape processing.

YES
Indicates VSE/SAM labeled tape processing.

NO
Indicates VSE/SAM unlabeled tape processing.

If TLBL= is specified in addition to LTAPE=NO, an unlabeled tape will still be processed using VSE/SAM. However, VSE/POWER will pass the tfilename value to VSE/SAM as for labeled tape processing. This operand value is normally needed only by Tape Management Systems which might use the // TLBL statement to indicate tape information, such as tape density. The TLBL= value then applies to all unlabeled tapes produced.

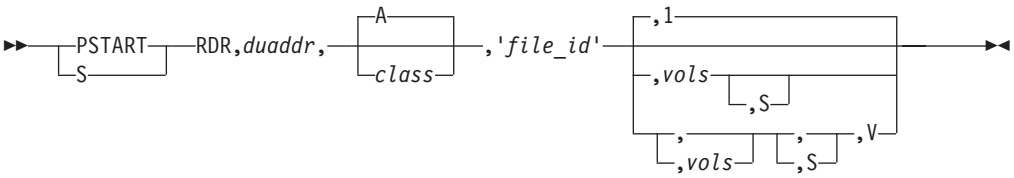
Format 5: Switching Spooling between Card Reader and Diskette



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 cuu address of the diskette I/O unit to be used.

Format 6: Spooling Input from Diskette to Disk



RDR
Code this operand as shown.

duaddr
For *duaddr*, specify the cuu address of the diskette I/O unit to be used.

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.

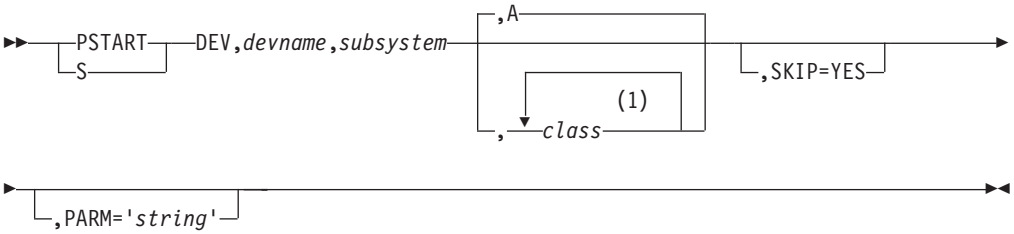
1 | vols
For *vols*, specify how many diskettes may be read. The reader task stops reading input either after it has read the specified number of diskettes or at the end of a diskette whose HDR1 label indicates that this is the last diskette of the file. You can specify any number from 01 to 99. VSE/POWER 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.

Format 7: Starting a Device Controlled by a Device Driving System

Use this format to cause a subsystem, for example PSF, or program in another partition to start a device under its control for processing spooled output.



Notes:

1 You may specify up to four classes.

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 can consist of up to eight alphameric characters. Do not use R000 through R250 as devname, since these are the reserved RJE user IDs, which are rejected by message 1R52I.

Note: If the device name is LOCAL, VSE/POWER returns queue entries destined for local processing or for user ID LOCAL.

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.

SKIP=YES
Code this operand if you want VSE/POWER to insert a skip-to-channel 1 preceding printed output.

Insertion takes place:

- If the output entry has either Machine format op-codes or ASA format op-codes converted to machine format during retrieval
- If a write command occurs but no skip-to-channel 1 was found before, while control commands preceding the first write command are all ignored, or
- If only control commands were found in the output entry.

Note: The SKIP option is also passed to a Device Driving System in the PORDSFLG of the PSTART Device Order. See “Start-Device Order” in *VSE/POWER Application Programming*.

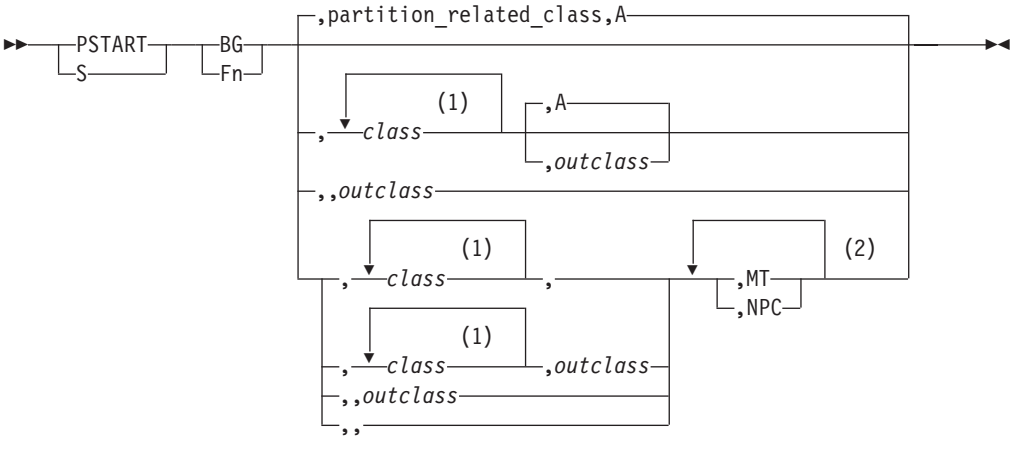
PARM='string'
For 'string', supply a device and subsystem specific parameter string, if this is required. VSE/POWER's requirements regarding the parameter string are:

- 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 ('').

Note: When PARM is used to pass logical destinations (in addition to the *devname*) to the Device Driving System, then one should not use R000 through R250 as the destination, since these are reserved RJE user IDs..

Format 8: Putting a Static Partition Under VSE/POWER Control

If the partition already is controlled by VSE/POWER, this command works like a VSE START command. You get a message if the partition is still active or not supported, or if its priority is not lower than that of the VSE/POWER partition and NPC is not specified.



Notes:

1 You may specify up to four classes.

2 Each option may be specified only once.

BG | Fn
Specify BG to place the background partition under control of VSE/POWER.

For n in Fn, specify the number of the applicable foreground partition.

partition_related_class | class
For *class*, specify the class(es) of the jobs that may be processed in the partition. You can specify up to four classes, every 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. For details, see "VSE/POWER Classes" on page 53.

The order in which you specify the classes is the order in which VSE/POWER processes jobs in the partition. If you omit the operand, VSE/POWER selects processing classes as follows:

```
0 - for BG (partition-related class)
1 - for F1 (partition-related class)
. . . .
9 - for F9 (partition-related class)
A - for FA (not partition-related class)
B - for FB (not partition-related class)
```

A | outclass
For *outclass*, specify the default class (an alphabetic character or numeral 0 through 9), 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 long running program that does not return to z/VSE Job Control at the end of its VSE/POWER reader job, but continues to read and process statements of further

VSE/POWER jobs made dispatchable for the execution class of the 'MT' [14] partition. For this partition, spooled output is not made available at the VSE/POWER job boundaries.

'MT' support causes VSE/POWER to do the following:

- If the user program still resides in the partition and there are no more input records for the partition, only the associated execution reader task is placed into the wait state accompanied by message 1Q34A. The writer tasks are not stopped as in normal partitions. Instead, they accept more output from the partition.
- Output from the partition is made available as a corresponding queue entry only if:
  - An LFCB macro is requested for output to printers with a forms control buffer, or
  - An IPWSEGM or SEGMENT macro is requested for other printers or punched output, or
  - Further processed jobs contain an * $$ LST or a * $$ PUN JECL statement, or
  - The partition is removed from VSE/POWER control by the PSTOP partition-id command.

  In order to keep all collected output together as one queue entry, the initial input job for this partition must include an * $$ LST or an * $$ PUN statement for every spool device that may be used later.

If normal programs (returning to z/VSE Job Control at end of the reader queue entry) are processed in the 'MT' partition, the above described handling of output may be beneficial under certain circumstances.

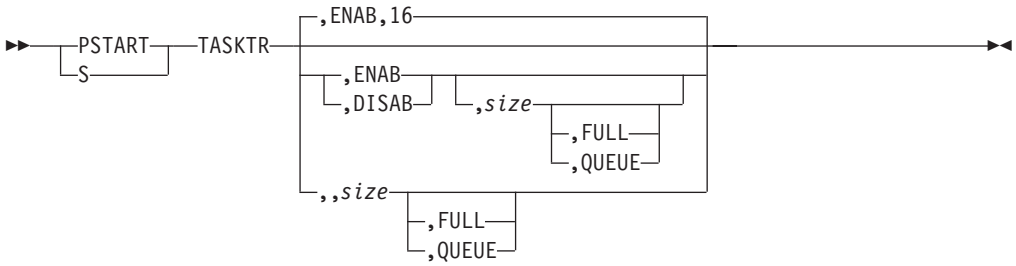
For details on how the 'MT' partition's VSE security authorization may be affected, see "Data Security and Multi-Tasking (MT) Partition" on page 13.

NPC

Bypasses the check that VSE/POWER has the highest dispatching priority.

14. 'MT' stands for 'multi-tasking'. This is a historical term reminding of a certain long running program coded in multi-tasking technique. Amongst other functions, this program was able to send the statements of further VSE/POWER jobs to other networking applications.

Format 9: Starting the Task Dispatch Trace



TASKTR
Causes the VSE/POWER task dispatcher to initiate recording the task status whenever a task is dispatched.

ENAB | DISAB
ENAB causes VSE/POWER to prepare recording and to start tracing immediately. This is the default.

DISAB causes VSE/POWER to prepare recording but to disable the tracing. Use the PVARY TASKTR command to enable tracing of the task status when required.

size
Specifies the `size` of the task trace area in KB. This area is obtained from the SETPFIX part of the VSE/POWER partition. The minimum specification is 2 and the default is 16. The maximum is 512. The trace area must be freed by means of a PSTOP TASKTR command before a trace area of a new size is allocated by VSE/POWER.

Find more detail about the task dispatching trace in "VSE/POWER Task Dispatching Trace" on page 516.

FULL
Causes VSE/POWER to verify the task control block (TCB) and real storage element (BCW) chains at each dispatch cycle for VSE/POWER tasks, in order to identify corruption of these chains as soon as possible. In case of damage, VSE/POWER terminates itself by a program check to swap the debug areas of the running VSE/AF DEBUG trace. The IDUMP taken by the VSE/POWER AB exit routine will include the VSE/AF DEBUG area to provide information about all VSE/AF tasks active between detection of the chain error and the previously successful TCB or BCW chain check (due to 'FULL'). This special IDUMP identifies itself by message 1Q2CI naming PHASE=IPW$$11 or IPW$$12 and TASK=TASKTR FULL (with the address of an artificial TCB located in IPW$$NU).

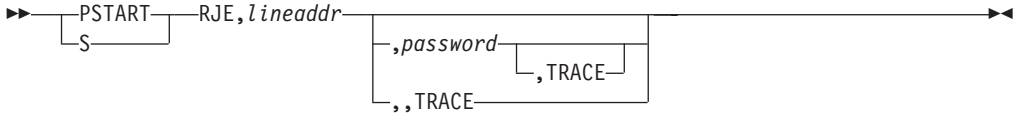
Note: If you use the FULL operand and the VSE/AF Debug Trace is not yet enabled, you are prompted by message 1RF2A to enable the VSE/AF debug trace by the AR command 'DEBUG ON,250K'. Otherwise, the produced IDUMP may not provide enough information to find the culprit.

QUEUE
Causes VSE/POWER to verify the class anchors and previous/next pointers not to point into the free queue record chain - at each dispatch cycle for a VSE/POWER task - to identify the corruption as soon as possible. In case of damage, an IDUMP on fly is taken - identified by message '1Q2JI ..., PHASE=IPW$$13' and spooling continues with further IDUMPs suppressed.

Note: the full QUEUE file trace is activated only for a queue file residing in the VSE/POWER partition Getvis area. The QUEUE option is ignored for a queue file in VIO.

Formats 10 and 11: Activating the Remote Job Entry Function

Use the following format to activate the RJE function on a certain BSC driven line.



Use the following format to activate the RJE function using the available SNA support.

RJE,lineaddr
Specify RJE,lineaddr if an RJE,BSC task is to be started.

For *lineaddr*, specify the cuu address of the telecommunication line that is to be used by the task.

password
specifies 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 operations for the specified telecommunication line recorded in the available telecommunication trace area.

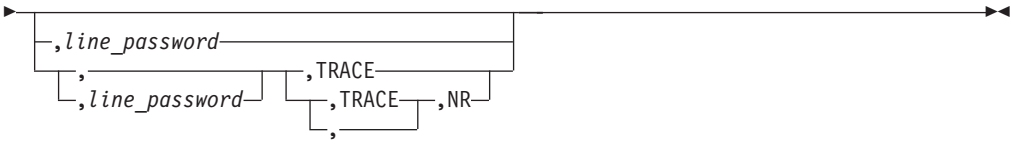
For displaying the location of this trace area, see also "Format 7: Displaying Various Status Information" on page 238.

Note: It is recommended to trace only one telecommunication line (both PNET and RJE) at one time to avoid missed entries in the telecommunication trace area.

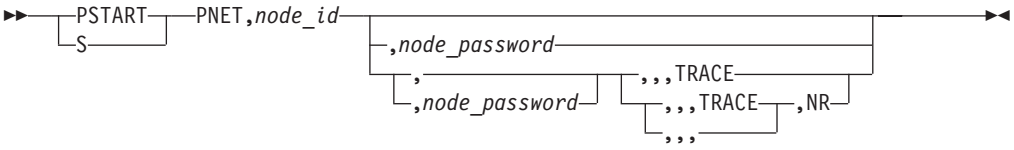
RJE,SNA
Specify RJE,SNA if the VSE/POWER SNA support for Remote Job Entry is to be activated. It supports remote stations generated by a PRMT macro as TYPE=LUT1.

Formats 12 and 13: Starting the Connection to Other Networking Nodes

Use format 12 to start a connection to another node via BSC or virtual CTCA support:



Use format 13 to start a connection to another node via SNA or TCP/IP (SSL) support:



PNET
Code this operand as shown.

node_id
For *node_id*, specify the name of the node to which a connection is to be made. This name must be known to VSE/POWER through the loaded Network Definition Table (NDT).

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 the local node. The password you specify can have up to 8 alphanumeric characters.

lineaddr
For *lineaddr*, specify the cuu address of the telecommunication line or virtual CTCA adapter that VSE/POWER is to use for communicating with the specified node.

Note: VSE/POWER supports only the virtual CTCA under VM.

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 CTCA support. The password you specify can have up to 8 alphanumeric characters.

TRACE
For format 12, specify TRACE if you want the BSC or CTCA line activity or any line problems to be traced in the telecommunication trace area. This causes all I/O operations for the specified node/line to be recorded in the available trace area.

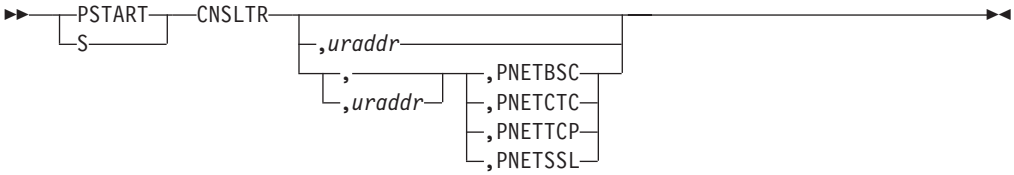
For format 13, specify TRACE if you want the following to be recorded in the telecommunication trace area for the started node:

- if SNA node: all information on VTAM SEND AND RECEIVE operations
- if TCP or SSL node: all information on TCP/IP socket calls (including SEND and RECEIVE) together with the CTCA line activity emulated for this node.

For displaying the location of this trace area, see also “Format 7: Displaying Various Status Information” on page 238.

NR
Specify NR if automatic restart of the networking function is not to be attempted by VSE/POWER.

Format 14: Starting the PNET Console Trace on the Console or Printer



CNSLTR
Initiates tracing of PNET BSC, CTC, TCP, and SSL activity (independent of a specific node) and writes recorded events to the console according to the connection type:

- for all BSC or CTC: line I/O events recording OUT-going and IN-coming NJE control information.
- For TCP and SSL: the emulated CTC I/O events, followed by the TCP/IP related EZASMI macro requests with their return codes.

For details, see "PNET Console Trace" in *VSE/POWER Networking*.

uraddr
For *uraddr*, specify the cuu address of a printer device that VSE/POWER should assign dynamically to its partition for collecting the "console-type" trace information on the named printer instead of on the console. Tracing on a printer is recommended for TCP/SSL nodes because they

- produce more VSE/POWER generated trace messages than BSC or CTC connections
- can be combined with the EZASMI macro interface trace and BSD flow trace of TCP/IP for VSE/ESA. For details, see "PNET Console Trace" in *VSE/POWER Networking*.

Note: The printer assignment cannot be removed by a subsequent PSTART CNSLTR command without the *uraddr* specification. Instead, the PSTOP CNSLTR command must be used for that purpose.

PNETBSC | PNETCTC | PNETTCP | PNETSSL
Specify one of the PNET connection types, if tracing activity should be initiated only for nodes defined in the network definition table as linked through the corresponding discipline.

Notes:

1. The default connection type (if none is specified) is *all* types.
2. With repeated PSTART CNSLTR,,PNETxxx commands, you can accumulate tracing for more than one connection type.
3. PSTOP can be used to stop the trace, but only totally and not by PNET connection type.

Format 15: Enabling Dumping of the PNET or RJE Line Trace Area

DUMPTR
Enables dumping of the telecommunication trace area (see "Format 7: Displaying Various Status Information" on page 238) into the dump sublib of the VSE/POWER partition whenever the area is full, in order to avoid wrapping. The size of this area is defined by the TRACESZ operand (default 24K) of the POWER macro. The area is filled by trace entries of RJE/BSC and all PNET connection types whenever an RJE line or PNET node has been PSTARTed with the TRACE operand.

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-related class 4 does not match the partition ID 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.

S TASKTR,,10
Reserves a work area of 10KB in the real storage part of the partition. Whenever a task gets control, records the task status in this task trace area.

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.

PSTART LST,00E,X'280',TLBL=TLABOUT

Starts a list task to print output from the labeled tape on tape unit 280 with the label 'TLABOUT'.

PSTART CNSLTR,02E,PNETTCP

Initiates recording of PNET Console Trace data to the printer device at address 02E for all PNET/TCP communication events (independent of node name).

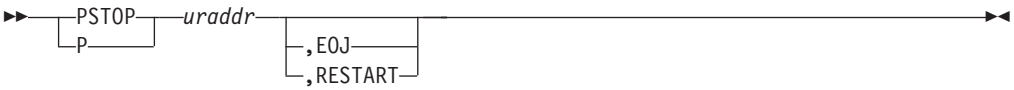
PSTOP: Stopping Tasks or Releasing a Static Partition from VSE/POWER

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
- A task dispatching trace
- Control of VSE/POWER over a static partition

Note: To restart a task after it has been stopped by PSTOP, use PSTART.

The formats of the PSTOP command are as follows:
- Format 1: Stopping Reading from or Writing to an I/O Device
- Format 2: Releasing a Static Partition from VSE/POWER Control
- Format 3: Stopping a Task Dispatching Trace
- Format 4: Stopping Reading from or Writing to Tape
- Format 5: Stopping Output on a Device Controlled by a Device Driving System
- Format 6: Stopping an RJE,BSC Line
- Format 7: Stopping Networking for a Node
- Format 8: Stopping RJE,SNA Activities
- Format 9: Stopping Active Spool-Access-Support Tasks
- Format 10: Stopping PNET Console Tracing or Dumping of Line Trace Area
- Format 11: Releasing a Physical Unit from VSE/POWER Control
- Format 12: Stopping a VSE/POWER Task Pending in Internal Wait
- Format 13: Stopping the TCP/IP Interface Before PEND
- Format 14: Stopping the TCP/IP SSL Interface Before PEND

Format 1: Stopping Reading from or Writing to an I/O Device



uraddr
For *uraddr*, specify the address of the device whose task is to be stopped. By specifying this operand alone, you cause VSE/POWER to do the following:
- For a reader task, it stops further input processing for the affected job. No entry for this job is placed into the input queue.
- For a writer task, it stops 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.

  If a 3800 printer is used, processing restarts with the copy group index being processed when the PSTOP command was issued.

EOJ
For a reader task, processing continues until all input for the job has been spooled. For a writer task, it continues until all printed or punched output for the current 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.

Note: When the task associated with a printer, punch, or tape device cannot be stopped by the PSTOP command, consider to use the PSTOP cuu,FORCE command of Format 12.

Format 2: Releasing a Static Partition from VSE/POWER Control

Specify the ID of the partition 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, the assignments of all devices that already have been spooled are released (for example, SYSIN, SYSLIST, or programmer logical units). Now 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 to release the background partition from control of VSE/POWER.

For *n* in *Fn*, specify the number of the applicable foreground partition.

Note:

1. Specifying the partition-id of a dynamic partition has no effect, since dynamic partitions are stopped after the completion of the single VSE/POWER job they process
2. Using the SET CONFIRM=PSTOP PART autostart statement you can protect your system against inadvertent PSTOP'ing of partitons. For details, see "SET: Setting VSE/POWER Startup Control Values" on page 473.

Format 3: Stopping a Task Dispatching Trace

TASKTR
Stops tracing of task dispatching events. The task trace area is cleared to hex zero and is returned to the storage pool of VSE/POWER.

Format 4: Stopping Reading from or Writing to Tape

tapeaddr
For *tapeaddr*, specify the channel and unit address of the drive that is being used by the affected function.

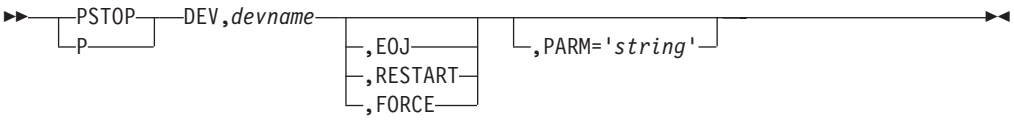
However, a task controlling input from or output to a tape cannot be stopped if this task is waiting for an operator's response. To stop the task, use a 'PGO tapeaddr,CANCEL' command instead.

EOJ
Specify EOJ if the task should stop on completion of the currently processed queue entry.

Note: When the task associated with a printer, punch, or tape device cannot be stopped by this PSTOP command, consider to use PSTOP cuu,FORCE of Format 12.

Format 5: Stopping Output on a Device Controlled by a Device Driving System

Use this format to stop further output for a device owned by a subsystem, for example PSF, or program in another partition.



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
At the end of the currently processed output queue entry, the subsystem stops requesting VSE/POWER for more output to the involved device.

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', 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 ('').

Format 6: Stopping an RJE,BSC Line

How to stop a line is described under "Stopping an RJE,BSC Line" on page 125.

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:

- If an RJE *reader* task is affected, it stops further input processing for the affected job. No entry for this job is placed into the input queue.

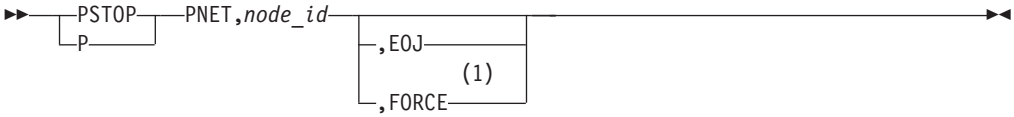
  If an RJE *writer* task is affected. it stops further processing of job output. However, this output is retained in the output queue. When the task is started again, processing of the output starts with the first record.
- It stops the line in accordance with the line protocol.
- It makes an attempt to send a line-stopped message (1R02I) to the connected terminal.

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 this operand is necessary, 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.

Format 7: Stopping Networking for a Node

There is no need for a PSTOP PNET,... command at normal end of VSE/POWER operation. The PEND command initiates the necessary shutdown functions.



Notes:

1 Effective only for PNET BSC.

PNET
Code this operand as shown.

node_id
For *node_id*, specify the name of the node for which networking activities are to be stopped.

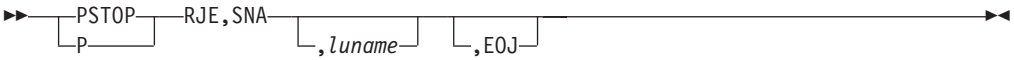
EOJ
Specify EOJ if the PNET task is to continue processing until all activity on the line or for the session has completed to end of job. This means that all transmitter tasks continue with their current transmissions and all receiver tasks continue receiving transmissions until EOJ is reached. It is 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 or FORCE, 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,node_id,... command. This causes VSE/POWER to transmit job or output data from the beginning of the job or output.

FORCE
Specify FORCE for a PNET BSC or TCP or SSL node when neither PSTOP...,node_id,EOJ nor PSTOP...,node_id (immediate) has terminated the connection to node_id correctly. The FORCE option stops the connection at once, in disregard of the NJE line protocol. The effect on active transmitters and receivers is as described for PSTOP...,node_id (immediate).

For PNET SNA or CTC, the FORCE option is ignored.

Format 8: Stopping 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 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 terminals without logging off the entire terminal.

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 terminal, VSE/POWER stops the line immediately.

Format 9: Stopping Active Spool-Access-Support Tasks

SAS
Code this operand as shown.

ALL
To stop all spool-access support tasks.

connect_id
For the *connect_id*, specify the VSE/POWER assigned connection ID of the spool-access support task that is to be stopped. VSE/POWER displays this ID, a number, in response to a "PDISPLAY A,SAS" command. This number is on the display line immediately behind the task name (SAS). For a sample display line, see "Format 8: PDISPLAY A" on page 270.

Format 10: Stopping PNET Console Tracing or Dumping of Line Trace Area

This command format is available for controlling VSE/POWER's trace function dynamically.

CNSLTR
Specify CNSLTR to stop the tracing and recording of all PNET BSC , CTC, and TCP activity events directed to either

- the console, or
- a printer, in case of PSTART CNSLTR,uraddr. As a result, the printer device is unassigned from the VSE/POWER partition.

DUMPTR
Specify DUMPTR to stop writing the contents of the telecommunication trace area (used by the RJE/BSC or PNET function) into the dump sublibrary of the VSE/POWER partition. At the same time, the partially filled trace area is also dumped to the dump sublibrary.

Format 11: Releasing a Physical Unit from VSE/POWER Control

Use this format to release a physical printer, punch, or tape unit from VSE/POWER control if no VSE/POWER task exists which owns the device. The following errors may require to use this format:

- VSE/POWER error
- VSE Supervisor error
- Hardware error

If due to such an error a physical unit has not been released from VSE/POWER control, the VSE/POWER task which once used the physical unit may no longer be active.

Do the following to see if a physical unit requires to be released:

- Use the LISTIO *cuu* command to see if the physical unit *cuu* is still assigned to the VSE/POWER partition
- Use the PDISPLAY A command to verify that no VSE/POWER task owns this *cuu*.

Note: This command is meaningful only if no VSE/POWER task exists which uses the specified physical unit. If a task exists which still owns this physical unit, a PSTOP cuu,UNASSGN command is rejected by message 1QABI, and you should use PSTOP Format 1, Format 4, or Format 12 instead.

```
►►—PSTOP cuu,UNASSGN———————————————————————————————————►◄
```

cuu
For *cuu*, specify the address of the device which has to be released from VSE/POWER control. You may specify a *cuu* only for a print, a punch, or tape device.

'*cuu*' may be the address of a device which has been used previously as *cuu* or *cuu2* in one of the following commands or statements:
- PSTART LST,*cuu* to start a local print task
- PSTART PUN,*cuu* to start a local punch task
- POFFLOAD op1,op2,*cuu* to write or read queue entries to or from tape
- PDISPLAY ...,TAPE=*cuu* to display queue entries residing on tape
- PSTART LST,*cuu*,X'*cuu2*' to start a local print task with input from tape
- PSTART PUN,*cuu*,X'*cuu2*' to start a local punch task with input from tape
- * $$ LST ..,DISP=T,TADDR=*cuu2* to spool list output to tape
- * $$ PUN ..,DISP=T,TADDR=*cuu2* to spool punch output to tape

If a tape device has to be released which has been addressed by *cuu2* above, you have to specify the *cuu2* for the tape within the PSTOP command and **not** the *cuu*. For example:

If you have started a local print task (called tape writer task) using the command

```
PSTART LST,00E,X'282'
```

you may force the release of the tape device using the command:

```
PSTOP 282,UNASSGN
```

UNASSGN
Code this operand as shown.

Format 12: Stopping a VSE/POWER Task Pending in Internal Wait

Use the FORCE operand to stop a VSE/POWER task which cannot be stopped by the Format 1 or Format 4 command, but which must be stopped immediately in order to get a physical unit released from VSE/POWER control, such as a printer, punch, or tape device.

Using the FORCE operand, a VSE/POWER task is stopped if it is in one of the following states:
- Waiting for virtual storage
- Waiting for real storage
- Waiting for an operator reply
- Waiting for I/O completion for a printer, punch, tape device

- Waiting for locked resource

If the task is not in one of the above states, VSE/POWER rejects the command by message *1QAEI* and the task may not be stopped.

This command is meaningful for the following VSE/POWER tasks:
1. A local print task started by PSTART LST,*cuu*
2. A local punch task started by PSTART PUN,*cuu*
3. A POFFLOAD task started by POFFLOAD op1,op2,*cuu*
4. A print status task started by PDISPLAY ...,TAPE=*cuu*
5. A local print task started by PSTART LST,*cuu*,*X'cuu2'*, input from tape
6. A local punch task started by PSTART PUN,*cuu*,*X'cuu2'*, input from tape
7. An execution writer task, which is started when a job is executed in a partition, and which uses a tape device as specified in a JECL statement:

   * $$ LST ..,DISP=T,TADDR=*cuu2*

   * $$ PUN ..,DISP=T,TADDR=*cuu2*

If a task which is associated with a *cuu2* tape should be stopped, specify the *cuu2* for the tape within the PSTOP command. For example, if you have started a local print task (tape writer) by:

```
PSTART LST,00E,X'282'
```

you may stop the task and force a release of the tape device using the command:

```
PSTOP 282,FORCE.
```


cuu

For *cuu*, specify the address of the device associated with the task to be stopped. You may specify a *cuu* only for a printer, a punch, or tape device.

If the task is waiting for a 'locked resource', it is possible that during the termination of the task the waiting-for-locked-resource condition occurs again. You may detect such a situation by issuing the PDISPLAY TASKS command: under the heading STATE(RX) you see L(address) for the appropriate task, where L stands for 'locked' and address stands for the resource being waited for. If the PSTOP command is entered once more with the operand FORCE, the task will be stopped then without releasing storage resources, and its accessed queue entry remains in 'execution state' (DISP=*). The queue entry can no longer be accessed by any task until VSE/POWER recovery is done, which can be triggered by a PEND FORCE termination.

FORCE

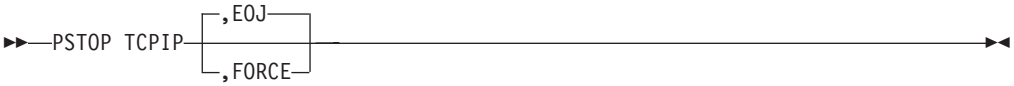
Code this operand as shown.

Format 13: Stopping the TCP/IP Interface Before PEND

When the TCP/IP (TD) subtask has been activated during loading of the Network Definition Table (see "Loading an NDT (Format 1)" on page 319), there may be reasons to terminate this task, and thereby the TCP/IP interface, prematurely before VSE/POWER PEND time, such as to:

- free resources (a VSE subtask, polling events, etc.)
- load a new NDT with a Local Port Number (see PNODE PORT= in "PNODE Generation Macro for Networking Support" on page 81) different from the one previously used (see also message 1RA5I, RC=0005)

- or because the TCP/IP subtask has ceased serving TCP nodes with TCP/IP socket requests.



EOJ
Specify EOJ when the TCP/IP interface, as represented by the TD subtask, should be terminated only when all TCP nodes have been stopped successfully by the PSTOP PNET command.

FORCE
Specify FORCE when the TCP/IP interface, as represented by the TD subtask, should be terminated abruptly by a subtask CANCEL request, which issues the message

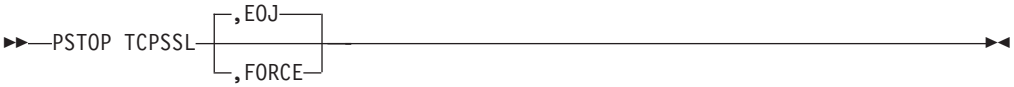
```
1Q2CI...CANCEL BY PSTOP
```

followed by an IDUMP of the VSE/POWER partition. Use the FORCE option only when the TD subtask does not react to PSTOP TCPIP,EOJ, although all PNET TCP nodes have been stopped.

Format 14: Stopping the TCP/IP SSL Interface Before PEND

When the TCP/IP SSL (SD) subtask has been activated during loading of the Network Definition Table (see "Loading an NDT (Format 1)" on page 319), there may be reasons to terminate this task, and thereby the TCP/IP SSL interface, prematurely before VSE/POWER PEND time, such as to:

- free resources (a VSE subtask, polling events, etc.)
- load an NDT with Local Port attributes (see PNODE SPORT=, SECTYPE=, KEYRING=, and DNAME= in "PNODE Generation Macro for Networking Support" on page 81) different from those previously used (see also message 1RA5I, RC=0006/7/8/9).
- or because the SD subtask has ceased serving SSL nodes with TCP/IP socket requests.



EOJ
Specify EOJ when the TCP/IP SSL interface, as represented by the SD subtask, should be terminated only when all SSL nodes have been stopped successfully by the PSTOP PNET command.

FORCE
Specify FORCE when the TCP/IP SSL interface, as represented by the SD subtask, should be terminated abruptly by a subtask CANCEL request, which issues the message

```
1Q2CI...CANCEL BY PSTOP
```

followed by an IDUMP of the VSE/POWER partition. Use the FORCE option only when the SD subtask does not react to PSTOP TCPSSL,EOJ, although all PNET SSL nodes have been stopped.

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 is 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 starts 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) is transmitted.

PVARY: Dis-/Enabling Exits, Task Trace, Dynamic Classes or Messages

A user-written exit routine is automatically enabled when it is loaded. You use the DISAB parameter to temporarily suspend the exit routine or the task trace and enable it again later on.

The DYNC parameter of this command applies to dynamic partition scheduling. It allows to enable or disable selected classes of the active Dynamic Class Table for dynamic scheduling of jobs. The command is accepted only when an active Dynamic Class Table exists at all. For further information refer to "Halt and Resume Dynamic Partition Allocation" on page 138

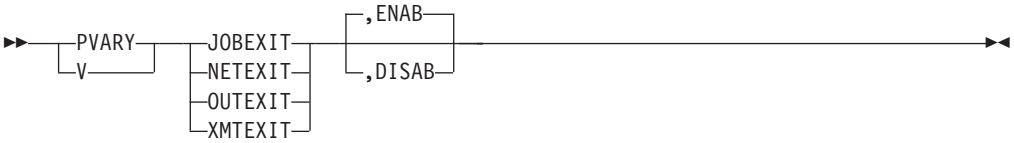
The MSG parameter of this command applies to reducing the console traffic caused by VSE/POWER messages. It allows to disable or to re-enable a selected message for displaying on the console, without affecting the display on the hardcopy file. The command may be requested by the control operator or by an authorized Spool Access Control command user.

The MAXSAS parameter of this command applies to specifying the maximum number of concurrent Spool Access Support tasks. The default threshold value is 250 tasks. The command may only be requested by the central operator.

The formats of the PVARY command are as follows:

- Format 1: Varying Exit Routines
- Format 2: Varying the Task Trace
- Format 3: Varying Dynamic Classes
- Format 4: Varying VSE/POWER messages
- Format 5: Varying the threshold value of maximum SAS tasks

Format 1: Varying Exit Routines



JOBEXIT | NETEXIT | OUTEXIT | XMTEXIT
These operands specify the type of user exit.

ENAB | DISAB
Specify ENAB to reactivate an exit routine previously disabled. ENAB is the default. Specify DISAB to deactivate a currently enabled exit routine.

Format 2: Varying the Task Trace

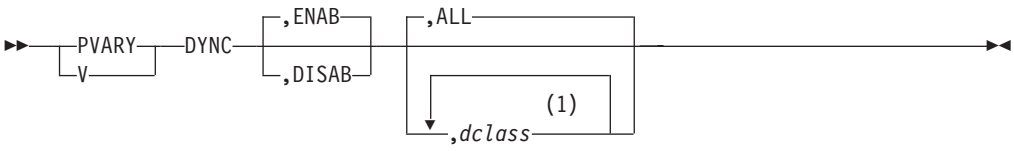
TASKTR
This operand specifies the running or previously disabled task trace.

ENAB | DISAB
Specify ENAB to reactivate the previously disabled task trace. ENAB is the default. Specify DISAB to deactivate the currently enabled task trace.

Note: Disabling places the 'TASKTRC DISABLED' constant into the task trace area at the current (next to be used) entry position. When the trace is enabled again, the constant will be overwritten.

For examples and usage, see "VSE/POWER Task Dispatching Trace" on page 516.

Format 3: Varying Dynamic Classes



Notes:

1 You can specify up to 12 alphabetic classes as defined in the active Dynamic Class Table.

DYNC
Specify this operand to enable or disable dynamic classes in the active Dynamic Class Table.

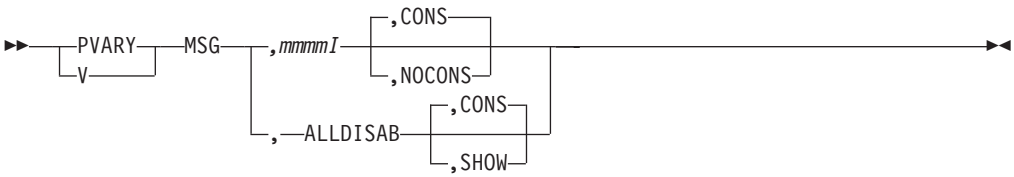
ENAB | DISAB
Specify ENAB to enable classes of the active Dynamic Class Table for processing of jobs. 'Invalid' flagged classes are not considered for enabling. ENAB is the default.

Specify DISAB to deactivate classes of the active Dynamic Class Table for processing of jobs.

ALL
Specify ALL to enable or disable all classes of the active Dynamic Class Table. ALL is the default.

dclass
Specify up to 12 alphabetic classes as defined in the active Dynamic Class Table. The specified class(es) are considered for enabling or disabling. For each specified class which cannot be located in the active dynamic class, message `1Q6EI` is issued.

Format 4: Varying VSE/POWER Messages



MSG
Specify this operand to disable or re-enable a VSE/POWER message for displaying on the console operator console.

mmmmI
is the Information ('I') message prefix consisting of 5 characters including the Information type code 'I' as the last character. Only VSE/POWER Information messages are accepted. Other message types such as Action ('A') or Decision ('D') are not accepted.

CONS
Specify this operand to permit a selected message to be displayed on the console and in the console hardcopy at the same time. This is the default.

Specify CONS in combination with ALLDISAB to select **all** messages previously **disabled** and to enable them again for console display.

NOCONS
Specify this operand to restrict the selected message to recording on the console hardcopy only. NOCONS cannot be specified together with ALLDISAB.

ALLDISAB
Specify this operand to select **all** messages previously **disabled** for console display and:
- either SHOW them by message 1Q8AI or
- enable them again for CONSole display as recorded per selected message by:

```
1Q8HI MESSAGE mmmmI BEEN DISABLED, NOW ENABLED FOR CONSOLE
```

SHOW
Specify this operand in combination with ALLDISAB to select all messages previously disabled for console display by PVARY MSG,...,NOCONS and request the following display line per selected message:

```
1Q8HI MESSAGE mmmmI IS DISABLED
```

Note:

1. Command is accepted from the central operator console or an authorized system administrator.
2. A message is suppressed only on the console, but is always recorded in the hardcopy file.
3. If the message is destined for a user or a remote system, the message is sent to its destination.
4. Some important messages are not issued by the general message routine, for example messages issued
   - during the initialization of VSE/POWER or
   - as response to the VSE/POWER command PDISPLAY STATUS.

   Therefore these messages are displayed on the console, although the PVARY command has been used to suppress them on the console.
5. PVARY 'NOCONS' information is kept in module IPW$$MM (loaded into the VSE/POWER partition) and is valid only for the current VSE/POWER session. At next VSE/POWER startup all messages are established again for "CONS". PVARY MSG commands may be placed into the VSE/POWER autostart procedure, for example after the PSTART partition commands.
6. If module IPW$$MM is reloaded (for example due to maintenance) in a running VSE/POWER system by the PLOAD PHASE command, previously established "NOCONS" information is lost.
7. Messages disabled for the console are still offered to 'console exits', that means for example REXX console operators or OEM products can still trigger processes based on those messages.

Format 5: Varying the Maximum Number of SAS Tasks

MAXSAS
Specify this operand to modify the maximum number of concurrent Spool Access Support (SAS-GET/PUT/CTL/GCM) tasks. The VSE/POWER default is 250. Both the maximum limit and the current number of SAS tasks can be found in the PDISPLAY STATUS report The tasks themselves and their XPCC-application are identified by the PDISPLAY A,SAS command.

The threshold of 250 is to protect against failing applications that CONNECT to SYSPWR without DISCONNecting and thereby force VSE/POWER to keep unused tasks in storage. Eventually so many SAS tasks are created, which gradually consume all partition Getvis or SETPFIX LIMT storage. The value of 250 may either be decreased by the PVARY command to provide a lower threshold. Or it must be increased, if your system really has such a high Spool Access traffic. When the threshold has been reached, additional XPCC connections are terminated with an XPCC DISCPRG request (IJBXRETCD/IJBXREAS = X'19/40') together with the VSE/POWER PXPRETCD/PXPFBKCD = X'10/07' equating to PXPRCNOC/PXP10MST. At the same time the central operator is alerted by action message 1Q3JA, which is deleted from the console as soon as the existing threshold value has been modified by the PVARY MAXSAS command

nnnn
Is the maximum number of concurrent Spool Access Support (SAS). The nimimum is 50 and the maximum is 2000. The VSE/POWER default is 250.

Examples of the PVARY Command

PVARY XMTEXIT,DISAB
Disables the currently loaded and enabled PNET transmitter exit from being given control by VSE/POWER.

V DYNC,C,G,X
Enables the dynamic classes C,G,X of the active Dynamic Class Table for dynamic scheduling of jobs in corresponding dynamic partitions.

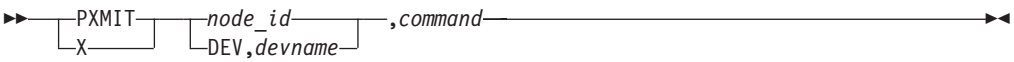
V MSG,1Q47I,NOCONS
Disables job logging message 1Q47I for the console display, while recording in the hardcopy file continues as normally.

PXMIT: Routing a Command

This command routes another command either to another node within the network or to a Device Driving 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.

Format of the Command



node_id
This operand is valid only if the PNET function is active. For *node_id*, 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, for example PSF, 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 130 characters.

Note: Other older VSE/POWER releases may accept a maximum command of 72 bytes only, and may truncate the received command.

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, see the publication *VSE/POWER Networking*.

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 (130 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,ALTER LOGDEST=3244
Passes the command ALTER LOGDEST=3244 to the subsystem that controls the device named PLOT1.

Chapter 6. JECL Statements

In a VSE system, job processing is controlled by the Job Control program. Job Control Language (JCL) statements pass control information to this program. In a job, such statements have the prefix '//'.

In addition, VSE/POWER and its Job Entry Control program manages the collection and scheduling of job input statements for partitions and the spooling of output from partitions. Job Entry Control Language (JECL) statements in a job pass such information to VSE/POWER. These statements begin with the prefix '* $$'.

Advantages of JECL Statements

To process a job in a VSE/POWER-controlled partition, you can submit a job with its normal z/VSE Job Control delimiters // JOB and /&:

```
// JOB
 . . .
 . . .
 . . .
/&
```

When such a job passes through the VSE/POWER read-in functions, it is added to the reader queue. It may be selected for processing by a VSE/POWER-controlled partition with default VSE/POWER scheduling attributes such as class, disposition, and priority.

You are advised, however, to surround your VSE jobs by the * $$ JOB and * $$ EOJ delimiter statements of the VSE/POWER Job Entry Control Language (JECL). These statements indicate the boundaries of your VSE/POWER job and specify when and on which system your job is to be processed. In addition, you may place other JECL statements into your VSE/POWER job. An * $$ LST statement, for example, can be used to specify when collected output of your job is to be made available as a list queue entry and how it should be handled. Thus, a job may then look like:

```
* $$ JOB JOBN=POWERJOB,...
* $$ LST CLASS=5,...
// JOB VSEJOB
 . . .
 . . .
/&
* $$ EOJ
```

For these statements, the VSE/POWER read-in function creates a reader queue entry named POWERJOB, with processing attributes as specified by the various operands of the * $$ JOB JECL statement.

Overview of JECL Statements

Table 14 on page 398 lists all VSE/POWER JECL statements in alphabetical order, summarizes their purpose, and shows when each JECL statement is processed by VSE/POWER. Starting on page 409 a complete description of the JECL statements and their operands also are given in alphabetical order.

Table 14. Overview of JECL Statements

JECL Statement	Function	Processing Time
* $$ CTL	Assigns a new default input class to VSE/POWER jobs.	Read-in [3]
* $$ DATA	Inserts data from the reader queue into a library member after this member was retrieved for inclusion into a VSE/POWER job.	Execution
* $$ EOJ	Indicates the end of a VSE/POWER job.	Read-in [3]
* $$ FLS	Indicates that a VSE/POWER job should be terminated by internal flushing.	Execution
* $$ JOB	Indicates the beginning of a VSE/POWER job and specifies the routing of jobs, output, and notify messages.	Read-in [3]
* $$ LST	Provides handling information for printer output; routes list output to a node.	Execution
* $$ PUN	Provides handling information for punched output; routes punch output to a node.	Execution
* $$ RDR	Inserts a diskette file into the input stream.	Read-in [3]
* $$ SLI	Inserts data from an accessible library.	Execution
* $$ /*	Indicates the end of a VSE job step (used with the SLI statement only) [1]	Execution
* $$ /&	Indicates the end of a VSE job (used with the SLI statement only) [1]	Execution
/*$SLI	Indicates end of input data for an SLI member [2]	Execution

[1] For details, see "* $$ SLI: Including a Library Member into a Job Stream" on page 463.

[2] For details, see "* $$ DATA: Inserting Data into an SLI Library Member" on page 410.

[3] In exceptional cases, these statements process at execution time. For details see, "Handling Read-in JECL Statements at Execution Time" on page 403.

The above figure identifies which of the JECL statements are processed at **read-in** time. Then they are interpreted and are transformed into:

*** $$ JOB →**
The Internal Job Header Record (JHR) of the reader queue entry

*** $$ RDR →**
Any group of statements read in from a diskette and placed into the reader queue entry

*** $$ EOJ →**
The Internal Job Trailer Record (JTR) of the reader queue entry

*** $$ CTL →**
Switching of the existing VSE/POWER reader input class.

All other statements are passed without transformation into the reader queue entry. They are interpreted only when a VSE/POWER-controlled partition starts to process the job, namely at the **execution** time.

Sources of JECL Statements

VSE/POWER jobs, or any series of statements that contain JECL statements, are accepted from the following sources:

- Reading for spooling as a queue entry from card reader, tape, diskette, RJE, PNET, and from the spool-access support as used by the Interactive Interface.
- Extending of queue entries during their interpretation for execution by data from VSE/ICCF or VSE/AF libraries using the * $$ SLI statement.
- Passing of * $$ LST/PUN/JOB statements to VSE/POWER during the execution of a user program using the IPWSEGM or SEGMENT macro call facility. For details on these macros, see *VSE/POWER Application Programming*.

Job Streams Without JECL Delimiters

When job streams that use only JCL delimiters (// JOB, /&) are submitted, the VSE/POWER read-in function switches to JCL mode, converts all // JOB statements to uppercase, and creates reader queue entries according to the following rules:

- All statements found between // JOB and /& are grouped into a VSE/POWER reader (RDR) queue entry named as the JCL JOB name
- If the // JOB statement has no name, a RDR entry named AUTONAME is created. For suppression of AUTONAME jobs, see "SET: Setting VSE/POWER Startup Control Values" on page 473.
- If no /& end delimiter but another // JOB statement is found, the previous job ends prematurely. The aborted job is added to the reader queue, and collection of data for the next job starts.
- If after a /& no // JOB statement but other data types follow, the previous job is added to the reader queue. The subsequent statements form an AUTONAME RDR entry until another /& or // JOB statement is encountered.
- If input reading via a reader device (00C, for example) ends without a /& statement, message

  ```
  1Q35A  JOB END INDICATION MISSING ON 00C, SUBMIT EITHER REST OF JOB OR '/&' STATEMENT
  ```

  warns the operator to complete the input stream with the missing statement. Then a reader queue entry can be created.

The submitted VSE jobs may, of course, contain the * $$ RDR and * $$ CTL JECL statements. Moreover, they may contain those jobs interpreted at execution time (see Table 14 on page 398).

The following figure illustrates the VSE/POWER read-in rules for input streams delimited by JCL only:

```
z/VSE Job Stream                    Comments

// JOB ONE                 RDR-queue entry ONE started for class A.
 . . .
// EXEC JOBSTEPA
/*
/&                         Terminates the queue entry.

// JOB TWO                 RDR queue entry TWO started for class A.
 . . .                     No * $$ JOB or * $$ EOJ required for
 . . .                     * $$ LST or * $$ PUN statements.
 . . .
 . . .
// EXEC JOBSTEPB
 . . .
* $$ PUN CLASS=X
 . . .
// EXEC JOBSTEPC
/*
/&                         Terminates the queue entry.

* $$ CTL CLASS=B           Default input class set to B.

// JOB EIGHT               RDR queue entry EIGHT started for class B.
 . . .
// EXEC JOBSTEPL
/*                         VSE/POWER generates the missing /& statement.

// JOB NINE                RDR queue entry NINE started for class B,
* $$ LST FNO=8X1           premature end of previous queue entry.
 . . .
// EXEC JOBSTEPM
* $$ LST JSEP=2,RBS=100
 . . .
// EXEC JOBSTEPN
/*
/&                         Terminates the queue entry.

// EXEC JOBSTEPO           RDR queue entry AUTONAME started for class B,
// EXEC JOBSTEPP           since no // JOB statement found.
/*
/&                         Terminates the queue entry.

// EXEC JOBSTEPQ           RDR queue entry AUTONAME started for class B.
/*
/&                         Terminates the queue entry.
```

Figure 65. Job Stream Examples Delimited by VSE JCL Statements Only

Note: When a queue entry has been terminated and been added to the queue, and the next job has not yet started, VSE/POWER is said to be **on job boundary**.

It is not recommended to submit VSE JCL-only job streams because of the above mentioned boundaries for building of VSE/POWER RDR jobs.

Job Streams with JECL Delimiters

The JECL delimiters * $$ JOB and * $$ EOJ can be inserted into the VSE job stream to denote the user-intended VSE/POWER job boundaries. Then the VSE/POWER read-in function switches to the JECL-mode, converts all JECL statements and VSE // JOB JCL statements to uppercase, and creates reader queue entries according to the following rules:

- All statements found between * $$ JOB and * $$ EOJ are grouped into one VSE/POWER RDR queue entry, which receives the name specified by the JNM operand of the * $$ JOB statement.
- If the JNM operand is omitted, a RDR queue entry called AUTONAME is created. For suppression of AUTONAME jobs, see "SET: Setting VSE/POWER Startup Control Values" on page 473.
- If no * $$ EOJ delimiter but another * $$ JOB statement is found, the previous job ends prematurely. It is added to the reader queue, and collection of data for the next job starts.
- If an * $$ EOJ statement is not followed by a * $$ JOB statement but followed by other data types, the read-in function switches from JECL to JCL mode. Then the rules for "Job Streams Without JECL Delimiters" on page 399 apply until another * $$ JOB statement is found and the JECL mode is re-entered again.
- If input reading via a card reader device 00C, for example, ends without a * $$ EOJ statement, the message

  ```
  1Q35A  JOB END INDICATION MISSING ON 00C, SUBMIT EITHER REST OF JOB OR '* $$ EOJ' STATEMENT
  ```

  alerts the operator to complete the input stream by the missing statement before the currently accumulated data can be added as a reader queue entry.

The submitted VSE/POWER job may, of course, be supplemented by the * $$ RDR and * $$ CTL read-in time JECL statements, and by all statements interpreted not before execution time (Table 14 on page 398).

The following figure illustrate the processing rules valid at VSE/POWER read-in time for input streams delimited by VSE/POWER JECL statements:

```
z/VSE Job Stream                    Comments

* $$ JOB JNM=THREE       RDR queue entry THREE started for class A.
// JOB THREE
   . . .
// EXEC JOBSTEPD
/*
/&
* $$ EOJ                 Terminates the queue entry.

* $$ CTL CLASS=B         Default class changed to B.

* $$ JNM=FOUR            RDR queue entry FOUR started for class B.
// JOB FOUR
   . . .
// EXEC JOBSTEPE
   . . .
/*
// EXEC JOBSTEPF         No * $$ EOJ required if
/*                       VSE/POWER job is followed
/&                       by * $$ JOB statement.

* $$ JOB JNM=FIVE        Terminates the queue entry prematurely,
// JOB FIVE              RDR queue entry FIVE started for class B,
                         No * $$ EOJ required.
   . . .
   . . .
// EXEC JOBSTEPG
   . . .
   . . .
/*
/&

* $$ JOB JNM=SIX         Terminates the queue entry prematurely,
// JOB SIX               RDR queue entry SIX started for class B.
   . . .
// EXEC JOBSTEPH         Multiple VSE jobs in one VSE/POWER job
/*                       * $$ JOB  and * $$ EOJ are required.
/&
// JOB SEVEN
// EXEC JOBSTEPI
/*
/&
* $$ EOJ                 Terminates the queue entry.
```

Figure 66. Examples of Job Streams Delimited by VSE/POWER JECL Statements

In all the examples shown in Figure 66, any z/VSE job that has been started by '// JOB' in a VSE/POWER job is terminated by '/&' in the same VSE/POWER job. This is the recommended structure.

In the following input stream example, however, the z/VSE job 'EIGHT' spans more than one VSE/POWER job and ends in VSE/POWER job 'TEN'.

```
* $$ JOB JNM=EIGHT
// JOB EIGHT
// EXEC PGM8
   . . .
* $$ EOJ
* $$ JOB JNM=NINE
// EXEC PGM9
   . . .
* $$ EOJ
* $$ JOB JNM=TEN
// EXEC PGM10
   . . .
/&
* $$ EOJ
```

If PGM8 is cancelled, z/VSE Job Control in a static partition will bypass all statements till /& (see also "Processing Sequence of Jobs" on page 134). Thus, it is not recommended to construct such job streams.

Handling Read-in JECL Statements at Execution Time

As shown in Table 14 on page 398, the JOB/EOJ/RDR/CTL JECL statements are interpreted at VSE/POWER read-in time (which means the job submission time from a programmer's or operator's view). Thus, they are physically removed from the created queue entry and they are no longer present when the job is selected for execution processing. However, there are three exceptions.

1. A non-PNET node has sent incomplete VSE/POWER jobs (for more details see section "Change of a Job's Characteristics" in *VSE/POWER Networking*.
2. A VSE/POWER job produced through the DISP=I facility provides the above JECL statements
3. The input stream of a writer-only partition contains the above statements (for details, see "JECL Influence on Writer-Only Partitions" on page 406).

In case 1 and 2, if the above JECL statements are still encountered at execution time, the read-in time JECL statements are processed as follows:

- * $$ CTL/RDR statements are ignored and handled as comments.
- * $$ EOJ statements are ignored and handled as comments, unless the z/VSE Access Control function is active. It will cause the job security values to be reset, including propagated values. The propagated values will become active again if a following * $$ JOB statement (without the SEC operand) is processed in the same job.
- * $$ JOB statements take effect for the following operands:

```
JNM=...., (used as name for output entries to create and updates
           the RDR-queue entry temporarily for the period of
           execution)
BLDG=....,
BTRNC=....,  (not effective for DISP=I)
DEPT=....,
ECHO=....,
ECHOU=....,
LDEST=....,
NETACCT=....,
NTFY=....,
PDEST=....,
PROGR=....,
PWD=....,
ROOM=....,
```

```
SEC=....,
SECAC=....,
SYSID=....,
UINF=....,
```

All other operands are checked for the next delimiter, but they are ignored.

Names and Numbers of Jobs and their Output

As mentioned in the previous sections, if JECL job delimiter statements are used (* $$ JOB and * $$ EOJ), VSE/POWER uses the job name specified in the * $$ JOB statement to log the job. If only JCL delimiters are used, VSE/POWER uses the name specified in the // JOB statement. For further details on **names** of read-in jobs, see "Job Streams Without JECL Delimiters" on page 399 and "Job Streams with JECL Delimiters" on page 401.

VSE/POWER assigns a number to every job that enters its queues. This enables VSE/POWER to distinguish between jobs for which identical names are specified. To control the processing of a job, your operator must use the job's name and, possibly, the assigned number in the applicable command.

Once a job has been logged by VSE/POWER, it is referred to as a VSE/POWER job or as a queue entry.

Any output produced by a job is assigned the same name as its RDR queue entry, unless the JNM operand of the * $$ LST/PUN statement has been specified to overwrite the default values.

The very first list (LST) and the very first punch (PUN) output from a VSE/POWER job have the same number as the input job, even if list/punch output is created for more than one spooled list/punch device. Any subsequent LST or PUN output, however, is a separate queue entry and has a unique job number. This aids in processing of VSE/POWER jobs.

With count-driven segmentation, the first 127 segments have the same number as the first segment itself. Each following group of 127 segments obtains a new job number. For accounting purposes, all list or punch account records contain the job number of the reader queue entry file in the ACNUMB field. For details on the account records refer to *VSE/POWER Application Programming*.

Hiding JECL Statements at Read-in Time

To catalog, for example, a VSE/POWER job stream in a VSE/AF library using a VSE/POWER controlled partition, an alternate JECL prefix '..$$' must be used instead of the normal * $$. All JECL statements of the job stream not to be interpreted at read-in time by VSE/POWER (see "Overview of JECL Statements" on page 397) must be coded with this alternate prefix. Then, they are treated as common user data.

At job execution time (for example, at catalog time) VSE/POWER replaces the alternate prefix by the normal * $$ before passing the job for processing, provided that the autostart statement SET SJECL=YES specifies that the substitution is desired. See "SET: Setting VSE/POWER Startup Control Values" on page 473 for details.

Note: If a VSE/AF library member which includes JECL statements is cataloged as described above and is executed by // EXEC PROC=procname, the JECL

statements are treated as comment. But if the member is included into a jobstreamby using the SLI statement, the JECL statements are processed by VSE/POWER as desired.

Uppercase Conversion of Characters in JECL/JCL Statements

Characters in JECL statements and the VSE // JOB JCL statement can be entered in uppercase or lowercase. Generally, lowercase letters are converted to uppercase. For example, when you enter a lowercase 'a', VSE/POWER converts this letter to a capital 'A'. VSE/POWER uses the OR operation, which converts characters with a hexadecimal 40 (X'40') value. Thus, X'20' is converted to X'60' , X'81' to X'C1', and so forth.

If special characters are part of the JECL statement (for example, in the UINF operand), they are altered as well. In some national languages, however, the conversion of a particular special character can generate an error by displaying non-meaningful characters on a console or a printer. For example, the special character '~' (tilde) has a hexadecimal representation of A1 according to standard EBCDIC format. This is converted to the hexadecimal value of E1 which represents no meaningful value in standard EBCDIC format. Some national languages may have similar special characters, which may lead to the same problem of not having the original character displayed.

Note: Because problems might arise when displaying non-printable characters on a console or printer, it is strictly recommended to use only hexadecimal values that, after conversion, represent printable characters.

In general, the conversion takes place for **all** JECL statements at job read-in time. This means that a job is:

- Read in by a locally attached reader or an RJE station
- Submitted by a spool-access support user (for example, from the Interactive Interface)
- Received from the network (converts only * $$ JOB statements).

In exceptional cases, however, the conversion does **not** take place. This is true if the job enters the system as follows:

- Received via the network from non-PNET nodes and the first record is not a * $$ JOB statement
- Has been produced using the DISP=I statement
- Is processed by a writer-only partition.

Correction of JECL Errors

If any of the * $$ JOB/EOJ/CTL/RDR JECL statements interpreted at **read-in** time is found in error, VSE/POWER displays the statement on the central operator console. This is followed by message 1Q37I, pointing to the incorrect column of the statement. Moreover, VSE/POWER:

- Ignores the statement, if it is an * $$ CTL statement.
- Adds the job with disposition HOLD to the reader queue, if it is a * $$ JOB statement.
- Flushes the job with message 1Q90I, if it is an * $$ RDR statement.

Any JECL statement which VSE/POWER finds in error during **execution** time can be corrected when the job is running. The failing JECL statement is reflected on the

central operator console. It is followed by message 1Q51I, pointing to the incorrect operand. Message 1R33D prompts for immediate correction.

The operator then may do one of the following:

- Correct the statement.

  The operator enters the correct JECL statement. The fully 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.

Notes:

1. Whenever the failing JECL statement contains either a PWD or a SEC operand, then any text following the '=' is suppressed, including other operands.
2. Whenever operator correction for message 1R33D cannot be done immediately, consider generating predefined responses to incorrect JECL statements using the SET 1R33D=FLUSH|IGNORE autostart statement (see "SET: Setting VSE/POWER Startup Control Values" on page 473).

JECL Influence on Writer-Only Partitions

A writer-only partition uses a non-VSE/POWER spooled SYSIN reader device from a card reader, tape, or disk. "Placing a Static Partition Under the Control of VSE/POWER" on page 107 describes how to define a static partition as a 'writer-only' partition. Jobs entering this partition are not spooled by VSE/POWER. Therefore, VSE/POWER JECL statements are not interpreted by the VSE/POWER read-in function. As a circumvention, VSE/POWER intercepts all statements written to SYSLOG that start with '* $$'. This is either done by z/VSE Job Control when it finds a comment statement starting with '* ' or by a user program doing its own write request to the console (SYSLOG). Then VSE/POWER can recognize:

1. * $$ JOB and * $$ EOJ statements to establish logical boundaries for VSE/POWER jobs.
2. * $$ LST and * $$ PUN statements to define attributes of spooled output and boundaries for output queue entries.

Control by Limited JECL

When you set up jobs that are **not** enclosed by * $$ JOB and * $$ EOJ JECL statements, note that:

1. If no * $$ LST/PUN statement is present, then all output for the spooled devices is dropped without warning !
2. If an * $$ LST/PUN statement is provided, then spooling of output starts with the attributes defined by the operands. When the next * $$ LST/PUN statement is encountered, the accumulated output entry is made available in the corresponding VSE/POWER queue with a name of

   AUTONAME
   If the JNM= operand has not been used

jobname
If the JNM='jobname' operand has been specified in the * $$ LST/PUN statement

Full JECL Control

In order to avoid the above unpredictable results, use the * $$ JOB and * $$ EOJ statements to denote the logical VSE/POWER job boundaries. This implies also that complete VSE jobs (from // JOB to /&) should be contained in the VSE/POWER job. Then, the following rules are applicable:

1. Because VSE/POWER is already at execution time when the * $$ JOB statement is encountered, only a limited number of the statement operands take effect for any later spooled output:

```
JNM=...., (becomes default output entry name)
BLDG=....,
DEPT=....,
ECHO=....,
ECHOU=....,
LDEST=....,
NETACCT=....,
NTFY=....,
PDEST=....,
PROGR=....,
PWD=....,  (effective for output of job)
ROOM=....,
SEC=....,
SECAC=....,
SYSID=....,  (effective for output of job)
UINF=....,
```

   All other operands are parsed for the next delimiter, but are ignored.
2. When a * $$ EOJ statement is encountered:
   - Output is made available in the corresponding VSE/POWER queue
   - VSE security authorization will be reset by z/VSE Job Control.

   Note: If another * $$ JOB statement is found, the writer-only partition does not imply the end of the previous VSE/POWER job. Instead an explicit * $$ EOJ statement is required!
3. Data-driven segmentation by additional * $$ LST/PUN statements with unlimited support for all their operands is possible too.
4. Other types of JECL statements are ignored as comments, namely:

```
* $$ RDR
* $$ CTL
* $$ FLS
* $$ SLI
* $$ DATA
```

5. Although only JECL statements start with the '* $$' prefix, which identifies them also as Job Control comment statements, they are *never* logged on SYSLST for security reasons.

General Writer-Only Attributes

The following points highlight the differences between the normal spooled partitions and the writer-only partitions:

1. The job logging message 1Q47I is issued whenever an * $$ JOB statement is encountered. The next available VSE/POWER job number is used, even when no job in the reader queue is present. This job number is passed to the next output entry, which will be created.
2. Continuation of an * $$ JOB statement is rejected by message 1Q49I.

3. PFLUSH'ing a writer-only partition is rejected by message 1R61I.
4. No waiting-for-work message 1Q34I is issued.
5. A PDISPLAY A(ctive) identifies the SYSLOG device being intercepted by the execution reader task, instead of the SYSIN device as with normal spooled partitions. For example, for a BG writer-only partition using SYSLOG device address 009 and for a normal partition F3 using SYSIN=00C when both partitions are waiting for work, the display command shows:

```
1R48I   BG,009,,
1R48I   F3,00C,3,      INACTIVE,
```

6. In a writer-only partition, JECL statements contained in a user program SYSIPT data stream are not written to SYSLOG because z/VSE Job Control is not active at that time. Hence, they are not intercepted by VSE/POWER and are considered as data. This allows library members (for example, for SLI inclusion) containing JECL statements to be cataloged entirely.

Coding Rules for JECL Statements Including Continuation

For the symbols used to show the format of JECL statements, see Chapter 1, "Understanding Syntax Diagrams," on page 1.

These are the fields of a JECL statement:

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 operands

- Operands can start in any column after the operation code, but you must use at least one single blank space to separate the first operand from the operation code.
- A comma separates one operand from another operand. This comma must be coded as shown in the syntax of the JECL statements.
- A blank ends the operand field. Therefore, operands cannot contain embedded blanks.
- Keyword operands consist of a keyword, followed by an equal sign, followed by a specified 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.
- Parentheses that can enclose several optional values may be omitted when there is only one such value specified.
- Specifying a user ID as operand value: for details, see "Coding Rules for VSE/POWER Commands" on page 193.
- Specifying *alphameric* characters as operand value: for details, see "Coding Rules for VSE/POWER Commands" on page 193.
- Specifying a blank character within the operand: this is accepted only for the UINF keyword operand, for example UINF='FIRST WEEK', when the specification is enclosed in single quotes.
- For keyword operands, do not code a comma in front of the first 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 every continuation statement must contain the * $$ prefix.
  4. An operand field continued must contain at least one operand or a comma.

  Notes:

  1. When the last operand is terminated by a blank, but a column 72 continuation character is provided, then the subsequent statement is expected to be a JECL continuation statement, whose contents is merely interpreted as a VSE/POWER comment.
  2. For processing of a * $$ LST or PUN statement (not submitted by the IPWSEGM macro), the final keyword operand (e.g. XYZ= value) may begin before column 71 and end even after column 72. However any subsequent keyword operand is ignored without a warning message.

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.

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 as a control-statement ID.

Note: VSE/POWER generally converts all JECL statements and the VSE // JOB statements to uppercase characters before spooling.

Positional Format

All JECL statements described contain only keyword operands. The formats reflect the keyword format. For compatibility, a limited number of JECL statements also still support 'positional format.' For a description of these statements, see *VSE/POWER Installation and Operations Guide* Version 2.2, SH12-5329-5.

* $$ CTL: Assigning a Default Class for Execution

The statement allows you to assign a new class to VSE/POWER jobs which are not controlled by JECL or for which no class was specified in the * $$ JOB statement.

The class you specify in this statement overrides the class specified in the PSTART RDR command or the general VSE/POWER default (class A), whichever applies.

An * $$ CTL statement may be placed at any VSE/POWER job boundary. For details see Figure 65 on page 400 and Figure 66 on page 402.

The statement overrides all 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. The statement is interpreted at read-in time.

Format of the Statement

CLASS=A | class
For class, specify the 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, dynamic class, 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 classes, see the description of the operand CLASS=class under "* $$ JOB: Marking the Start of a VSE/POWER Job" on page 413.

If no class is specified, VSE/POWER uses class A.

* $$ DATA: Inserting 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 a * $$ SLI statement. A member so retrieved is referred to as an SLI member.

This statement is interpreted at job execution time.

Format of the Statement

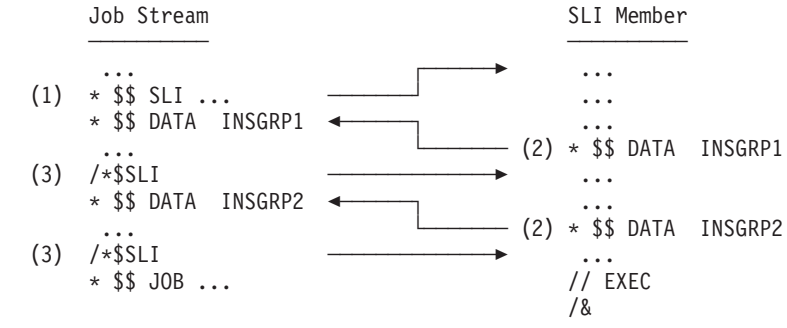
name
For name, specify the name of the corresponding * $$ DATA statement in the retrieved library member where data is to be inserted. The name can be from one to eight alphameric characters long; the first character of this name must be alphabetic.

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 67 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 67. Use of * $$ DATA Statements*

Data to be inserted into an SLI member cannot include any of the following:
- Another * $$ DATA statement. The job is flushed with 1QC0I, RC=0031.
- An * $$ SLI statement. The job is flushed with 1QC0I, RC=0030.

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 * $$ DATA statements may be used together. However, they must appear in the job stream in the same order as the corresponding images appear in the retrieved library member. For more information about update statements, see "* $$ SLI: Including a Library Member into a Job Stream" on page 463.

* $$ EOJ: Marking the End of a VSE/POWER Job

The statement marks the end of a VSE/POWER job. It is generally 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. It is interpreted at read-in time.

Format of the Statement

The statement has no operands.

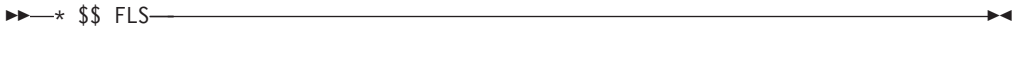
* $$ FLS: Terminating Immediately a VSE/POWER Job

The statement marks any point in a VSE/POWER job where job processing should be terminated unconditionally by internal flushing. It should mainly be placed in job control $CANCEL routines.

The statement is ignored in writer-only partitions. It is interpreted at job execution time and triggers message *1Q4FI*.

For more details see “Interaction with z/VSE Conditional Job Control Language” on page 108.

Format of the Statement



The statement has no operands.

* $$ JOB: Marking 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. For details, see "Names and Numbers of Jobs and their Output" on page 404.

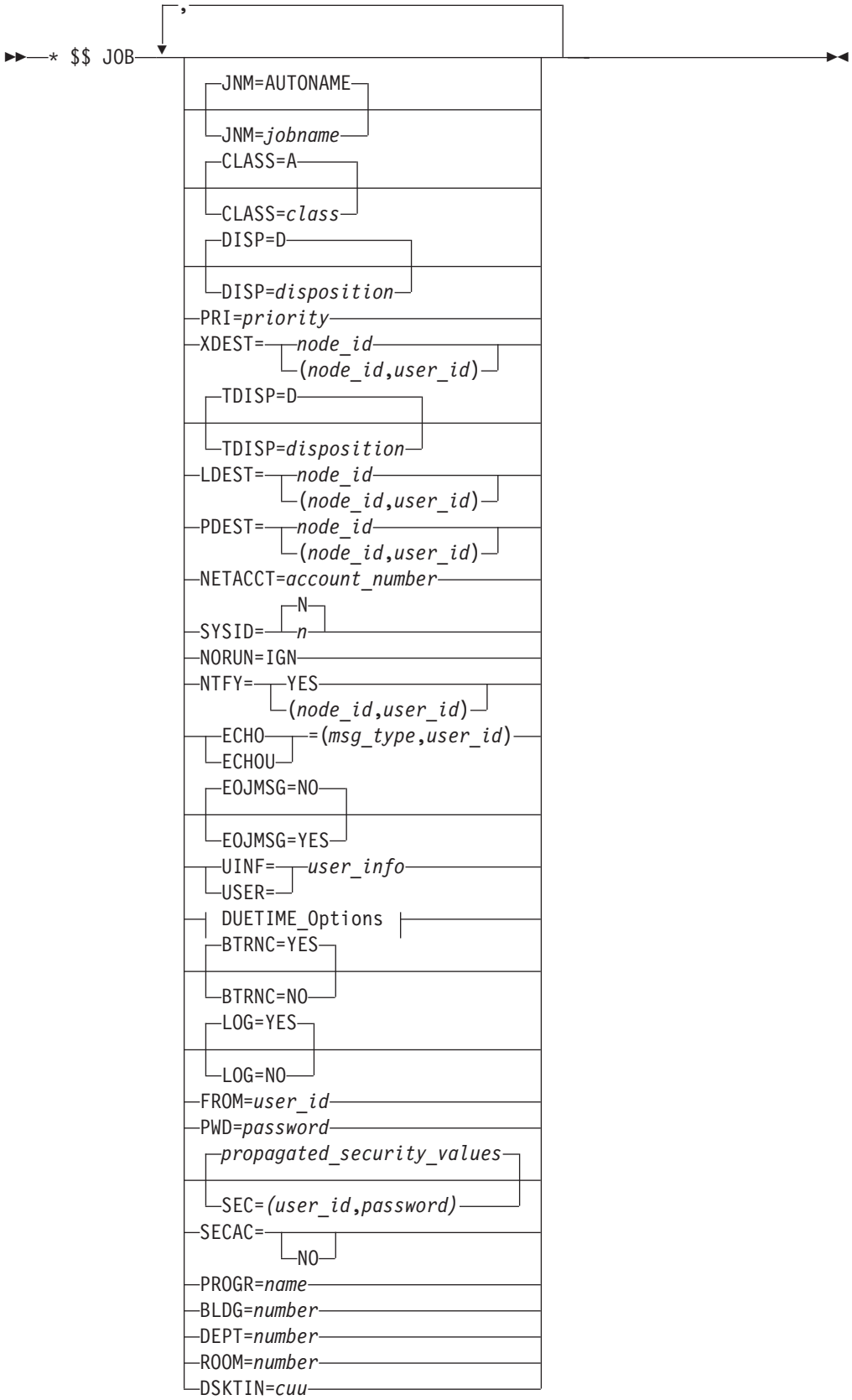
If a VSE/POWER job consists of two or more VSE jobs, these jobs should be delimited by a /& statement at the end. This is because if a VSE job fails, the job stream is ignored up to the next VSE-job delimiter (// JOB or /&) or VSE/POWER-job delimiter (* $$ JOB or * $$ EOJ statement).

If the * $$ JOB statement is found to be in error when the job is read in, message 1Q37I is issued and the job is queued with disposition HOLD.

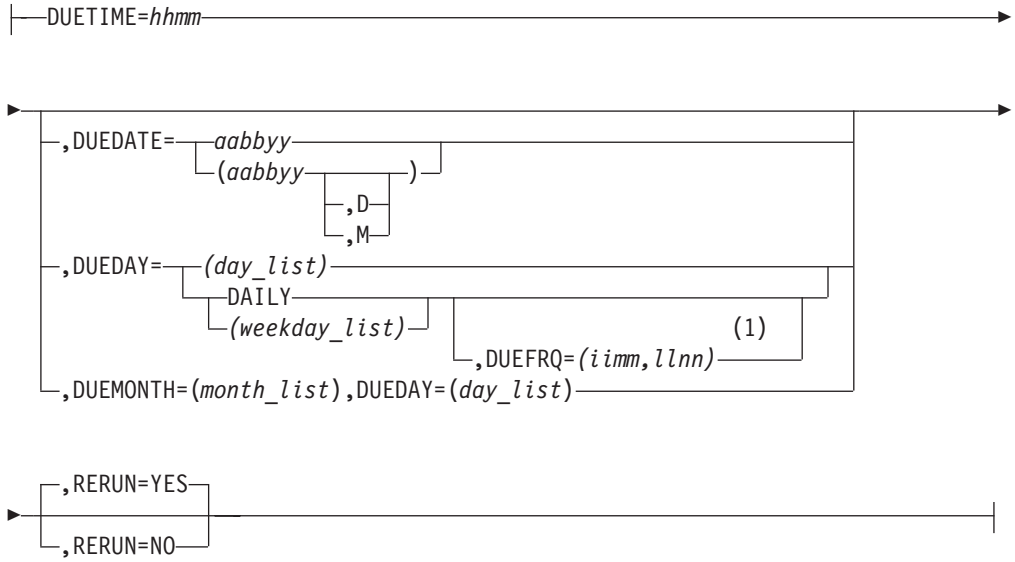
The * $$ JOB statement is required in a writer-only partition, in which case no continuation statement is allowed. It must follow immediately the * $$ EOJ statement of the preceding VSE/POWER job.

In general, the statement is interpreted at read-in spooling time. However, for writer-only partitions, or when it is received from a non-PNET node, or when it is contained in a DISP=I created job, the * $$ JOB statement is interpreted not before job execution time. If it is in error, it can be corrected then. For details, see "Coding Rules for JECL Statements Including Continuation" on page 408.

Format of the Statement



DUETIME_Options:



Notes:

1 For DUEFRQ, do not specify RERUN=YES; instead RERUN=NO becomes the default.

JNM=AUTONAME|jobname

For *jobname*, specify the name by which the job entry is to be known to VSE/POWER. If a job contains several VSE jobs, they are all logged as one VSE/POWER job using the job name which you specify. If the operand is omitted, VSE/POWER chooses the jobname=AUTONAME. (For suppression of AUTONAME jobs, refer to "SET: Setting VSE/POWER Startup Control Values" on page 473.)

You can specify a name consisting of two to eight alphameric characters. If you specify only one character for jobname, VSE/POWER issues no error message. However, manipulation of the job becomes difficult because the queue management commands interpret a one character job name as class parameter.

For a list of what is, for VSE/POWER, an alphameric character, see "Coding Rules for VSE/POWER Commands" on page 193.

Do not use any of the following as a job name:

```
ALL      LOCAL
FREE     RJE
HOLD     WRUN
```

A job name such as this may conflict with specifications in the PDISPLAY command.

Notes:

1. For rules of naming output entries of a job, refer to "Names and Numbers of Jobs and their Output" on page 404.
2. For access to jobname and jobnumber of the active VSE/POWER job, refer to *VSE/POWER Application Programming*.

BLDG=number

Specifies the programmer's building number to be placed on the job separators and recorded in the list and punch account records. You can specify a number

of up to eight alphameric characters. The specified value (default is blanks) is passed to the list or punch output queue entry produced by the current job. One can overwrite this value using the BLDG= operand of a * $$ LST or * $$ PUN statement.

BTRNC=YES | NO
The operand applies to the recording of spooled jobs on the data file.

YES
To truncate trailing blanks at the end of every record. This is the default. This function saves disk space on the data file.

NO
specifies that no blank truncation should take place. Specify NO, for example, when the reader queue entry is to be processed by the spool-access support GET service and **all** originally spooled characters should be passed to the requestor.

See also "Recording of Spooled Data on the Data File" on page 179.

CLASS=A | class
For *class*, specify the class that is to be assigned to the job in the input queue. You can specify a character from 0 to 9 and A-Z.

You can use this specification for grouping jobs that require the same I/O configuration, partition, dynamic class, 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 and recorded in the list and punch account records. You can specify a number of up to eight alphameric characters. The specified value (default is blanks) is passed to the list or punch output queue entry produced by the current job. One can overwrite this value using the DEPT= operand of a * $$ LST or * $$ PUN statement.

DISP=D | disposition
For local 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, "VSE/POWER Disposition Codes," on page 497.

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.

DUETIME=hhmm
This operand specifies the processing time using hh for the hour and mm for the minute, in 24-hour clock time (from 0000 through 2359). Leading zeroes must be specified.

For more information, see "Invocation of Time Event Scheduling Support" on page 178.

DUEDATE=aabbyy | (aabbyy,D) | (aabbyy,M)
This operand specifies the processing day using yy for year. According to the format defined for your system (that is either ddmmyy or mmddyy) aa is month and bb is day, or aa day and bb month. Leading zeroes must be specified. If you want to send the job within a network to a system whose date format differs from yours, you should use the format (aabbyy,D) | (aabbyy,M) where D means that day comes first and M means that month comes first. This operand is used when the job is to be scheduled just once. VSE/POWER checks that year is in a meaningful range. The range is:

current year - 10 <= year specified <= current year + 3.

For yy-year specified, VSE/POWER determines the century according to the 'fix-88-window rule':

0 ≤yy ≤88 is the 21st century, yielding the 4 digit year 20yy

88 < yy ≤99 is the 20th century, yielding the 4 digit year 19yy.

For more information, see "Invocation of Time Event Scheduling Support" on page 178.

DUEDAY=DAILY
This operand specifies that the job has to be scheduled every day of the year.

For more information, see "Invocation of Time Event Scheduling Support" on page 178.

DUEDAY=(weekday_list)
It specifies weekdays (MON for Monday, TUE for Tuesday, WED for Wednesday, THU for Thursday, FRI for Friday, SAT for Saturday and SUN for

Sunday) separated by commas and enclosed in parentheses. If just one abbreviation is specified, the parentheses may be omitted. It is also possible to specify a range using a hyphen between two weekdays. This specification must be enclosed in parentheses. A specification (aaa-bbb) means the job will be scheduled on each of the days between and including aaa and bbb (for example, MON-WED, FRI-MON).

For more information, see "Invocation of Time Event Scheduling Support" on page 178.

DUEDAY=(day_list)
The operand specifies a list of days within a month. The days are specified as numbers separated by commas and enclosed in parentheses. If just one number is specified, the parentheses may be omitted. A leading zero need not to be specified.

It is possible to specify a range, using a hyphen between two numbers. This specification must be enclosed in parentheses. A specification (n-m) means the job will be scheduled on each of the days between and including n and m, whereby m must be greater than n. If the DUEMONTH operand is not used, the job is scheduled on the specified days of every month.

The following are valid specifications of DUEDAY=(day_list):
- DUEDAY=15
- DUEDAY=(1,15,30)
- DUEDAY=(1-5)
- DUEDAY=(1-5,15,25-30)

For more information, see "Invocation of Time Event Scheduling Support" on page 178.

DUEFRQ=(iimm,llnn)
The Due Time Frequency operand specifies an interval using *ii* for hours and *mm* for minutes (from 0001 through 2359), after which a job must be rescheduled, and specifies a last time using *ll* for the hour and *nn* for the minute (from 0000 to 2400), after which a job is no longer to be scheduled for processing. Leading zeroes must be specified. The operand requires a DUETIME to be specified that defines the time when scheduling must occur for the first time. This operand is accepted only if DAILY or a *weekday_list* has been specified for the operand DUEDAY. DUEFRQ is rejected if DUEDATE or a *day_list* has been specified for DUEDAY. For details and usage, refer to "Scheduling a Job More Than Once Per Day" on page 171.

DUEMONTH=(month_list)
DUEMONTH specifies a list of months within a year. The months are specified as numbers separated by commas and enclosed in parentheses. If just one number is specified, the parentheses may be omitted. 1 stands for January, 2 for February, etc. A leading zero needs not to be specified.

In addition it is possible to specify a range using a hyphen between two numbers. This specification must be enclosed in parentheses. A specification (n-m) means the job will be scheduled for each of the months between and including n and m, whereby m must be greater than n.

If for both operands DUEMONTH and DUEDAY a list is specified, the job is scheduled on all specified days of all specified months.

The following are valid specifications of DUEMONTH=(month_list):
- DUEMONTH=6
- DUEMONTH=(3,6,9,12)

- DUEMONTH=(1-3)
- DUEMONTH=(1-3,5-7,12)

For more information, see "Invocation of Time Event Scheduling Support" on page 178.

ECHO=(msg_type,user_id)
This operand indicates that the job execution console messages are to be passed to the local system console and the additional indicated console name. For a detailed description of user console refer to *z/VSE Operation*.

For *msg_type* specify either:

ALL
to pass all job messages

REPLY
to pass only messages requiring an operator response

For *user_id* specify the user ID whose user console is to receive the job messages.

ECHOU=(msg_type,user_id)
This operand indicates that the job execution console messages are to be passed to the indicated local console name only. For a detailed description of user console refer to *z/VSE Operation*.

For *msg_type* specify either:

ALL
to pass all job messages

REPLY
to pass only messages requiring an operator response

For *user_id* specify the user ID whose user console is to receive the job messages.

EOJMSG=YES|NO
Specify YES if you want VSE/POWER to issue message 1Q4DI after job execution has finished. The message will be issued at the console for each job having this indication, regardless of whether the job runs in a static or in a dynamic partition, or whether JLOG has been specified in the POWER generation macro.

Specify NO or omit the operand if the message should not be issued.

FROM=user_id
'user_id' specifies the ID of the user being allowed to manipulate or retrieve the job. Up to eight alphameric characters can be specified. Note that IDs R000 to R250 take effect as RJE remote IDs. For a list of what is, for VSE/POWER, an alphameric character, see page 194.

Any local point of origin, local card reader, tape, RJE, spool access support input or PUTSPOOL is thus overwritten for the job. The output of this job reflects this specification in the FROM attribute of the queue display, while the corresponding TO-attribute shows the point of origin (unless it was overwritten by destination parameters of the * $$ JOB/LST/PUN statements).

For jobs received via the network from non-PNET nodes with an * $$ statement still contained in the input stream, the FROM= parameter is honored, too. The 'from user' specified therein is allowed to access the job at

its final destination. Refer also to "Scope of GET/CTL Access to Queue Entries" in *VSE/POWER Application Programming*.

For writer-only partitions, the FROM= specification is ignored.

Notes:

1. If a VSE/POWER job contains both the FROM operand in the * $$ JOB statement and an * $$ SLI statement for ICCF library member inclusion, then the userid assigned by the FROM operand is also used to gain access authority to ICCF library members. That means for example, the locally read in job with FROM=USERX obtains the same ICCF access rights as the job submitted by the IUI-ICCF user 'USERX' not specifying the FROM operand at all.

   If the FROM operand is not to influence the ICCF access rights, then include the SET FROM=NOSLI statement in your VSE/POWER startup sequence (see also the SET FROM statement on page 481).
2. For access to the ID of the user who submitted the active VSE/POWER job, refer to "Dynamic Access to VSE/POWER Job Attributes" in *VSE/POWER Application Programming*.

LDEST=node-id | (node-id,user_id)

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 *node-id*, specify either 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 output is to remain at the execution node; example: LDEST=(*,user_id).
- A null string (specify LDEST=(,user_id)) if the output is destined for a user at the node of origin, namely:
  - a user at the local node, provided the job has been read in a this node (and has not been received via networking) for execution, or
  - a user at the origin node, provided the job has been read in at, for example, origin Node A and has been sent for execution to another node, for example, Node B.

For *user_id*, specify the ID of the user who is to get the output. For a definition of user_id, see "Coding Rules for VSE/POWER Commands" on page 193. For details, see the DEST operand of the * $$ LST statement.

LOG=YES | NO

Specify *NO* if certain SYSLOG messages should be handled as follows:

1. z/VSE Job Control '// JOB ...' and 'EOJ ...' messages are suppressed on the console (but appear in the hardcopy file)
2. VSE/POWER waiting-for-work messages 1Q34I or 1Q3EI are not issued at all.

Other VSE/POWER, Job Control, or application program messages are not affected. This option will be shown in the VSE/POWER 1Q47I start-job logging message for checking purposes. End-of-job processing is then reported by message 1QC7I.

Specify YES or omit the operand if no message should be suppressed on SYSLOG.

Note: The LOG operand

1. Has no effect in writer-only partitions.
2. Is not passed on to DISP=I created jobs.
3. Has no effect if found in an * $$ JOB statement interpreted at job processing time.
4. Is not related to the Job Control or Attention Routine LOG or NOLOG commands.

For further details, see "Unrecorded Processing" on page 120.

NETACCT=account-number
This operand specifies the networking account number for NJE jobs. Up to 8 alphameric characters can be specified. For details of network accounting see *VSE/POWER Networking*.

NORUN=IGN
Specify IGN to ignore the SET NORUN=YES specification for this job during VSE/POWER restart after abnormal termination.

NTFY=YES | (node-id,user_id)
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 (1Q5DI).
- 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.
- Job output is created and default attributes have to be taken for the list or punch output entry (1Q8CI).

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.

(node-id,user_id)
If notify messages for the job are to be sent to a specific user.

For *node-id*, specify the name of the node to which the user specified for 'user_id' 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: (,user_id). If a user at the executing node is to receive notify messages, you may replace the executing node name by an asterisk (*); for example: (*,user_id).

For *user_id*, specify the ID of the user who is to receive the messages. For a definition of user_id, see "Coding Rules for VSE/POWER Commands" on page 193. For details, see the DEST operand of the * $$ PUN statement.

Specify 0 for user_id if the messages are to be routed to the central operator.

If your VSE system operates under a VM system, 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 a VM system.

PDEST=node-id | (node-id,user_id)
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 *node-id*, specify either 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 output is to remain at the execution node; example: PDEST=(*,user_id).
- A null string (specify PDEST=(,user_id)) if the output is destined for a user at the node of origin, namely:
  - a user at the local node, provided the job has been read in a this node (and has not been received via networking) for execution, or
  - a user at the origin node, provided the job has been read in at, for example, origin Node A and has been sent for execution to another node, for example, Node B.

For *user_id*, specify the ID of the user who is to get the output. A definition of user_id, is given under "Coding Rules for VSE/POWER Commands" on page 193. For details, see the DEST operand of the * $$ PUN statement.

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 VSE/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 and recorded in the list and punch account records. You can specify a name of up to 20 alphameric characters. If the name contains special characters, such as a comma or a blank, you must enclose the name within a pair of apostrophes. An apostrophe within the name must be entered as two apostrophes. The specified value (default is blanks) is passed to the list or punch output queue entry produced by the current job. One can overwrite this value using the PROGR= operand of a * $$ LST or * $$ PUN statement.

PWD=password
Specify this operand if you wish to have the job protected by a VSE/POWER password. The password you define must be specified if anyone other than the central operator or the system administrator wants to manipulate the job while it is in the local queue (for example, have the job processed in another partition).

If a program needs to access a password protected job, make sure it presents this password in its service request. If spool-access support is used, it is a GET or CTL service request. If the spool macro support is used, it is a GETSPOOL or CTLSPOOL macro request.

A password must be defined as an alphameric string of up to eight characters. It can also be specified in an * $$ LST or * $$ PUN statement.

Note: The former mismatch of the default hex zero password (for locally submitted jobs) with the default blank password (for access from the programmer interface) has been removed in favor of the new spool-access support.

It is recommended to specify this operand last. See "Correction of JECL Errors" on page 405.

RERUN=YES | NO

RERUN specifies whether a job is or is not to be rescheduled if the due date has expired during the time VSE/POWER was down.

With the default specification of RERUN=YES, the job will be processed only once even when its due date has expired more than once during the system down time.

With the specification of RERUN=NO, the job will not be rescheduled, irrespective of how often its due date has expired during the down time. However, if the due date has expired the same day as VSE/POWER is restarted, the RERUN=NO specification will not take effect. Thus it is avoided that a scheduling event has been missed because the system was brought down for a few minutes.

If the operand RERUN=NO has been specified and (a) the due date has expired earlier than the current date and (b) a job has to be scheduled just once, this job is queued into the non-dispatchable queue. Its disposition is changed from D to H, or from K to L.

Note: For further background information refer to "Scheduling of Jobs at VSE/POWER Startup Time" on page 173.

ROOM=number

Specifies the programmer's room number to be placed on the job separators and recorded in the list and punch account records. You can specify a number of up to eight alphameric characters. The specified value (default is blanks) is passed to the list or punch output queue entry produced by the current job. One can overwrite this value using the ROOM= operand of a * $$ LST or * $$ PUN statement.

SEC=(user_id,password) | propagated security values

Use this operand to access VSE protected resources. Specify the VSE security userid and password values for the VSE job(s) contained within the VSE/POWER job. Both the user_id and the password must be specified. The VSE/POWER job can inherit propagated VSE security values from a parent job. For details, see "Job Execution with Propagated VSE Security Values" on page 12. For detailed information on the security password and on the syntax of the user_id, see *z/VSE Administration*.

The operand may also be specified for a job at read-in time on an unsecured system. The operand does, however, not take effect until the job is executed on a system on which the VSE access control feature is activated.

Specify:

user_id
The VSE security user_id for this job.

password
The VSE security password for this job.

It is recommended to specify the SEC operand last. For details see "Correction of JECL Errors" on page 405.

SECAC=
This operand allows you to control Spool Access Protection for an individual job and its output by inheritance.

If this parameter is not specified, and VSE/POWER Spool Access Protection is active, the job will be Spool Access Protected *if possible*. For the definition of "spool access protected entry", see "VSE/POWER Spool Access Protection" on page 14.

SECAC=NO indicates that the job and its output are *required not* to be Spool Access Protected. If active, VSE/POWER Spool Access Protection is ignored.

SYSID=N | n
This operand applies to shared-spooling. 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 VSE/POWER generation macro).

SYSID=N
If the job can be processed on any of the sharing systems, meaning SYSID=Null.

Omit the operand if your job is to be processed on any of the sharing systems.

The specified (or default) SYSID is also passed to any list or punch output queue entry produced by the current job. You can overwrite this default rule using the SYSID operand of the * $$ LST or * $$ PUN statement.

TDISP=D | disposition
For transmission disposition, specify how the VSE/POWER job is to be handled when placed into the transmission queue on your node. Your specification may be one of the following:

D Delete after transmission. The job is automatically scheduled for transmission according to its priority. When transmission is complete, VSE/POWER deletes the job from the transmission queue.

H Hold the job in the transmission queue. It is not dispatched for transmission 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 transmission. The job is automatically scheduled for transmission according to its priority. When transmission is complete, VSE/POWER retains the job in the transmission queue with a disposition of L.

L Leave the job in the transmission queue. It is not dispatched for transmission 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.

For further information on dispositions, see also Appendix A, "VSE/POWER Disposition Codes," on page 497.

UINF=user_info (or USER=user_info)

The keywords UINF= and USER= are synonymous. Specify up to 16 bytes to be used as user information. The user information is retained in a 16-byte internal field, left-justified with trailing blanks, and it is:

- displayed when the operand FULL=YES is specified in the PDISPLAY command via the U='...' display field
- included in Execution account records
- printed on separator pages
- printed on SYSLOG with message 1Q47I.
- passed on to list and punch output entries created by the subject job

It can be altered using the UINF= operand of PALTER and searched for in queue manipulation commands by the CUINF= operand.

Syntax conventions:

- For *user_info*, specify up to 16 characters or blanks. If the character string includes a blank or comma, the entire string must be enclosed in a pair of single quotes; otherwise, the first blank or comma is interpreted as a delimiter by VSE/POWER. If you are nesting single quotes inside single quotes, you must specify them as two adjacent single quotes, as shown under "Examples of the PALTER Command" on page 211.
- It is recommended to specify characters whose hexadecimal representation is not affected by uppercase conversion. All values are converted to uppercase, as described under "Uppercase Conversion of Characters in JECL/JCL Statements" on page 405.
- For a display of the recorded user information, refer to "Format 1-4: PDISPLAY LST,FULL=YES" on page 257.

Note: For access to the UINF='...' information of the active VSE/POWER job, refer to "Dynamic Access to VSE/POWER Job Attributes" in *VSE/POWER Application Programming*.

XDEST=node_id|(node_id,user_id)

The operand specifies where VSE/POWER is to route the job for processing.

For *node_id*, specify the name of the applicable node as defined to VSE/POWER by way of the PNODE macro.

For *user_id*, specify the ID of the destination user, a VSE system with VSE/POWER and operating under a VM system, 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 user_id specification.

The operand is ignored for a writer-only partition.

* $$ LST: Defining 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 * $$ LST statement for the same spooled printer
An IPWSEGM macro (with the KEEP=NO operand) request for the same spooled printer
SEGMENT macro for the same spooled printer
A VSE/POWER end-of-job (* $$ 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 every * $$ LST statement within a VSE/POWER job, VSE/POWER creates a separate list-queue entry and assigns a job number to this entry provided that data was spooled for that queue 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 spooled (at job execution time) to a non-3800 printer, VSE/POWER ignores these operands except for FLASH=.

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*).

The statement is interpreted at job execution time. If it is in error, it can be corrected then. For more information, see "Coding Rules for JECL Statements Including Continuation" on page 408. If it is ignored, conditional job control may be searching for a GOTO label as described under "Interaction with z/VSE Conditional Job Control Language" on page 108.

If your * $$ LST statement does **not** contain the LST=cuu|SYSxxx operand (see page 437), it takes effect for the first printer in the PRINTERS=cuu,cuu,... chain for the related partition. Thus, if you want to ensure that the * $$ LST statement always takes effect for your output entry, it is recommended to specify the LST operand for the assigned printer. For more details, refer to "Important Specifications for Output Spooling" on page 147.

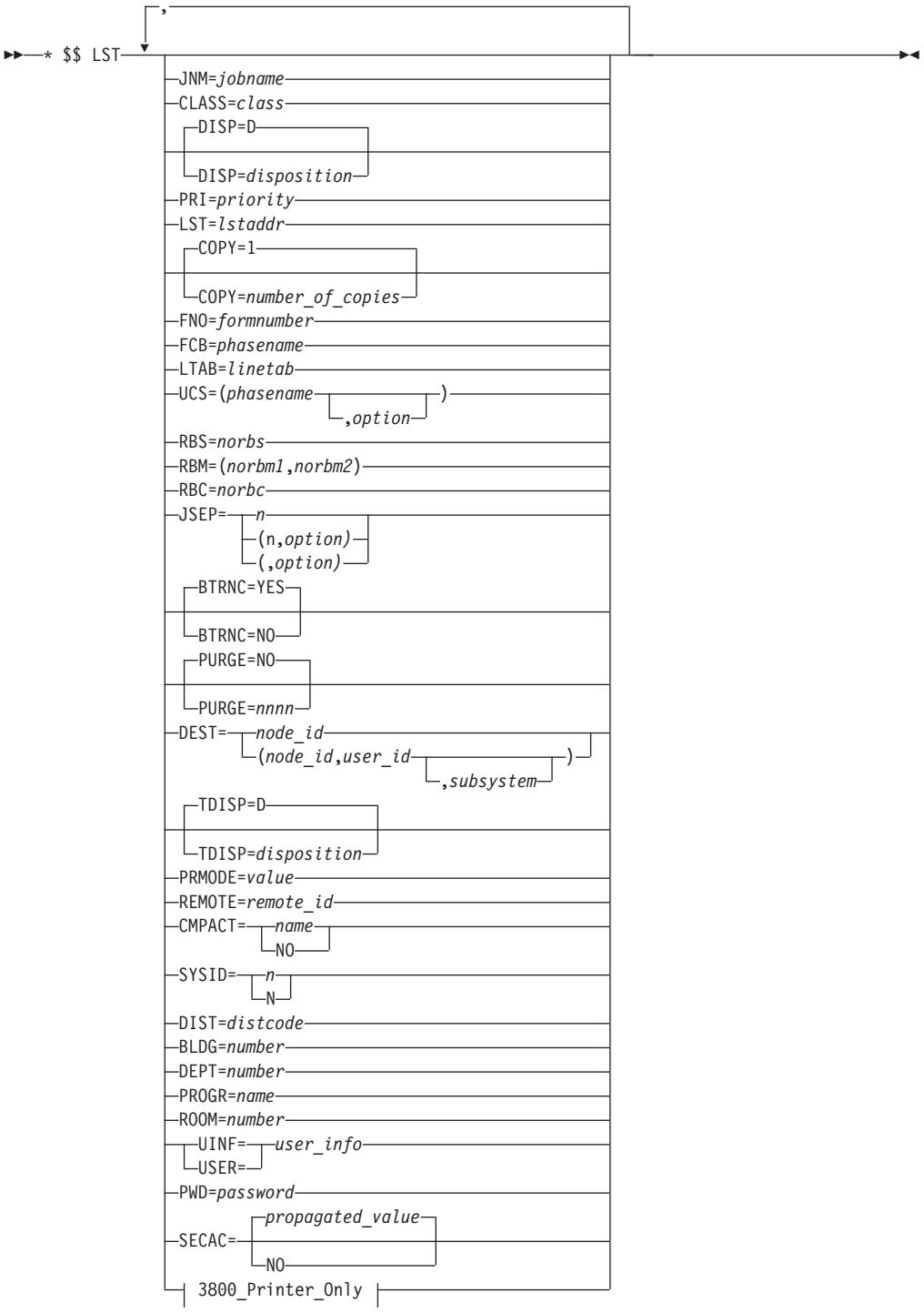
For tape spooling (DISP=T), please refer to "Tape Processing with VSE/POWER" on page 151.

Note: The listed operands of the * $$ LST statement are interpreted and recorded by VSE/POWER for its "own" later processing of the output entry to be created. Therefore, all operands are called "native" VSE/POWER operands. For specifying your private, so-called "user-defined" operands in the LST

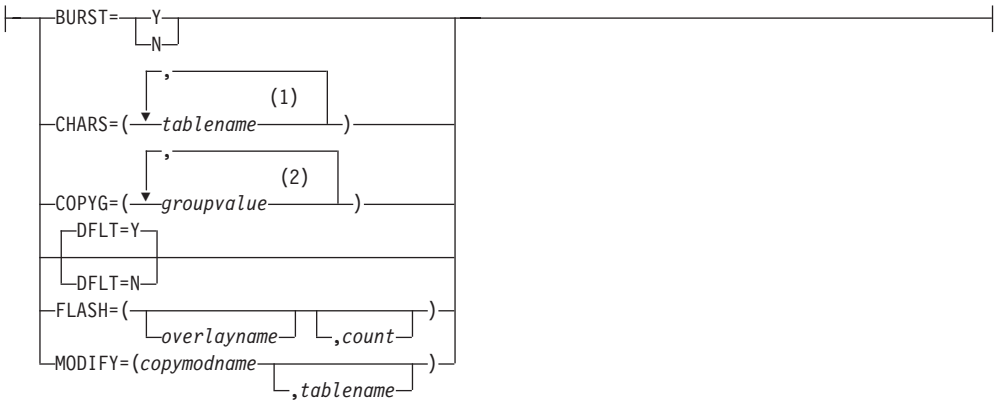
JECL statement (as required by PSF or CICS, for example), refer to "DEFINE: Specifying User-Defined Output Operands" on page 488.

Note: Refer to "Important Specifications for Output Spooling" on page 147 for hints in solving problems with spooling output.

Format 1: Spooling Print Output to Disk



3800_Printer_Only:



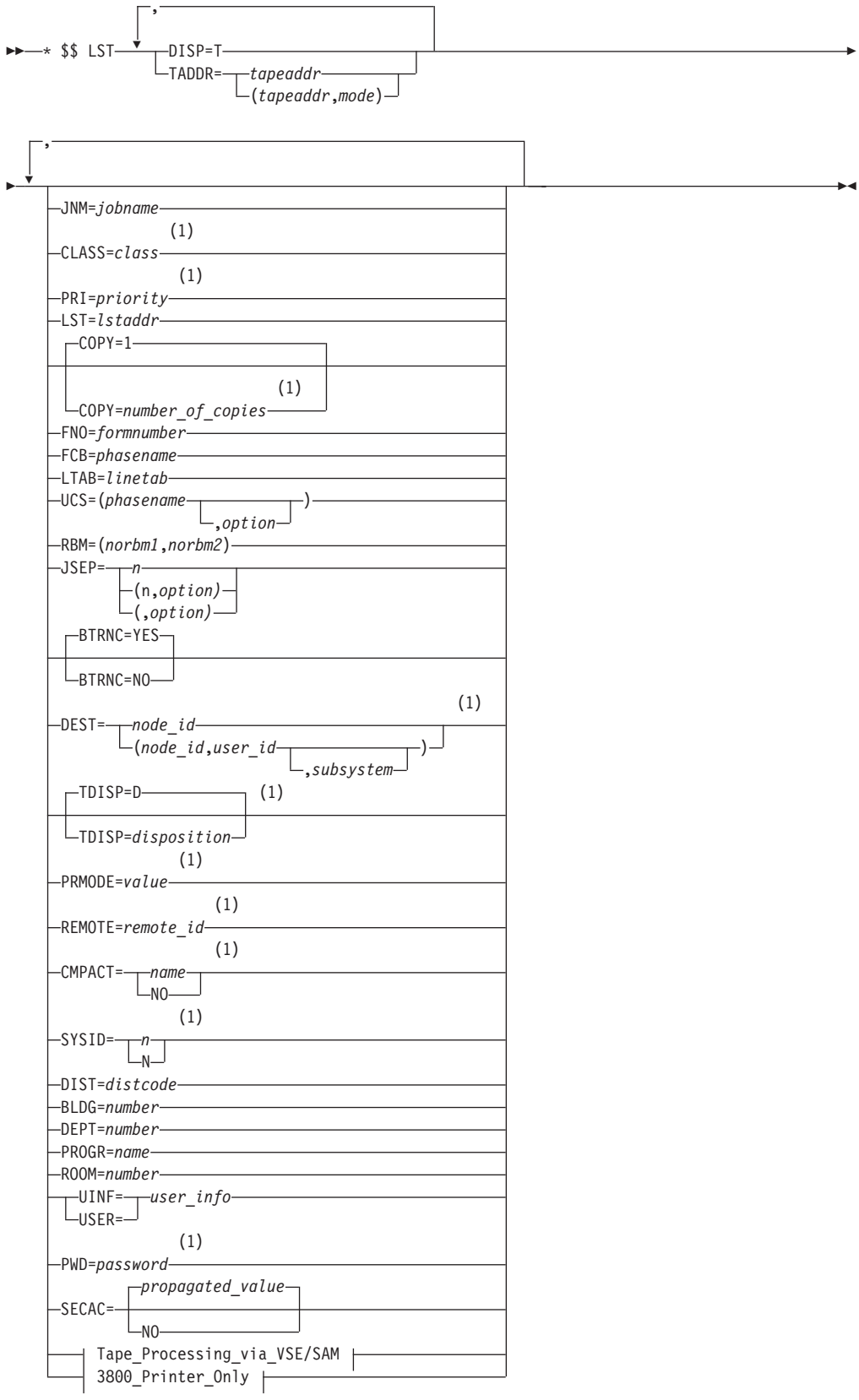
Notes:

1 You can specify up to four table names.

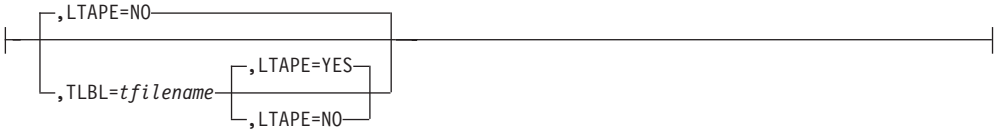
2 You can specify up to eight group values.

The ’3800_Printer_Only’ operands (except for FLASH=) are effective only if output is spooled to a 3800 type printer.

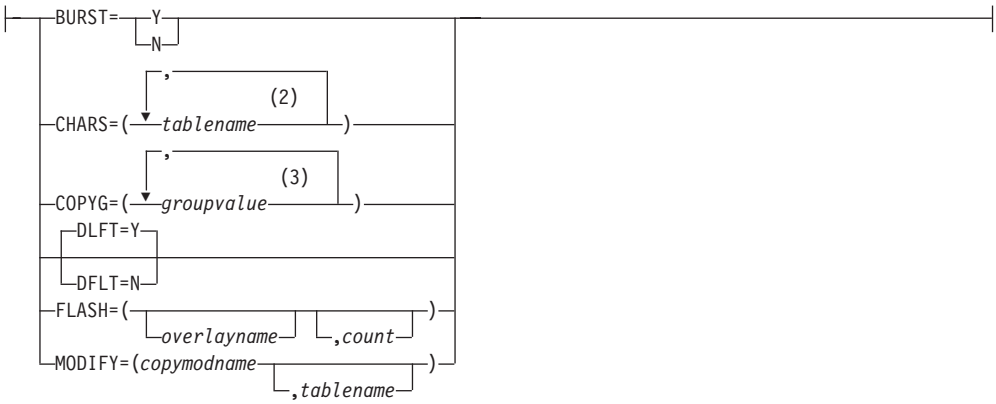
Format 2: Spooling Print Output to Tape



Tape_Processing_via_VSE/SAM:



3800_Printer_Only:



Notes:

1 The operands
- Have no effect when output spooled to tape is printed later
- Take effect again when the tape-spooled output is loaded to the list queue by POFFLOAD LOAD|SELECT.

2 You can specify up to four table names

3 You can specify up to eight group values.

The '3800_Printer_Only' operands (except for FLASH=) are effective only if output is spooled to a 3800 type printer.

In addition to the above listed 'native' VSE/POWER operands, you can define your own JECL output operands in autostart DEFINE statements and specify them in the * $$ LST or * $$ PUN statement. For details on the sequence of interpretation and other information, see "DEFINE: Specifying User-Defined Output Operands" on page 488.

BLDG=number
Specifies the programmer's building number to be placed on the job separators and recorded in the list account record. You can specify a number of up to eight alphameric characters. If omitted, VSE/POWER uses the BLDG= specification, if any, in the * $$ JOB statement.

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 every output are automatically separated from those of the preceding output. Copies and copy groups are also separated.

BTRNC=YES|NO
The operand applies to the recording of spooled output on the data file.

YES
To truncate trailing blanks at the end of every record. This is the default. This function saves disk space on the data file.

NO
specifies that no blank truncation should take place. Specify NO, for example, when the list queue entry is to be processed by the Spool-Access Support GET service and **all** originally spooled characters should be passed to the requestor.

See also "Recording of Spooled Data on the Data File" on page 179.

CHARS=(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. If neither CHARS nor UCS is coded and at real printing time output goes to an IBM 3800 printer, VSE/POWER uses the CHARS definition specified in the VSE attention command SETDF

- Given at spooling time when output is produced for an 3800 type printer
- Given at real printing time when output is produced for a non 3800 printer.

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.

For *class*, you can specify any letter of the alphabet, or any numeral 0 through 9.

If you omit the operand, VSE/POWER assigns the class that was specified as default output class in the PSTART command for the static partition used to execute your program. For a dynamic partition, the default output class is 'A', or it is the execution class itself, if the SET DYNOUTCL=DYNCL autostart statement has been activated.

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, see Appendix B.

COPY=1 | number_of_copies
The operand specifies the number of copies to be printed from the list queue entry. 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)
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.

For groupvalue, specify the number of copies of every 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 every 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.

DEPT=number
Specifies the programmer's department number to be placed on the job separators and recorded in the list account record. You can specify a number of up to eight alphameric characters. If omitted, VSE/POWER uses the DEPT= specification, if any, in the * $$ JOB statement.

DEST=node_id | (node_id,user_id) | (node_id,user_id,subsystem)
Any specification in this operand overrides the specification in the LDEST operand of the * $$ JOB statement.

If you specify REMOTE=remote_id 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 ID to the ID specified for remote_id in the REMOTE operand.

Specify for:

node_id
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 (specify DEST=(,user_id)) if the output is destined for a user at the node of origin, namely:
  - a user at the local node, provided the job has been read in a this node (and has not been received via networking) for execution, or

- a user at the origin node, provided the job has been read in at, for example, origin Node A and has been sent for execution to another node, for example, Node B.

user_id
One of the following:

- The applicable user ID. For a definition of user_id, see "Coding Rules for VSE/POWER Commands" on page 193. If you specify a user ID (other than R000 or 000), 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. Such a spool entry can be manipulated only by the origin user ID. If Spool Access Protection is active, the user ID accessing or manipulating the entry must be authenticated (see "VSE/POWER Spool Access Protection" on page 14). In this case, if the entry is to be accessible by non-authenticated users, SECAC=NO should also be specified.

Note: Specifying *only* a node_id for the DEST operand will nullify any default user ID and make the output processable on a local printer.

subsystem
Applies if, for node_id, you specified the name of an OS/390 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 (local) *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 automatically is 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.

The user program owns the DISP=N device until another * $$ LST statement for the same device or the end of the VSE/POWER job comes.

T Spool the output to tape. For dependencies, see also the TADDR= operand on page 443.

This operand is ignored if specified in a * $$ LST statement used to segment output via the IPWSEGM KEEP=YES macro.

If you specify DISP=T, the following operands are ignored:

```
PURGE=nnn       RBC=       RBS=
```

To start printing the output spooled on tape, use the PSTART LST,uraddr,X'tapeaddr' command. For * $$ LST operands that do not take effect at print time, see "Format 2: Spooling Print Output to Tape" on page 429.

To load the output spooled on tape to the list queue, use the POFFLOAD LOAD|SELECT command. Reloaded entries will get disposition D.

For further information on dispositions, see also Appendix A, "VSE/POWER Disposition Codes," on page 497.

DIST=distcode

For *distcode* specify the distribution code (one up to eight alphameric characters) for output created under VSE and printed by a VM writer task. The distribution code is passed with the CP CLOSE command to VM, but it does not change the distribution code in the VM-CP directory. The distribution code is printed on VM separator pages.

VSE/POWER prints the distribution code on its own separator pages, even for non-VM writer tasks.

The distribution code is also passed to other nodes. The code is interpreted, however, only by PNET nodes.

The distribution code will be shown in the PDISPLAY command if for output queue entries the option FULL=YES is set.

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 23. 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) and for determination of the separator page size. 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 startup 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 VSE/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 VSE/POWER generation macro matches the default FCB image with regard to channel positions and page size. If you already want to use the printer-specific (non-3800) default FCB at spool time, refer to the SET FCB=DEFAULT start-up statement on page 476. For 3800 printers, the default FCB is extracted from the FCB= specification of the SETPRT or SETDF statement/command.

Notes:

1. VSE/POWER does not support Select Vertical Format (SVF). When list output is to be produced at an SNA terminal, the operator at this station has to set up the vertical tab tables. However, the FCB name is sent to the terminal 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 an IPWSEGM or 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.
4. If an already used FCB is recataloged, use the PDELETE FCB command; for details see "Format 3: Clearing the VSE/POWER FCB Table" on page 222.
5. The first and last half-inch of a 3800 page cannot be used for printing, although this area is contained in the FCB (or even LTAB) page size.

FLASH=(overlayname,count)| (,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 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.

JNM=jobname
For *jobname* specify the name VSE/POWER should assign to the list queue entry. You can specify a name consisting of two to eight alphameric characters. For a list of what VSE/POWER considers an alphameric character, see "Coding Rules for VSE/POWER Commands" on page 193.

Do **not** use any of the following as a job name:

```
ALL      LOCAL
FREE     RJE
HOLD     STATUS
```

If output segmentation is specified by the RBS parameter, each segment gets the name specified as *jobname*.

If you omit the operand, the output gets the name specified in the * $$ JOB statement.

JSEP=n | (n,option) | (,option)
The operand specifies the number of desired job separator pages preceding and trailing a printed queue entry and, optionally, whether separator pages are to be inserted also between every copy of an output file. If you omit this operand, VSE/POWER uses your definition in the JSEP and COPYSEP operands of the VSE/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, see the JSEP operand under "POWER Generation Macro" on page 57, and "Separator Pages - Layout and Control" on page 181.

Specify for:

n The number (any from 0 to 9) of job separator pages to be printed.

If you specify JSEP=0, no separator pages are printed, even if a non zero value is specified in the JSEP parameter of VSE/POWER generation macro. 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 VSE/POWER generation macro.

LST=listaddr
Use this operand to specify the printer device for which the * $$ LST statement should take effect. For example, if a program produces two or more different reports which, in addition, may require different forms. You should specify this operand in the * $$ LST statement for every 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.

Note: If you *omit* this operand, the default is the first printer address specified at partition startup for which your * $$ LST statement applies. If, for example

```
PRINTERS=00E,01E,02E
```

was specified during startup, 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.

For more details, refer to "Important Specifications for Output Spooling" on page 147.

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 VSE/POWER generation macro if you omit the operand.

In the operand, use number pairs to 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 (STDOPT LINES value) for d12.

For VSE/POWER, the sum of d0 (and d1) 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 startup) 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.

See also "When Print Output is Misaligned" on page 116.

LTAPE=YES|NO
If specified, VSE/SAM is used during OPEN, CLOSE and end-of-volume (EOV) tape processing.

This operand is ignored if specified in a * $$ LST statement used to segment output via the IPWSEGM KEEP=YES macro.

This operand must follow the DISP=T operand.

YES
Indicates VSE/SAM labeled tape processing.

NO
Indicates VSE/SAM unlabeled tape processing.

If TLBL= is specified in addition to LTAPE=NO, an unlabeled tape is to be processed using VSE/SAM. However, in addition to LTAPE=NO, VSE/POWER will pass the tfilename value to VSE/SAM as for labeled tape processing. This operand value is normally needed only by Tape Management Systems which might use the // TLBL statement to indicate tape information, such as tape density. The TLBL= value then applies to all unlabeled tapes produced.

Note: Unlabeled tape processing may result in more than one output tape being produced; however, VSE/POWER processes only a single unlabeled input tape.

MODIFY=(copymodname)| (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.

PRMODE=value
This operand applies to the printing of output and specifies the printer processing mode. Any alphameric value with the length of 1-8 characters is accepted. VSE/POWER does not interpret this operand for its own printing. Instead, the specified value is handed over to spool-access support users and to other networking nodes. Because this operand affects the components which process the output entry, specify values which are meaningful to these components. The value will be saved in the general section of the data set header record (DSHR).

For the handling of the 'native' operand 'PRMODE' at the same time as the 'user-defined' operand, see "DEFINE: Specifying User-Defined Output Operands" on page 488.

PROGR=name
Specifies the name of the person (programmer) responsible for the job. This name is printed on the separator pages and recorded in the list account record. You can specify a name of up to 20 alphameric characters. If the name contains special characters, such as a comma or a blank, you must enclose the name within a pair of apostrophes. An apostrophe within the name must be entered as two apostrophes. If omitted, VSE/POWER uses the PROGR= specification, if any, in the * $$ JOB statement.

PURGE=NO|*nnnn*
This operand applies to the conditional suppression of list output for the spooled device. For *nnnn* you may specify a return code value of 0-4095 (greater values are reduced to 4095).

The output spooled for this device is purged if a job has terminated without cancel condition and the maximum program return code (collected over several job steps) is
not provided at all, or
does not exceed *nnnn* (or 4095 whichever is smaller)

If output is segmented by multiple * $$ LST statements or the IPWSEGM or SEGMENT macro, cancel and maximum return code conditions effective at the point of segmentation are used. Note, however, that return codes are not made available before Job Control end-of-job time.

If used with other * $$ LST operands, the PURGE=*nnnn* specification

- Is ignored with DISP=T/N or RBC=m(>0).
- Is ignored for writer-only partitions.
- Resets count-driven segmentation to RBS=0.
- Resets the 'limit exceeded' message to RBM=(0,0).

If output is segmented implicitly by a SETPRT or LFCB request, the PURGE=nnnn specification is not valid.

Specify `NO` or omit the operand if the spooled output should be made available in the LST or XMT queue.

See also "Unrecorded Processing" on page 120.

PWD=password
Specify this operand if you wish to have your output protected by a VSE/POWER 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: Marking the Start of a VSE/POWER Job" on page 413. It is recommended to specify this operand last. See "Correction of JECL Errors" on page 405.

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 999,999. If you specify 0, no checkpoint is taken. The output (incomplete queue entry) then is discarded when VSE/POWER starts again.

RBM=(norbm1,norbm2)
Specify for:

norbm1
The number of lines (list records) to be intercepted at output spooling time before the operator gets the "limit exceeded" message 1Q52I. Specify a number from 0 to 999,999. A specification of 0 for *norbm1* and 0≤*norbm2*≤999999 means that no warning message is given at all.

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 VSE/POWER generation macro.

norbm2
The additional number of list records to be intercepted every time before VSE/POWER reissues the above mentioned message. Again, you can specify a number from 0 to 999,999. A specification of 0<*norbm1*≤999999 but 0 for *norbm2* means that message 1Q52I is issued once when *norbm1* is reached but no further warning message is created.

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 VSE/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.)

This operand is ignored if specified in a * $$ LST statement used to segment output via the IPWSEGM KEEP=YES macro.

For norbs specify a number from 1 to 999,999. 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 VSE/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, for example, you request four copies of a 1000 page output and a segment count of 500, then your output is arranged as follows:
1. Four copies of pages 1 through 500, followed by

2. Four copies of pages 501 through 1000.

If your job produces more than 127 segments, VSE/POWER assigns a new job number to each following group of 127 segments and starts with suffix 1 for each new job number.

For special considerations related to RBS segmentation of CPDS queue entries, refer to "Page Counting for CPDS Queue Entries" on page 146.

REMOTE=remote_id
For remote_id, specify the remote ID, a three-digit number, to which the output is to be routed.

Specify 0 for remote_id 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.

ROOM=number
Specifies the programmer's room number to be placed on the job separators and recorded in the list account record. You can specify a number of up to eight alphameric characters. If omitted, VSE/POWER uses the ROOM= specification, if any, in the * $$ JOB statement.

SECAC=
Overrides the SECAC mode inherited from the job. SECAC=NO indicates that the output is *required not* to be spool access protected. If active, VSE/POWER Spool Access Protection is ignored.

If this parameter is not specified, and VSE/POWER Spool Access Protection is active, and the origin job did not specify `* $$ JOB SECAC=NO`, then by default the output will be Spool Access Protected *if possible*. To obtain Spool Access Protection eligibility, the output spool entry will be tagged with the user ID(s) obtained from:

1. The origin userid of the job if available
2. The target userid obtained from the DEST=(...,userid) operand (or the LDEST/PDEST operand of the `* $$ JOB` statement), if available

Otherwise, the entry is not eligible for Spool Access Protection.

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 VSE/POWER generation macro).

SYSID=N
If the output is to be available on any of the sharing systems, meaning SYSID=Null.

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 *z/VSE System Control Statements*.

At completion of the tape spooling function with user specified density, VSE/POWER resets the density specification to the standard one defined at IPL or by a later permanent ASSGN statement.

Specifying this operand forces a disposition of T.

This operand is ignored if specified in a * $$ LST statement used to segment output via the IPWSEGM KEEP=YES macro.

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.

For labelled tape processing only one output file can be written to a single tape. For unlabelled tape processing, two or more output files can be written to a single tape.

Multivolume segmentation is supported. The output to a multivolume segmentation tape is split at logical boundaries, which is either a skip to channel 1 or a full page.

TDISP=D | disposition
For (transmission) *disposition*, specify how the list output is to be handled when placed into the transmission queue on your node. Your specification may be one of the following:

D Delete after transmission. The output entry is automatically scheduled for transmission according to its priority. When the transmission is complete, VSE/POWER deletes the output entry from the transmission queue.

H Hold the output entry in the transmission queue. It is not dispatched for transmission until the operator either
- Changes the entry's disposition to K or D by way of a PALTER command, or
- Issues a PRELEASE command for the output.

K Keep after transmission. The output entry is automatically scheduled for transmission according to its priority. When transmission is complete, VSE/POWER retains the output entry in the transmission queue with a disposition of L.

L Leave the output entry in the transmission queue. It is not dispatched for transmission until the operator either:
- Changes the entry's disposition to K or D by way of a PALTER command, or
- Issues a PRELEASE command for the output.

For further information on dispositions, see also Appendix A, "VSE/POWER Disposition Codes," on page 497.

TLBL=tfilename
For tfilename specify one to seven characters of a tape label `// TLBL filename...`. It is assumed that the `// TLBL` statement is available to the VSE/POWER partition. For further details see "Labeled Tape Support" on page 152. For the syntax of the TLBL statement, see *z/VSE System Control Statements*.

If specified and LTAPE= is not specified, LTAPE=YES is assumed.

This operand must follow the DISP=T operand.

For more information about using the TLBL statement see the description of LTAPE.

UCS=(phasename)|(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 lowercase letters to be printed in uppercase.

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 ID and makes a print band verification by comparing the specified print band ID with that of the mounted band.

UINF=user_info (or USER=user_info)
The keywords UINF= and USER= are synonymous. Specify up to 16 bytes to be used as user information. The specified character string replaces whatever string (if any) was specified in the `* $$ JOB` statement for the job. The user information is retained in an internal field, left-justified with trailing blanks, and it is:

- displayed when the operand FULL=YES is specified in the PDISPLAY command via the U='...' display field
- included in List account records
- printed on separator pages

It can be altered using the UINF= operand of PALTER and searched for in queue manipulation commands by the CUINF= operand.

Syntax conventions:

- For *user_info*, specify up to 16 characters or blanks. If the character string includes a blank or comma, the entire string must be enclosed in a pair of single quotes; otherwise, the first blank or comma is interpreted as a delimiter by VSE/POWER. If you are nesting single quotes inside single quotes, you must specify them as two adjacent single quotes, as shown under "Examples of the PALTER Command" on page 211.
- It is recommended to specify characters whose hexadecimal representation is not affected by uppercase conversion. All values are converted to uppercase, as described under "Uppercase Conversion of Characters in JECL/JCL Statements" on page 405.
- For a display of the recorded user information, refer to "Format 1-4: PDISPLAY LST,FULL=YES" on page 257.

* $$ PUN: Defining 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 * $$ PUN statement for the same spooled punch device
- An IPWSEGM macro (with the KEEP=NO operand) request for the same spooled punch device
- A SEGMENT macro for the same spooled punch device
- An end-of-job (* $$ 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 every * $$ PUN statement within a VSE/POWER job, VSE/POWER creates a separate punch-queue entry and assigns a job number, provided that data was spooled for that queue entry. 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.

The statement is interpreted at job execution time. If it is in error, it can be corrected then. For more information, see "Coding Rules for JECL Statements Including Continuation" on page 408. If it is ignored, conditional job control may be searching for a GOTO label as described under "Interaction with z/VSE Conditional Job Control Language" on page 108.

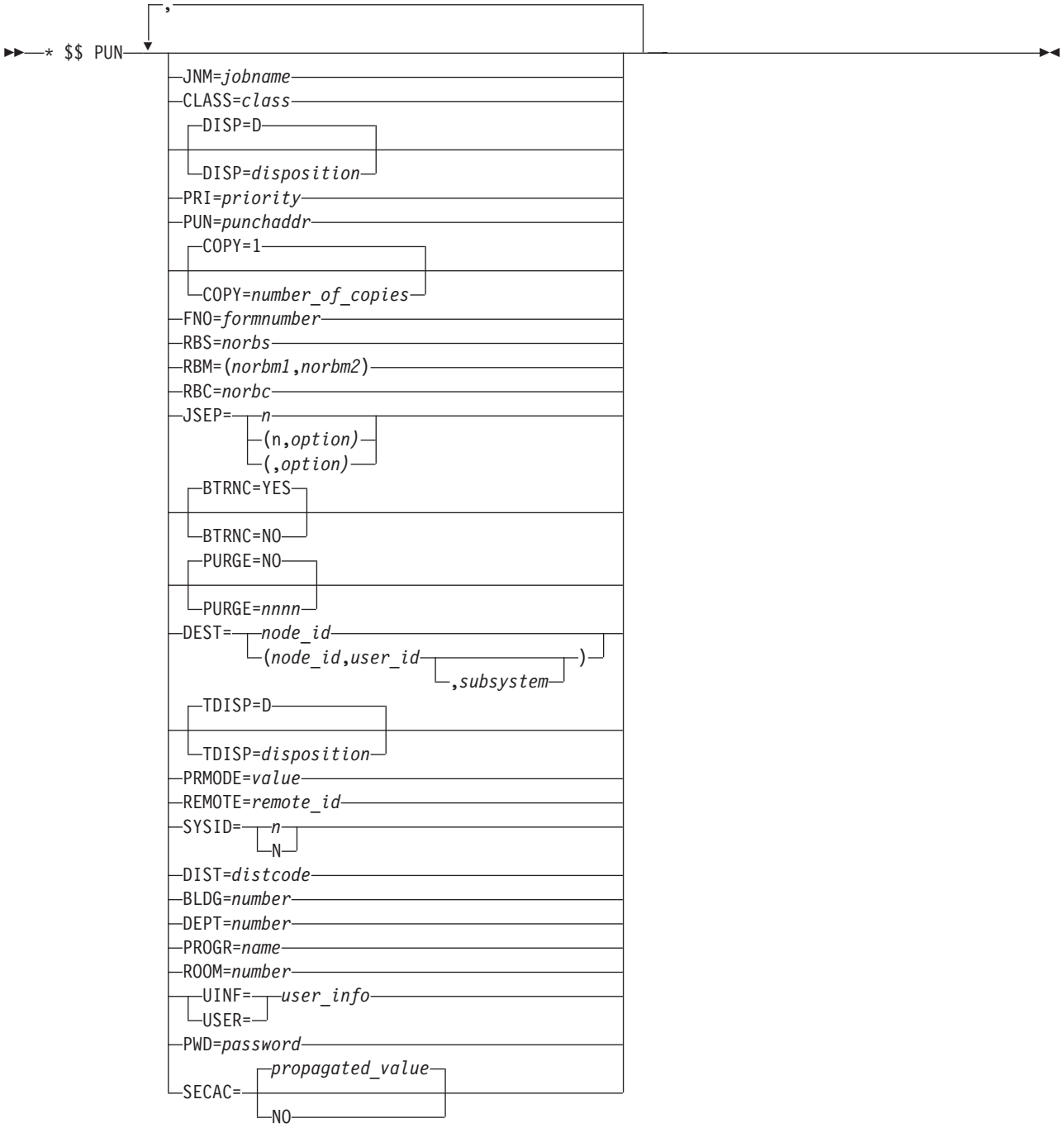
If your * $$ PUN statement does **not** contain the PUN=cuu/SYSxxx operand (see also page 455) it takes effect for the first punch device in the PUNCHES=cuu,cuu chain for the related partition. Thus, if you want to make sure that the * $$ PUN statement and its attributes always take effect for your output entry, it is recommended to specify the PUN operand for the assigned punch device. For more details, refer to "Important Specifications for Output Spooling" on page 147.

For tape spooling (DISP=T), please refer to "Tape Processing with VSE/POWER" on page 151.

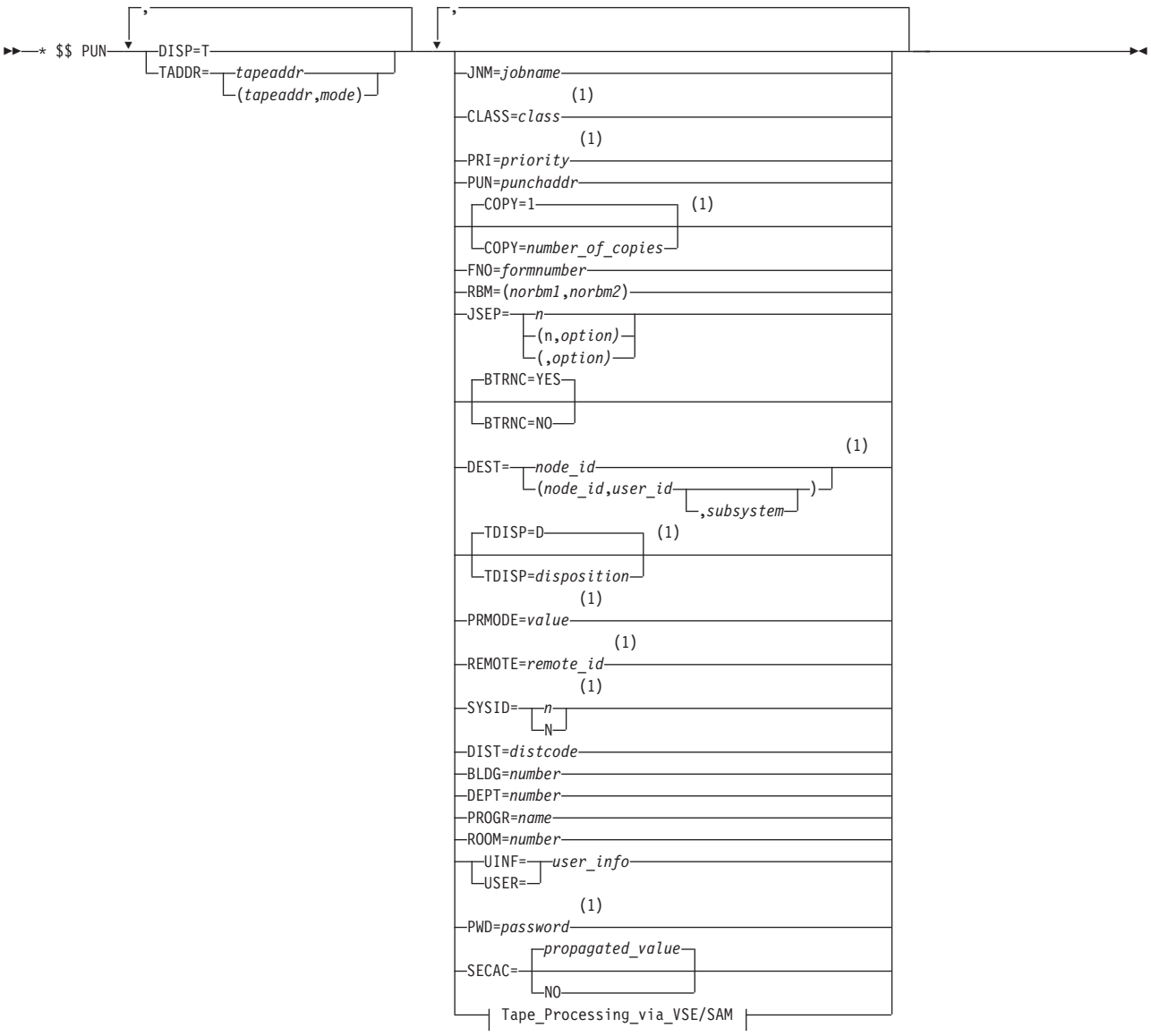
Note: The listed operands of the * $$ PUN statement are interpreted and recorded by VSE/POWER for its ″own″ later processing of the output entry to be created. Therefore, all operands are called ″native″ VSE/POWER operands. For specifying your private, so-called ″user-defined″ operands in the PUN JECL statement (as required by PSF or CICS, for example), refer to "DEFINE: Specifying User-Defined Output Operands" on page 488.

Note: Refer to "Important Specifications for Output Spooling" on page 147 for hints in solving problems with spooling output.

Format 1: Spooling Punch Output to Disk



Format 2: Spooling Punch Output to Tape



Tape_Processing_via_VSE/SAM:

Notes:

1 The operands

- Have no effect when output spooled to tape is punched later
- Take effect again when the tape-spooled output is loaded to the punch queue by POFFLOAD LOAD|SELECT.

In addition to the above listed 'native' VSE/POWER operands, you can define additional JECL output operands of your own in autostart DEFINE statements and specify them in the * $$ PUN or * $$ LST statement. For the sequence of interpretation and for more information, see "DEFINE: Specifying User-Defined Output Operands" on page 488.

BLDG=number
Specifies the programmer's building number to be placed in the punch account record. You can specify a number of up to eight alphameric characters. If omitted, VSE/POWER uses the BLDG= specification, if any, in the * $$ JOB statement.

BTRNC=YES|NO
The operand applies to the recording of spooled output on the data file.

YES
To truncate trailing blanks at the end of every record. This is the default. This function saves disk space on the data file.

NO
specifies that NO blank truncation should take place. This may be desired when the punch queue entry is to be processed by the Spool-Access Support GET service and **all** originally spooled characters should be passed to the requestor.

See also "Recording of Spooled Data on the Data File" on page 179.

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 or any numeral 0 through 9.

If you omit the operand, VSE/POWER assigns the class that was specified as default output class in the PSTART command for the static partition used to execute your program. For a dynamic partition, the default output class is 'A', or it is the execution class itself, if the SET DYNOUTCL=DYNCL autostart statement has been activated.

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.

DEPT=number
Specifies the programmer's department number to be placed punch account record. You can specify a number of up to eight alphameric characters. If omitted, VSE/POWER uses the DEPT= specification, if any, in the * $$ JOB statement.

DEST=node_id| (node_id,user_id)| (node_id,user_id,subsystem)
Any specification in this operand overrides the specification in the PDEST operand of the * $$ JOB statement.

If you specify REMOTE=remote_id 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 ID to the ID specified in REMOTE=remote_id.

Specify for:

node_id
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 (specify DEST=(,user_id)) if the output is destined for a user at the node of origin, namely:
  - a user at the local node, provided the job has been read in a this node (and has not been received via networking) for execution, or
  - a user at the origin node, provided the job has been read in at, for example, origin Node A and has been sent for execution to another node, for example, Node B.

user_id
One of the following:

- The applicable user ID. For a definition of user_id, see "Coding Rules for VSE/POWER Commands" on page 193.

  If you specify a user ID (other than R000 or 000), 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. Such a spool entry can be manipulated only by the origin user ID. If Spool Access Protection is active, the user ID accessing or manipulating the entry must be authenticated (see "VSE/POWER Spool Access Protection" on page 14). In this case, if the entry is to be accessible by non-authenticated users, SECAC=NO should also be specified.
- PWR$JOB together with the ID 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=node_id).

  Notes:

  1. Punch jobs with a specification of DEST=(,PWR$JOB) do *not* inherit VSE security as do DISP=I jobs.
  2. When punch jobs with a specification of DEST=(,PWR$JOB) still contain a * $$ JOB statement, a receiving VSE/POWER system will *not* process this statement as is done for DISP=I jobs.

  Note: Specifying *only* a node_id for the DEST operand will nullify any default user ID and make the output processable on a local punch device.

subsystem
Applies if, for node_id, you specified the name of an OS/390 MVS node.

For subsystem, give the name of the external writer that is to process the spooled output.

DISP=D | disposition

For local 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) in a work file 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 (unless JNM is specified) but always a different job number as the entry that produces the output. The new entry's class, priority, system ID, user information, destination, and password 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 and with the value of TDISP as the transmission disposition. Your output therefore is put into the reader queue at the final destination.

Note: Refer to "Important Specifications for Output Spooling" on page 147 for hints in solving problems for problems with DISP=I. If the output is to have different spooling characteristics than the default values (e.g.CLASS) then these values have to be included on the * $$ PUN statement.

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.

The user program owns the DISP=N device until another * $$ PUN statement for the same device or the end of the VSE/POWER job comes.

T Spool the output to tape. For dependencies, see also the TADDR= operand on page 457.

This operand is ignored if specified in a * $$ PUN statement used to segment output via the IPWSEGM KEEP=YES macro.

If you specify DISP=T, the following operands are ignored:

```
PURGE=nnn       RBC=       RBS=
```

To start punching the output spooled on tape, use the PSTART PUN,uraddr,X'tapeaddr' command. For * $$ PUN operands that do not take effect at punch time, see "Format 2: Spooling Punch Output to Tape" on page 448.

To load the output spooled on tape to the punch queue, use the POFFLOAD LOAD|SELECT command. Reloaded entries will get disposition D.

For further information on dispositions, see also Appendix A, "VSE/POWER Disposition Codes," on page 497.

DIST=distcode

For *distcode* specify the distribution code (one up to eight alphameric characters) for output created under VSE and punched by a VM punch task. The distribution code is passed with the CP CLOSE command to VM; the distribution code in the VM-CP directory remains, however, unchanged.

The distribution code is also passed to other nodes; the code is interpreted, however, only by PNET nodes.

The distribution code will be shown in the PDISPLAY command if for output queue entries the option FULL=YES is set.

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.

JNM=jobname

For *jobname* specify the name VSE/POWER should assign to the punch queue entry. You can specify a name consisting of two to eight alphameric characters. For a list of what VSE/POWER considers an alphameric character, see "Coding Rules for VSE/POWER Commands" on page 193.

Do **not** use any of the following as a job name:

```
ALL      LOCAL
FREE     RJE
HOLD     STATUS
```

If output segmentation is specified using the operand RBS, each segment gets the name specified as *jobname*.

If you omit the operand, the output gets the name specified in the * $$ JOB statement.

JSEP=n | (n,option) | (,option)
The operand specifies the number of separator cards wanted for the job and if separator cards are to be inserted between every 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 VSE/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, see the discussion of the JSEP operand under "POWER Generation Macro" on page 57.

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 VSE/POWER generation macro.

LTAPE=YES | NO
If specified, VSE/SAM is used during OPEN, CLOSE and end-of-volume (EOV) tape processing.

This operand must follow the DISP=T operand.

YES
Indicates VSE/SAM labeled tape processing.

NO
Indicates VSE/SAM unlabeled tape processing.

If TLBL= is specified in addition to LTAPE=NO, an unlabeled tape will still be processed using VSE/SAM. However, VSE/POWER will pass the tfilename value to VSE/SAM as for labeled tape processing. This operand value is normally needed only by Tape Management Systems which might use the // TLBL statement to indicate tape information, such as tape density. The TLBL= value then applies to all unlabeled tapes produced.

Note: Unlabeled tape processing may result in more than one output tape being produced; however, VSE/POWER processes only a single unlabeled input tape.

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.

PROGR=name
Specifies the name of the person (programmer) responsible for the job. This name is placed in the punch account record. You can specify a name of up to 20 alphameric characters. If the name contains special characters, such as a comma or a blank, you must enclose the name within a pair of apostrophes. An apostrophe within the name must be entered as two apostrophes. If omitted, VSE/POWER uses the PROGR= specification, if any, in the * $$ JOB statement.

PRMODE=value
This operand applies to the punching of output and specifies the printer processing mode. Any alphameric value with the length of 1-8 characters is accepted. VSE/POWER does not interpret this operand for its own punching. Instead, the specified value is handed over to spool-access support users and to other networking nodes. Because this operand affects the components which process the output entry, specify values which are meaningful to these components. The value will be saved in the general section of the data set header record (DSHR).

For the handling of the 'native' operand 'PRMODE' at the same time as the 'user-defined' operand, see "DEFINE: Specifying User-Defined Output Operands" on page 488.

PURGE=NO|nnnn
This operand applies to the conditional suppression of punch output for the device selected by the PUN= operand. For *nnnn* specify a return code value of 0-4095 (greater values are reduced to 4095).

The output spooled for this device will not be added to the specified queue if a job has terminated without cancel condition and the maximum program return code (collected over several job steps) is either
not provided at all, or
does not exceed *nnnn* (or 4095, whichever is smaller).

If output is segmented by multiple * $$ PUN statements or the IPWSEGM or SEGMENT macro, cancel and maximum return code conditions effective at the point of segmentation are used. Note, however, that return codes are not made available before Job Control end-of-job time.

In context with other * $$ PUN operands, the PURGE=*nnnn* specification

- Is ignored with DISP=T/N/I or RBC=m(>0)
- Is ignored for writer-only partitions
- Resets count-driven segmentation to RBS=0
- Resets 'limit exceeded' message to RBM=(0,0)

If output is segmented implicitly by a SETPRT or LFCB request, the PURGE=nnnn specification is not effective.

Specify NO or omit the operand if the spooled output should be made available in the PUN or XMT queue.

See also "Unrecorded Processing" on page 120.

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 every 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 startup. 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.

For more details, refer to "Important Specifications for Output Spooling" on page 147.

PWD=password
Specify this operand if you wish to have your output protected by a VSE/POWER 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. It is recommended to specify this operand last. See "Correction of JECL Errors" on page 405.

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 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 PUN,CDISP=X command and then take appropriate action.

For norbc, you can specify a number from 1 to 999,999. If you specify 0, no checkpoint is taken. The output (incomplete queue entry) then is 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 at output spooling time before the operator gets the "limit exceeded" message 1Q52I. Specify a number from 0 to 999,999. A specification of 0 for *norbm1* and 0≤*norbm2*≤999999 means that no warning message is given at all.

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 VSE/POWER generation macro.

norbm2
The additional number of punch records to be intercepted every time before VSE/POWER reissues the above mentioned message. Again, you can specify a number from 0 to 999,999. A specification of 0<*norbm1*≤999999 but 0 for *norbm2* means that message 1Q52I is issued once when *norbm1* is reached but no further warning message is created.

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 VSE/POWER generation macro.

RBS=norbs
The operand specifies the number of punch records processed 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 999,999. 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 VSE/POWER generation macro.

The operator gets a message whenever a count-driven segmentation occurs.

Note: If your job produces more than 127 segments, VSE/POWER assigns a new job number to each following group of 127 segments and starts with suffix 1 for each new job number.

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 are not lost.

The operand is ignored if DISP=T is specified.

This operand is ignored if specified in a * $$ PUN statement used to segment output via the IPWSEGM KEEP=YES macro.

REMOTE=remote_id
For remote_id, specify the remote ID, a three-digit number, to which the output is to be routed.

Specify 0 for remote_id 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.

ROOM=number
Specifies the programmer's room number to be placed in the punch account

record. You can specify a number of up to eight alphameric characters. If omitted, VSE/POWER uses the ROOM= specification, if any, in the * $$ JOB statement.

SECAC=
Overrides the SECAC mode inherited from the job. SECAC=NO indicates that the output is *required not* to be spool access protected. If active, VSE/POWER Spool Access Protection is ignored.

If this parameter is not specified, and VSE/POWER Spool Access Protection is active, and the origin job did not specify `* $$ JOB SECAC=NO`, then by default the output will be Spool Access Protected *if possible*. To obtain Spool Access Protection eligibility, the output spool entry will be tagged with the user ID(s) obtained from:

1. The origin userid of the job if available
2. The target userid obtained from the DEST=(...,userid) operand (or the LDEST/PDEST operand of the `* $$ JOB` statement), if available

Otherwise, the entry is not eligible for Spool Access Protection.

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 VSE/POWER generation macro).

SYSID=N
If the output is to be available on any of the sharing systems, meaning SYSID=Null.

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 cuu 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 *z/VSE System Control Statements*.

At completion of the tape spooling function with user specified density, VSE/POWER resets the density specification to the standard one defined at IPL or by a later permanent ASSGN statement.

The streaming-mode specifications for the IBM 8809 (30 and 90) are not supported by VSE/POWER.

The specification of tapeaddr forces a disposition of T.

If DISP=T is specified and tapeaddr is omitted, VSE/POWER prompts the operator by message 1Q55D for a tape-unit address when your job is being executed.

For labelled tape processing only one output file can be written to a single tape. For unlabelled tape processing, two or more output files can be written to a single tape. Multivolume segmentation is supported when a tape becomes full.

This operand is ignored if specified in a * $$ PUN statement used to segment output via the IPWSEGM KEEP=YES macro.

TDISP=D|disposition
For transmission disposition, specify how the punch output is to be handled when placed into the transmission queue on your node. Your specification may be one of the following:

D Delete after transmission. The output entry is automatically scheduled for transmission according to its priority. When transmission is complete, VSE/POWER deletes the output entry from the transmission queue.

H Hold the output entry in the transmission queue. It is not dispatched for transmission until the operator either:
- Changes the entry's disposition to K or D by way of a PALTER command, or
- Issues a PRELEASE command for the output entry.

K Keep after transmission. Your output is automatically scheduled for transmission according to its priority. When the transmission is complete, VSE/POWER retains the output entry in the transmission queue with a disposition of L.

L Leave the output entry in the transmission queue. It is not dispatched for transmission until the operator either:
- Changes the entry's disposition to K or D by way of a PALTER command, or
- Issues a PRELEASE command for the output.

For further information on dispositions, see also Appendix A, "VSE/POWER Disposition Codes," on page 497.

TLBL=tfilename
For tfilename specify one to seven characters of a tape label `// TLBL filename...`. It is assumed that the // TLBL statement is available to the VSE/POWER partition. For further details see "Labeled Tape Support" on page 152. For the syntax of the TLBL statement, see *z/VSE System Control Statements*.

If specified and LTAPE= is not specified, LTAPE=YES is assumed.

This operand must follow the DISP=T operand.

For more information about using the TLBL statement see the description of LTAPE.

UINF=user_info (or USER=user_info)
The keywords UINF= and USER= are synonymous. Specify up to 16 bytes to be used as user information. The specified character string replaces whatever character string was specified in the * $$ JOB statement (if any) for the job. The user information is retained in an internal field, left-justified with trailing blanks, and it is:
- displayed when the operand FULL=YES is specified in the PDISPLAY command via the U='...' display field
- included in Punch account records

It can be altered using the UINF= operand of PALTER and searched for in queue manipulation commands by the CUINF= operand.

Syntax conventions:

- For *user_info*, specify up to 16 characters or blanks. If the character string includes a blank or comma, the entire string must be enclosed in a pair of single quotes; otherwise, the first blank or comma is interpreted as a delimiter by VSE/POWER. If you are nesting single quotes inside single quotes, you must specify them as two adjacent single quotes, as shown under "Examples of the PALTER Command" on page 211.
- It is recommended to specify characters whose hexadecimal representation is not affected by uppercase conversion. All values are converted to uppercase, as described under "Uppercase Conversion of Characters in JECL/JCL Statements" on page 405.
- For a display of the recorded user information, refer to "Format 1-4: PDISPLAY LST,FULL=YES" on page 257.

* $$ RDR: Adding 3540 Files to Input Jobs

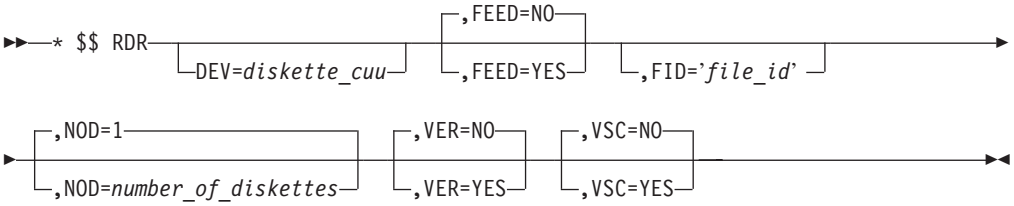
The * $$ RDR statement may be used to insert data from a diskette into an input stream for spooling to the reader queue under control of a local task. The data may be inserted, for example, into an input stream from one of the following:

A card reader or an IBM 3540
A remote terminal
Another partition

A diskette file inserted by an * $$ RDR statement may not itself contain another * $$ RDR statement.

In a writer-only partition, this statement is ignored. The statement is interpreted at read-in spooling time. If it is in error, the job is flushed. For more information, see "Coding Rules for JECL Statements Including Continuation" on page 408.

Format of the Statement



DEV=diskette_cuu
The operand does not apply to an * $$ RDR statement for a SYSIN file.

For diskette_cuu, specify the cuu 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. 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 VSE/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

  Here, 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. If it does not find the file, it prompts the operator to mount the correct diskette or to cancel the job.
- The command's format is: PSTART RDR,uraddr

  In this case, the command does not specify the address of a diskette unit. VSE/POWER then:

  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 with 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 with 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 address.

* $$ SLI: Including a Library Member into a Job Stream

To include a library member into a VSE/POWER job stream, code a * $$ SLI statement that defines the named library member to be added. This member is subsequently referred to as an SLI member (**S**ource **L**ibrary **I**nclusion).

An SLI member must contain only fixed length 80 byte records.

VSE/POWER searches for an SLI member in one of the following:
- A VSE/AF sublibrary defined in the LIBDEF SOURCE search chain for the VSE/POWER partition
- One or more VSE/AF sublibrary(s) not in the VSE/POWER partition LIBDEF SOURCE search chain.
- VSE/ICCF libraries.

VSE/POWER includes the requested SLI member at the point it finds the * $$ SLI statement when the spooled job is executed.

The statement is ignored in a writer-only partition.

The statement is interpreted at job execution time. If it is in error, it can be corrected then. For more information, see "Coding Rules for JECL Statements Including Continuation" on page 408.

For examples on how to code SLI statements, see "Example for Inserting Via * $$ DATA Statement" on page 469 or "Example for Updating Via $SLIxnnn Statements" on page 469.

Rules for Coding the SLI Statement

Use of this statement is based on the VSE/POWER default or the specified values of either the MEMTYPE or the SUBLIB operand in the POWER generation macro. The job is flushed with message 1QC3I if VSE/POWER cannot find the requested SLI member or a security violation occurs.

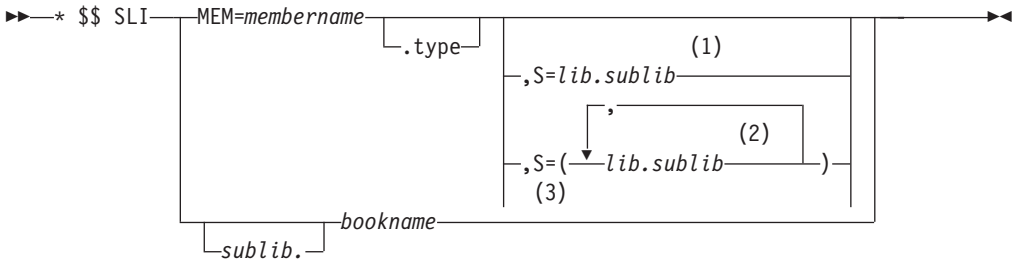
A VSE/POWER job may include more than one * $$ SLI statement. For rules on coding an SLI member, see "Rules for Coding an SLI Member" on page 466.

Format 1: Including a Member from a VSE/AF Sublibrary

This format allows to:

- Omit the S=lib.sublib operand. In this case, the VSE/POWER **partition** LIBDEF SOURCE search chain is searched for the member.
- Specify the S=lib.sublib operand. Then VSE/POWER searches the member in the user-defined order in up to 3 sublibraries which may either be included in the VSE/POWER partition LIBDEF SOURCE search chain or not.

  For examples on the search order of sublibraries, see page 465.



Notes:

1 *S*= is ignored (no syntax check is performed) for nested * $$ SLI members. The library definition chain that is defined in the first * $$ SLI statement will be used for all of the nested members that must be fetched.

2 Up to three sublibraries may be specified.

3 *S*= is ignored (no syntax check is performed) for nested * $$ SLI members. The library definition chain that is defined in the first * $$ SLI statement will be used for all of the nested members that must be fetched.

MEM=membername | membername.type

For membername, specify the name that was used to catalogue the SLI member. If this name begins with $$, VSE/POWER replaces the

- Second dollar sign by the second character of the partition-id, if the job executes in a **static** partition: $$ becomes $0 for BG, $1 for F1, ...$B for FB (foreground 11).
- First dollar sign by the first character of the partition-id, if the job runs in a **dynamic** partition. That means, the first dollar sign is replaced by the dynamic class character: $$ becomes C$ for dynamic partitions C1,C2, ... CW, or R$ for all partitions started for the dynamic class R.

For type, specify the member 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. The MEM operand must be coded only once. (Usually, keyword operands can occur more than once, in which case VSE/POWER uses the last specification.)
2. PHASE or DUMP may **not** be used as a member type.
3. The membername must follow the rules which the Librarian requires for membernames.

sublib.bookname | bookname

This operand is supported for compatibility reasons only. If you use the operand, VSE/POWER uses your specification for bookname as the membername specification. It uses your 'sublib.' specification, if present, as the .type specification.

If you omit the 'sublib.' specification, VSE/POWER uses the type you specified for the POWER generation macro operand SUBLIB, or its default value.

S=lib.sublib

Identifies the name of one or more sublibrary(s) in which a member is to be found for source library inclusion and in the order to be searched.

A maximum of three sublibraries may be specified. The sublibrary(s) need not be in the VSE/POWER partition LIBDEF SOURCE search chain.

lib
identifies the name of the library.

sublib
identifies the name of the qualified sublibrary to be searched for the member as specified in the MEM operand.

Notes:

1. When sublibraries are not included in the VSE/POWER partition LIBDEF SOURCE search chain, an implicit LIBDEF OPEN is done at job execution time. Then the DLBLs for the specified libraries are searched in the following order:
   a. VSE/POWER user label area
   b. VSE/POWER partition standard label area
   c. System standard label area
2. For performance reasons, it is recommended that any partition issue a LIBDEF statement for a library containing a specified source sublibrary. This avoids that the job which contains the * $$ SLI statement has to wait on resources for the library open process, in particular for VSAM libraries.

   If the library which contains the sublibrary has not been already opened by a partition prior to the * $$ SLI statement execution, a severe performance degradation may take place while processing the * $$ SLI statement; for example, when accessing a VSAM library, the open request Getvis storage costs are very high.

Example of the search order of sublibraries

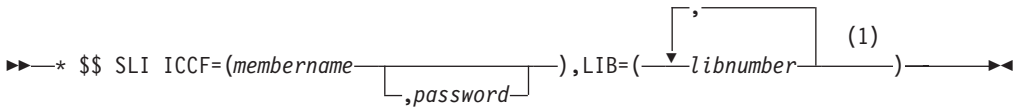
The following example demonstrates the usage of the S operand:

```
* $$ SLI MEM=MOD1,S=(PRIMARY.SMITH,PRIMARY.TESTED,PROD.BASE)
```

In the above example, the sublibraries are searched in the following order:

1. PRIMARY.SMITH
2. PRIMARY.TESTED
3. PROD.BASE

Format 2: Including a Member from a VSE/ICCF Library



Notes:

1 Up to three library numbers may be specified.

ICCF=(membername) | (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 to be included into the currently spooled job stream. This name can consist of alphameric characters in any combination.

For a list of what is, for VSE/POWER, an alphameric character, see page 194. Note that VSE/POWER is more restrictive in its rules for admissible member names than VSE/ICCF.

A * $$ SLI statement with this operand may be part of a job that is being read by a local reader task (from a diskette unit or another partition, for example). In this case, VSE/POWER requires also a valid // ID statement or * $$ JOB SEC statement if the ICCF library to be accessed is neither a common nor a public one. And the // ID statement or * $$ JOB SEC statement is only valid if z/VSE was initially loaded with SEC=YES.

For information on accessing ICCF library members, refer to *VSE/ICCF Administration and Operation*, "Submit to Batch (DTSSUBMT) Utility Program. "

password
The password by which the member is protected, if there is such a password.

If the * $$ SLI statement is contained in a VSE/POWER job that provides a userid by the FROM operand of the * $$ JOB statement, then consider the ICCF access rights as described for the FROM operand on page 419.

LIB=(libnumber)
The operand specifies the VSE/ICCF controlled library (or libraries) to be searched for the named SLI member.

For `libnumber`, specify the number of every 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 containing the following JECL statements will be processed as indicated at job execution time:

- * $$ LST/PUN statements - fully supported
- * $$ CTL/RDR/EOJ statements - ignored
- * $$ JOB statement - partly supported, effective only for certain operands (see "Handling Read-in JECL Statements at Execution Time" on page 403).
- * $$ SLI statement - partly supported due to the S=lib.sublib operand being ignored for nested SLI members.

A * $$ LST/PUN/JOB statement **within** an SLI member can be continued from one line to the next.

A * $$ SLI statement within an SLI member **cannot** be continued from one line to the next, and instead the statement is flagged at execution time by message 1Q49I and either operator correction is requested by message 1R33DI or predefined action is taken according to SET 1R33D=FLUSH|IGNORE.

If the member is to include * $$ LST/PUN/SLI/JOB statements, code them with the * $*x* LST, * $*x* PUN, * $*x* SLI or * $*x* JOB prefix within the job that catalogues the library member. In the modified prefix, *x* can be any character other than blank or $.

Note: If the member containing the data which is to be included by the * $$ SLI statement is cataloged with a job using a * $$ SLI statement as well, the

prefix of the LST, PUN, DATA or SLI statements have to be ..$$ instead of * $x. If the * $x prefix is used, the statement will be executed at catalog time, because VSE/POWER interprets it as a statement belonging to the * $$ SLI statement of the catalog job.

At catalog time the prefix *..$$* is replaced by the prefix * *$$* and a * $$ JOB/LST/PUN/SLI statement (with the prefix * *$$*) is written into the library member. If the library member is included in a job stream, * $$ LST/PUN/SLI statements are processed, whereas a * $$ JOB statement is ignored. However, a * $*x* JOB statement within an SLI member is partly supported, effective only for certain operands (see "Handling Read-in JECL Statements at Execution Time" on page 403).

* $$ DATA statements (if used) must follow the * $$ SLI statement in the same sequence as that of the corresponding * $$ DATA statements in the member. Data that you insert by means of an * $$ DATA statement must be delimited by a /*$SLI statement or by /* or /&. For details and restrictions, see "* $$ DATA: Inserting Data into an SLI Library Member" on page 410 and also "Example for Inserting Via * $$ DATA Statement" on page 469.

Update statements (with $SLIxnnn sequence number) must follow the * $$ SLI statement in your job stream in the same sequence as that of the corresponding statements (with sequence numbers) in the SLI member. For details and restrictions, see "Changing an SLI Member During Execution."

Nesting of SLI statements is possible. That is, a retrieved SLI member may include SLI statements. The following rules apply in this case:

- An SLI nesting error occurs if duplicate member names are used, even when the members are in different sublibraries. While nesting is possible to a virtually unlimited level, be aware that every additional level adds to the complexity of your application. In addition, for every level of nesting, VSE/POWER needs a certain amount of virtual storage.
- When SLI members are nested, e.g., from level-1 to level-2 and back again to level-1, statements of different levels may also be modified (see "Updating Member Statements Using $SLIxnnn Sequence Numbers" on page 468). In this case, the comparison of sequence numbers is done in the sequence of stepping through the levels, i.e., numbers of level-1 up to call of level-2, then numbers of level-2 up to return to level-1, followed by numbers of the remaining part of level-1.

Note: An SLI member may invoke a VSE procedure, but a VSE procedure may not invoke an SLI member.

Cataloging an SLI Member

For details on how to catalog a member in a sublibrary and how to define this sublibrary in a sublibrary-search chain for a partition, consult the manual *z/VSE Guide to System Functions*.

Changing an SLI Member During Execution

There are two basic methods of making dynamic changes to an SLI member:

- Supplying statements to be inserted into the member, based on an * $$ DATA statement.
- Supplying new and changed statements to update the member based on a sequence number in both the member and update statements.

Inserting Statements in Place of * $$ DATA Statements

See "* $$ DATA: Inserting Data into an SLI Library Member" on page 410 and "Example for Inserting Via * $$ DATA Statement" on page 469.

Updating Member Statements Using $SLIxnnn Sequence Numbers

Statements may be dynamically added to, deleted from, or replaced in an SLI member, based on the value of a sequence number present in columns 78 through 80[15] of both the member and update statements. These sequence numbers control the order of the statements in the resulting job stream. To make such changes, submit the 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 This statement is to be inserted behind the member statement with the same sequence number or before the next member statement with a higher sequence number.

B This statement is to be inserted before the member 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 before the next member statement with a higher sequence number (or logically appended if the sequence number exceeds the last number present in the member).

nnn =
A statement-sequence number.

The following rules apply to updating:

1. When an update statement has a sequence number higher than all statements of the SLI member, the update statement is actually processed at its place in the job stream rather than being first appended to the member. The resulting job stream is the same, however.
2. When an update statement requests "deletion", the contents of the update statement are not checked by VSE/POWER.
3. When an update statement results in replacement of a certain sequence number or in insertion before or between sequence number positions of the member being updated, the update statement *must not itself* be an * $$ SLI statement (although members can also contain * $$ SLI statements; see "Rules for Coding an SLI Member" on page 466). Otherwise, VSE/POWER will flush the job with message 1QC0I RC=0032. However, * $$ SLI statements can be included in update statements *behind* the last statement of the SLI member, because these statements are effectively processed as part of the input job stream, as noted in point 1.

For details, see "Example for Updating Via $SLIxnnn Statements" on page 469.

15. To apply sequence numbers to existing VSE library members, you may use // EXEC LIBR with the 'UPDATE mn.mt SEQUENCE=' command.

Example for Inserting Via * $$ DATA Statement

The following example shows the use of the * $$ SLI and * $$ DATA statements. The example applies accordingly to members that are to be retrieved from a VSE/ICCF controlled library.

Job to retrieve a member named ASSM and to insert data into this member:

```
Job from card reader            Cataloged member ASSM

* $$ JOB   INSERT
* $$ SLI   MEM=ASSM                  // JOB INSERT
* $$ DATA  INPUTA                    // EXEC ASSEMBLY
           POWER ACCOUNT=YES         * $$ DATA INPUTA
           END                       /&
/*
* $$ EOJ


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 for Updating Via $SLIxnnn Statements

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

```
* $$ JOB   ASSEMSLI
// JOB     ASSEMBLE                XMPL0010
// ASSGN   SYSPCH,182              $SLI0020   updated (replaced)
// EXEC    ASSEMBLY                XMPL0030
DEFAULT    POWER ACCOUNT=YES                  inserted
           END                                inserted
/*
// MTC     WTM,SYSPCH              XMPL0050
// MTC     WTM,SYSPCH              XMPL0060
// MTC     REW,SYSPCH              XMPL0070
/&                                 XMPL0080
// JOB     CATALOG                 XMPL0090
// ASSGN   SYSIPT,182              $SLI0100   updated (replaced)
// ASSGN   SYS010,183              $SLIA100   updated (inserted)
// EXEC    LIBR                    XMPL0110
   CATALOG DEFAULT.OBJ             XMPL0120
/*                                 XMPL0130
// MTC     RUN,182                 $SLI0140   updated (replaced)
/&                                 XMPL0150
// JOB     LINK                    XMPL0160
// OPTION  CATAL                   XMPL0170
       INCLUDE DEFAULT             XMPL0180
/*                                 XMPL0200
// EXEC    LNKEDT                  XMPL0210
/&                                 XMPL0220
* $$ EOJ
```

Chapter 7. VSE/POWER Autostart Statements

Overview

VSE/POWER offers an automatic startup function, called autostart. The following autostart statements are available to start VSE/POWER automatically:

SET defines various control values.

DEFINE
defines any additional JECL output operands

FORMAT
shows if the VSE/POWER files are to be formatted

READER
defines an input spooling device

PRINTERS
defines list-output spooling devices

PUNCHES
defines punch-output spooling devices

Autostart statements always start in column 1. They must be submitted in uppercase via the SYSIPT device assigned to the VSE/POWER partition.

If an error is detected in the operands of a SET or DEFINE statement, it is logged on the console with failure message 1Q13I (for SET) or 1Q09I (for DEFINE), and no correction by an operator is possible. You can also nullify these autostart statements by simulating a Job Control comment when you let the statements start with '* '.

If an error is detected in the operands of a FORMAT, READER, PRINTERS, or PUNCHES statement, VSE/POWER expects the corrected statement to be entered from the console. For example, if you specify FORMAT=xx, VSE/POWER requests by message 1Q11D a format decision from the console- which is an easy method to switch dynamically between warm and cold start.

Autostart statements normally are 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:

```
(PSTART)    PLOAD
PALTER      PRELEASE   PACT
PDISPLAY    PSTOP      PVARY
```

Note: Commands not mentioned in this list, or incorrectly specified commands, or commands that have been commented (* PLOAD, for example) are not recognized as commands and are ignored without any warning message.

In an autostart procedure, the autostart statements have to be arranged in *this* sequence, following the // EXEC POWER statement:

1. **SET**, if any [16]
2. **DEFINE**, if any [16]
3. **FORMAT**, if any [17]
4. PSTART for a static 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.

Re-Display of Specified Autostart Statements

Use the PDISPLAY AUSTMT command, described under "Format 14: Displaying Information About Used Autostart Statements" on page 245, if you are:

- In doubt if intended startup statements become active
- Cannot remember a certain used autostart statement
- Cannot relate your current system behavior to the autostart overwrites provided by the SET statements.

16. If any SET or DEFINE statements are placed erroneously behind the FORMAT statement, VSE/POWER ignores them. However, the operator is warned by message 1Q06I.

17. VSE/POWER needs a formatting decision. If (after SET or DEFINE) no FORMAT= statement can be found, you get prompted by message 1Q11D.

SET: Setting VSE/POWER Startup Control Values

Values defined by way of a SET statement override the definitions made in the POWER generation macro or provide additional start-up values.

AUTONAME
To enable automatic flushing or holding of any series of job statements that are read in as an AUTONAME job.[1]

CCWCKX
To perform an extended check of the CCW operation.[1]

CCW01
To suppress a 'Write No Space' for VM list tasks.[1]

CH9CH12
To re-establish FCB Channel 9/12 overflow posting for VSE/POWER local list tasks.[1]

CONFIRM
To request confirmation, by reply to message 1QZ3D, that a given command should be processed by VSE/POWER.[1]

DBLK To define the size of the Data Block record of the data file.

DLSEP
To force identical separator pages and drop the 'last-one' separator page for all list tasks to be started.[1]

DLSEPSAS
To force identical separator pages and drop the 'last one' separator page for all Spool Access Support (SAS) orr Device Service (DST) tasks.[1]

DYNAL
To prefer static partition support versus allocation of dynamic partitions.[1]

DYNOUTCL
To modify the default output class of dynamic partitions.[1]

FCB To have VSE/POWER use a default FCB at output spooling time, instead of the LTAB of the POWER macro.[1]

FROM
To reduce ICCF access rights for jobs specifying the FROM operand in the * $$ JOB statement.[1]

GETVQFL
To override startup default location for creating the queue file copy. [1]

HOLDCL
To override the default disposition and class of $LSTnnnn or $TAPnnnn output entries.[1]

IGNREC
To change the disposition of a queue entry to DISP=Y when it has been flagged by message 1Q4LI (local processing) or 1Q4KI (remote processing). [1]

INTFLUSH
To transfer the remaining part of a job to Job Control after flushing due to VSE/POWER-internal reasons.[1]

ISEP To enforce identical separator pages for all printed output of local list tasks.[1]

ISEPSAS
To enforce identical separator pages for all output requested by Spool Access Support (SAS) or Device Service (DST) tasks[1]

JCMQ To increase the default size of the Job Completion message queue. [1]

LOGEQNO
To ignore suppression of SYSLOG data.[1]

LONGCMD
To restrict usage of the specified command to the long form only.[1]

LST0DAT
To suppress list queue entries which contain non-printable data. [1]

NORUN
To identify jobs in progress whenever the system fails.[1]

PDUMP
To ignore SETPRT requests originating from PDUMP and avoid segmentation of dumps.[1]

OPNDST
To raise the default time interval for reattempting PNET/SNA connections.[1]

OUTEXIT
To enable passing control to a user OUTEXIT phase for Spool-Access GET requests to the list and punch queue.[1]

PALTER
To change local and transmission disposition simultaneously.[1]

PNET To have VSE/POWER use a Network Definition Table (NDT) that is different from the one defined in the POWER macro

RSCSROOM
To transfer CP distribution code to VM/CP via PNET.[1]

SECAC
To enable or inhibit activation of Spool Access Protection of eligible spool entries.[1]

SECNODE
To switch to another name of the security zone than defined in the POWER macro.

SEGCHECK
To enable the RJE Workstation Program to do segmentation.[1]

SJECL To enable replacement of the alternate JECL prefix.[1]

SKIP To have VSE/POWER insert a skip to channel 1.[1]

SYSID
To replace the shared spooling system id specified in the POWER macro

WORKUNIT
To enable multi-processor support in VSE/POWER.[1]

1QZ2A
To suppress the '1QZ2A' message header of CICS and PSF messages.[1]

1Q30D
To request operator prompting at abnormal termination time.[1]

1Q41I To suppress the message '1Q41I' if the device type at spooling time intentionally differs from that used for actual print/punch.[1]

1R33D
To suppress operator prompting by message 1R33D when an incorrect JECL statement is found at execution time.[1]

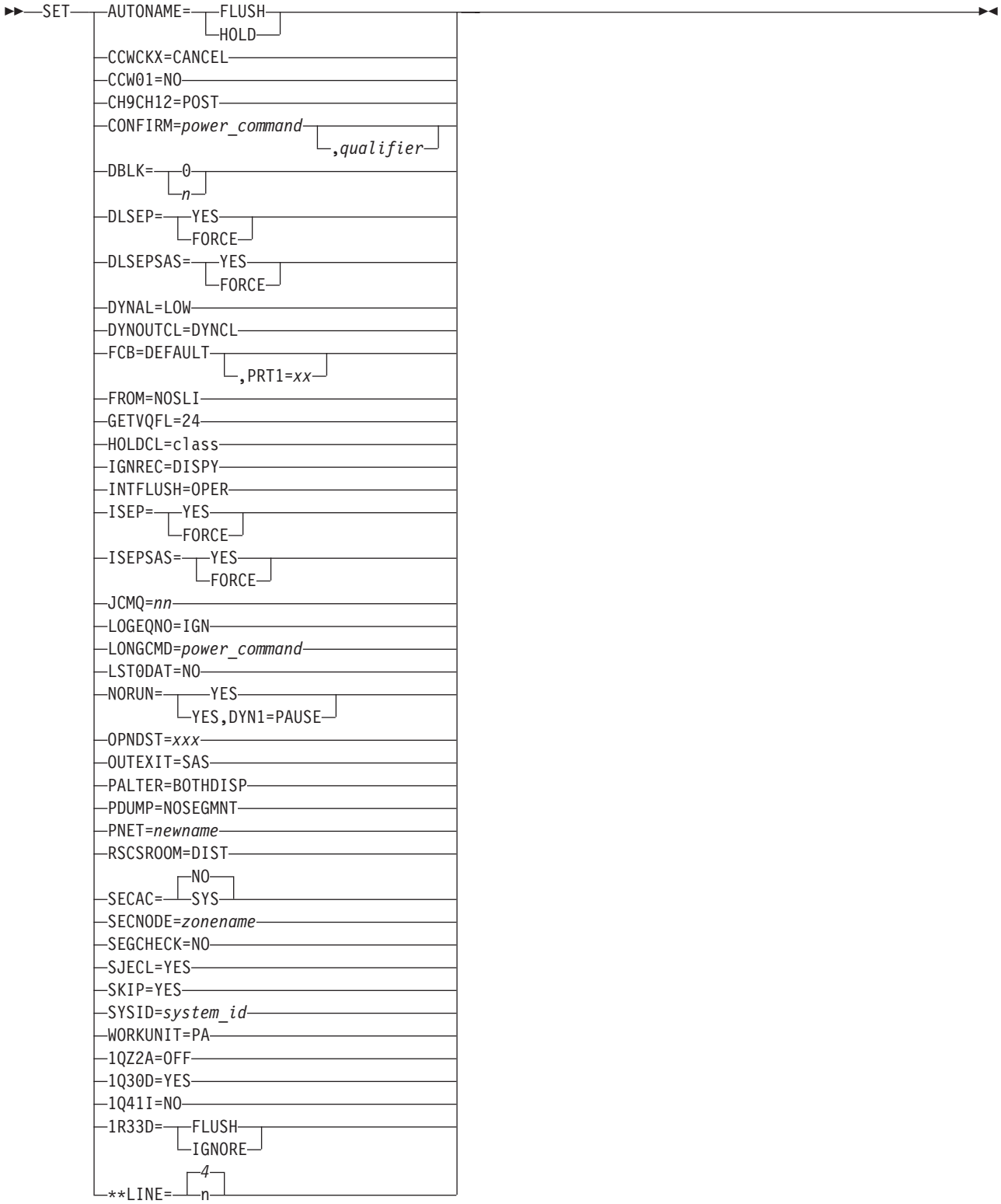
****LINE**
To define the number of asterisk lines printed over the perforation.[1]

Notes:

1. For these start-up control values, **no** corresponding POWER macro operands are available.
2. The majority of the SET statements are used to control job execution or output printing. Therefore, in shared spooling or PNET networking environments, make sure that not only the receiving node but more importantly the executing or printing node specifies the desired SET statements.

SET statements, if used in your autostart procedure, must follow immediately the `// EXEC powerphase` statement.

Format of the Command



AUTONAME=FLUSH|HOLD

This operand applies to reading of job input through local or remote readers or submission from other partitions. When read-in data does not start with a valid * $$ JOB or // JOB statement, VSE/POWER adds it as an AUTONAME entry to the reader queue with DISP=D and CLASS=A. For details, see "Job Streams Without JECL Delimiters" on page 399 and "Job Streams with JECL Delimiters" on page 401. With the following options, you can request VSE/POWER to prevent or delay processing of such AUTONAME jobs with questionable content:

FLUSH
will delete AUTONAME jobs immediately

HOLD
will add AUTONAME jobs to the reader queue with disposition HOLD.

In both cases, the local or remote operator, or the submitting program, is informed by message 1Q3HI.

Note: Whenever you use SET AUTONAME=, you should not specify AUTONAME as the intended jobname or submit * $$ JOB statements without the JNM= operand. Otherwise, the FLUSH|HOLD option will also apply to these jobs.

CCWCKX=CANCEL

The operand causes VSE/POWER to perform an extended check of the CCW operation codes when producing list output for SYSLST.

When specified, VSE/POWER issues the message 1R30I with RC = 0002 and cancels the job if one of the following (hexadecimal) CCW codes is used to write data to SYSLST:

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	00	-	-	-	-	-	-	-	-	-	-	-	0C	0D	0E	0F
1x	10	-	-	-	-	15	16	-	-	-	1A	-	1C	1D	1E	1F
2x	20	21	22	-	24	-	26	27	-	29	2A	2B	2C	2D	2E	2F
3x	30	31	32	-	-	-	36	-	-	39	3A	3B	3C	3D	3E	3F
4x	40	41	42	-	44	45	46	-	-	49	4A	4B	4C	4D	4E	4F
5x	50	51	52	-	-	55	56	-	-	59	-	-	5C	5D	5E	5F
6x	60	61	62	-	64	65	66	-	-	69	6A	-	6C	6D	6E	6F
7x	70	71	72	-	-	75	76	-	-	79	7A	-	7C	7D	7E	7F
8x	80	81	82	-	84	85	86	-	-	-	8A	-	8C	8D	8E	8F
9x	90	-	92	-	-	95	96	97	-	-	9A	-	9C	9D	9E	9F
Ax	A0	-	A2	-	A4	A5	A6	A7	-	-	AA	-	AC	AD	AE	AF
Bx	B0	-	B2	-	-	B5	B6	B7	-	-	BA	-	BC	BD	BE	BF
Cx	C0	-	C2	-	C4	C5	C6	C7	-	-	CA	-	CC	CD	CE	CF
Dx	D0	-	D2	-	-	D5	D6	D7	-	-	DA	-	DC	DD	DE	DF
Ex	E0	-	E2	-	-	E5	E6	E7	-	E9	EA	-	EC	ED	EE	EF
Fx	F0	F1	F2	-	-	F5	F6	F7	-	F9	FA	-	FC	-	FE	-

Figure 68. CCW Operation Code Table

Without specifying the operand CCWCKX, VSE/POWER writes for some of the above listed codes a data record containing one blank without issuing an error message, although an unmeaningful code has been specified.

It is recommended to use this operand. Without using this operand, the data addressed by the CCW may be ignored, which causes loss of possibly essential information. This operand has been introduced, to be compatible. Some existing programs might not have cared about losing information and can not be corrected due to missing source code.

CCW01=NO
Use this operand to avoid misalignment of forms for list tasks that are started with option VM or SP.

When printing jobs with different forms it may happen that the next form is misaligned. This is caused by an X'01' operation-code CCW, which is inserted by VSE/POWER at begin of a list queue entry to test if the printer is ready. For the D/T4248 printer, a 'SET CLEAR PRINT' op-code is inserted instead of a X'01' CCW.

If CCW01=NO is set, an X'01' or 'SET CLEAR PRINT CCW is suppressed before the LOAD FCB request, when the appropriate list task is started with option VM or SP.

CH9CH12=POST
Use this operand to re-establish FCB channel 9/12 overflow posting for VSE/POWER local lists tasks, because it may be required by OEM code. Posting has been standard for all releases previous to VSE/POWER 6.1.2.

Starting with VSE/POWER 6.1.2, the printing performance of a local list tasks is improved. Print I/O requests with their 'WRITE' command-chain are no longer interrupted when FCB channel 9 or 12 overflow occurs. That means, the I/O requests do not have to be re-scheduled by the list task.

Overflow posting is not needed by a VSE/POWER list task at print-time, because at output spool-time VSE/POWER has already signalled the channel 9/12 overflow conditions to the executing spooled partition.

CONFIRM=power_command | power_command,qualifier
Use this operand to request confirmation, by message 1QZ3D, that a command (long or short form) should be processed by VSE/POWER, provided the command is issued by the central (local) operator. Confirmation is requested by prompting message

```
1QZ3D PROCESS 'power_command'? CONFIRM WITH 'YES', ELSE 'NO'
```

For example, SET CONFIRM=PEND will prompt the operator with

```
1QZ3D PROCESS 'PEND'? CONFIRM WITH 'YES', ELSE 'NO'
```

Note: To request confirmation for multiple VSE/POWER commands, repeat the SET statement for each command.

If a qualifier is specified together with the VSE/POWER command, this specific command format will request confirmation. Currently, only the following *power_command,qualifier* combination is accepted by VSE/POWER:

- `PSTOP,PART`
- `PRELEASE,QUEUE,ALL`

Note:

1. As soon as you specify, for example, CONFIRM=PSTOP and CONFIRM=PSTOP,PART, confirmation message 1QZ3D will be issued twice for a 'PSTOP partition' command.
2. VSE/POWER has established a 'quasi' default 'SET CONFIRM=PDELETE,QUEUE,ALL' statement to protect against inadvertent deletion of all entries of a selected queue.

DBLK=0|n
With this operand you may overwrite the DBLK specification generated for the VSE/POWER phase being used for the current start up.

For an explanation of the data block (DBLK) size '0' and the accepted range 'n', refer to the description of the DBLK operand of the POWER macro on page 61.

In case of an invalid SET DBLK specification, the autostart statement is ignored and the generated DBLK value takes effect.

DLSEP=YES|FORCE
Use DLSEP if you want to drop the 'last-one' separator page for every list output of any started list task, and at the same time control the number of lines printed over the perforation. Via autostart statement SET **LINE, you can define the size of the separator page by specifying how many perforation lines (when DLSEP is not specified, normally 8 identical lines starting with '****') are printed. If 'SET **LINE=n' is not specified, a default of 4 perforation lines are printed. See also "Drop 'Last-One' and Print Shorter Separator Page" on page 184.

VSE/POWER takes the number of separator pages from the JSEP operand of the POWER generation macro and allows overwriting according to Table 7 on page 185.

For DLSEP=YES and JSEP=0 in the generation macro, VSE/POWER *will not print* a separator page.

For DLSEP=FORCE and JSEP=0 in the generation macro, VSE/POWER enforces *one* separator page (without an additional 'last-one').

Note: SET DLSEP and SET ISEP are mutually exclusive. If both are specified the second statement will be rejected by message 1Q13I.

DLSEPSAS=YES|FORCE
Use DLSEPSAS if you want to drop the 'last-one' separator page for every list output retrieved by a Spool Access (SAS) or a Device Service (DST) task, and at the same time control the number of lines printed over the perforation. Via autostart statement SET **LINE, you can define the size of the separator page by specifying how many perforation lines (when DLSEPSAS is not specified, normally 8 identical lines starting with '****') are printed. If 'SET **LINE=n' is not specified, a default of 4 perforation lines are printed. See also "Drop 'Last-One' and Print Shorter Separator Page" on page 184.

VSE/POWER takes the number of separator pages form the JSEP operand of the POWER generation macro and allows overwriting according toTable 7 on page 185.

For DLSEPSAS=YES and ISEP=0 in the generation macro, VSE/POWER *will not generate* a separator page.

For DLSEPSAS=FORCE and ISEP=0 in the generation macro, VSE/POWER enforces *one* separator page (without an additional 'last-one').

Note:
1. .SET DLSEPSAS and SET ISEPSAS are mutually exclusive. If both are specified the second statement will be rejected by message 1Q13I.
2. Any specified and tailored number of VSE/POWER separator pages is only passed via the Spool Access interface to a SAS or DST task, if the PWRSPL OPT=RETSEP parameter has been specified by the application for the GET-OPEN request.

DYNAL=LOW
This operand causes VSE/POWER to handle dynamic partition allocation with lower priority than servicing existing static partitions. See also "Attributes of Dynamic Partitions" on page 133.

DYNOUTCL=DYNCL
Use this operand to modify the default output class 'A' of dynamic partitions. Specifying DYNOUTCL=DYNCL causes VSE/POWER to start each dynamic partition with a default output class equal to its execution class - which is the dynamic class character itself.

FCB=DEFAULT | DEFAULT,PRT1=*xx*
This operand applies to spooling of list output to non-3800 printers. It causes VSE/POWER to use a default FCB when controlling the spooling of list output instead of the LTAB specified in the POWER macro. This affects only list output spooled by a job running in a partition if neither the FCB nor the LTAB operand has been specified in a * $$ LST statement. For further details, refer to "When Print Output is Misaligned" on page 116.

This operand does not apply to list output spooled via the Spool-Access support. Such output is written to the VSE/POWER spool file by **not** interpreting any FCB. If an FCB has been specified within the SPL, this FCB is used only at print-time.

The default FCB is selected as described in "System Buffer Load" in *z/VSE System Control Statements*.

For 3800 printers, FCB defaults are taken from the SETPRT command or statement or from the SETDF command.

Printer Type	FCB Name
1403	n.a.
3211 (PRT1)	$$BFCB
3203-5 (PRT1)	$$BFCB00
3289-4 (PRT1)	$$BFCB10
3262 (PRT1)	$$BFCB22
4245 (PRT1)	$$BFCB23
4248 (PRT1)	$$BFCB
4248 (Native) and 6262	$$BFCBWM

For example, the default FCB for a 6262 printer or a native 4248 printer is $$BFCBWM.

As printers of different type (for example 3211, 4245, 3203, 3289, 3262) are added during IPL time with the same device type PRT1, the operand

```
PTR1=xx
```

may be specified to define the device specific default FCB. The xx are the two last characters of the device specific default FCB, for example the default FCB for a 4245 printer is $$BFCB23 and therefore PRT1=23 has to be specified.

Notes:

1. Only one default FCB for PRT1 printers may be specified. If printers of two different device types have been added as PRT1, the same default FCB is used no matter which printer is currently used for the list output.

2. If the PTR1=xx operand is omitted, $$BFCB is used for printers added as PRT1.
3. The PTR1=xx operand does not influence the default FCB for printers added with a different device type code than PRT1.
4. If the default FCB is loaded, but VSE/POWER detects some error with this FCB, the message 1Q54I is issued and the LTAB is used as specified in the POWER macro.

FROM=NOSLI
Use this operand if the userid specified by the FROM operand of a * $$ JOB statement should not be used to gain access right to ICCF library members. For details on the ICCF interface refer to the FROM operand on page 419.

GETVQFL=24
Use this operand during testing of OEM software that accesses the queue file in storage by non-standard interfaces. When specified, VSE/POWER requests queue file storage below the 16 MB line, although Getvis-31 space has been allocated.

HOLDCL=class
Use this operand for $LSTnnnn and $TAPnnnn output entries to overwrite the default output class A and disposition D with the specified class value and disposition H. For class you can specify a character from 0–9 or A-Z. A $LSTnnnn entry is created by the PDISPLAY ALL,LST or the PDISPLAY TOTAL,LST,... command. A $TAPnnnn entry is created by the PDISPLAY entry,TAPE=cuu,OUT=LST command.

IGNREC=DISPY
During actual local printing or punching of a queue entry or while passing a queue entry to RJE (SNA or BSC) it may happen that an 'invalid' CCW op-code is detected.

This happens for

- **Local** processing, when the printer or punch type defined for spooling time is not CCW op-code-compatible with the printer or punch defined for actual printing or punching time.
- **Remote** processing, when the CCW op-code used at spooling time is not compatible with the CCW op-code tables used at RJE processing time.

If queue entries contain invalid CCW op-codes message 1Q4KI (remote processing) or 1Q4LI (local processing) will be issued, when the entry has been processed.

Specify IGNREC=DISPY in order to have output queue entries set to disposition Y, if these entries contain records flagged by message 1Q4KI or 1Q4LI.

If it is necessary to examine the ignored CCW op-codes of a queue entry with a non-dispatchable temporary disposition of Y:

- An appropriate list or punch task has to be started with the option SHOWIGN, and
- The queue entry has to be returned to its original disposition using the PALTER ...DISP=* command.

For further details refer to "If Output Records are Ignored" on page 115.

INTFLUSH=OPER
This operand applies to job execution. When a job is flushed by the operator, VSE/POWER passes the rest of the job stream to Job Control, provided Job Control is in the GOTO mode (ON $CANCEL GOTO label). To cause VSE/POWER to perform the same action in the case of internal flushing, use SET INTFLUSH=OPER. VSE/POWER will then pass the rest of the job to Job Control if Job Control is in the GOTO mode. For "internal flushing", refer to "Interaction with z/VSE Conditional Job Control Language" on page 108.

ISEP=YES | FORCE
Use this operand if you want to have 'identical' separator pages for list output by any started local list task. For a description of identical separator pages, refer to the ISEP operand of the PSTART command on page 362. For separator page specification, and overwriting, refer to Table 7 on page 185.

For ISEP=YES and JSEP=0 in the generation macro, VSE/POWER *will not print* a separator page.

For ISEP=FORCE and JSEP=0 in the generation macro, VSE/POWER enforces *one* separator page (and an additional 'last-one').

Note: SET DLSEP and SET ISEP are mutually exclusive. If both are specified, the second statement will be rejected by message 1Q13I.

ISEPSAS=YES | FORCE
Use this operand if you want to have 'identical' separator pages for all output entries retrieved by a Spool Access (SAS) or a Device Service (DST) task. For a description of 'identical' separator pages, refer to the ISEP operand of the PSTART (see 366) For separator page specification and overwriting, refer to Table 7 on page 185.

For ISEPSAS=YES and ISEP=0 in the generation macro, VSE/POWER *will not generate* a separator page.

For ISEPSAS=FORCE and ISEP=0 in the generation macro, VSE/POWER enforces *one* separator page (without an additional 'last-one').

Note:
1. .SET DLSEPSAS and SET ISEPSAS are mutually exclusive. If both are specified the second statement will be rejected by message 1Q13I.
2. Any specified and tailored number of VSE/POWER separator pages is only passed via the Spool Access interface to a SAS or DST task, if the PWRSPL OPT=RETSEP parameter has been specified by the application for the GET-OPEN request.

JCMQ=nn
This operand specifies how many fixed format messages can be queued for any user who submitted jobs via the spool-access support (SAS) option SPLGF1QM in function byte SPLGFB1. The queue size can range from 0 to 99 messages. If you omit the operand, the default of 20 messages is assumed. The statistics status report shown in Figure 7 on page 46 helps you to determine whether the default or specified queue size is adequate or not.

If JCMQ=0 is specified, fixed format messages can not be queued at this node and the messages are lost. However, when these messages are not destined for this node or system they are transmitted to the relevant target nodes or system.

For detailed information on the job completion message retrieval support (GCM Service) consult *VSE/POWER Application Programming*.

LOGEQNO=IGN
The operand applies to the suppression of SYSLOG data of VSE/POWER jobs that use the LOG=NO operand of the * $$ JOB statement. Specifying LOGEQNO=IGN causes the LOG=NO option to be ignored. Thus, all SYSLOG data is logged in your VSE/POWER system.

For details on the suppression of SYSLOG data, see "Unrecorded Processing" on page 120.

LONGCMD=*power_command*
Use this operand to restrict the usage of the selected VSE/POWER command to its long form. For example, SET LONGCMD=PDELETE in the autostart will reject the short-form command 'L queue,...' with message 1R9AI.

Note: To restrict multiple VSE/POWER commands to their long forms, repeat the SET statement for each command.

LST0DAT=NO
During creation of a LST queue entry through job execution, it may happen that only a specific printer control character without any data is spooled. This may result in a LST queue entry with a page count of zero. If you specify LST0DAT=NO, such LST queue entries are suppressed.

The operand has no effect on segmented output or on output created with the Spool-Access Support interface.

NORUN=YES|YES,DYN1=PAUSE
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 (except for reader queue entries that specify the * $$ JOB NORUN=IGN operand). You can get 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.

Notes:

1. If after a system failure VSE/POWER recovery/restart has assigned disposition X to at least one queue entry, (recorded by message 1QBCI on the console) then
   - all static partitions, which are started with autostart and which find a job eligible to run, and
   - if NORUN=YES,DYN1=PAUSE is specified, all dynamic classes with 'maximum number of partitions within class equal to 1', which are started by PLOAD DYNC during autostart, and which find a job eligible to run

   are placed into Job Control // PAUSE mode (except for eligible jobs found that specify the * $$ JOB NORUN=IGN operand). This is to avoid that a subsequent job starts to run. In addition, MSG 1Q36I is issued once to clarify this situation.
2. For shared spooling, *all* sharing systems should use the SET NORUN=YES|YES,DYN1=PAUSE autostart option. Then, whenever a shared system does recovery for its own SYSID (see message 1QB7I), jobs that were active before recovery will get the desired disposition 'X', and partitions will enter the // PAUSE mode.

   Restriction: Whenever a shared system does (full) recovery for another SYSID, jobs that were active on the other system will receive

disposition 'X', but when the other system later starts up by itself, it will *not* alert the operator by message 1QBCI, nor will its static partitions enter the // PAUSE mode.

3. For emulation of the SET NORUN function via 'UPSI 1' control, refer to "Restart and Recovery after Abnormal End" on page 188.
4. For jobs which should restart in any case, such as CICS and VTAM start jobs, you can use the * $$ JOB operand NORUN=IGN (ignore) in order to ignore the SET statement.

OPNDST=xxx

The operand applies to networking, in particular to the establishment of sessions to other SNA nodes.

For 'xxx', specify a value of 1 to 255 minutes, which VSE/POWER shall use as a time interval to re-attempt a connection via the OPNDST request to other SNA nodes.

If the OPNDST operand is not specified, a default value of 10 seconds is used. This amount provides for immediate connection completion as soon as the resources become available. On the other hand, repetitive VTAM messages may occur if the resources are not made available for a longer period.

OUTEXIT=SAS

This operand enables Spool-Access GET service requests for the local LST|PUN output queue to pass control to an enabled user-written OUTEXIT routine.

PDUMP=NOSEGMNT

This operand applies to 3800 spooling — when a job spooling SYSLST output to a 3800 printer issues a PDUMP request(s) or produces a program check or cancellation dump, VSE/POWER will ignore the implicit SETPRT macro calls of the PDUMP or DUMP routine in order to avoid segmenting the dump output into multiple LST queue entries.

PALTER=BOTHDISP

The operand applies to the PALTER command concerning the disposition in connection with networking. Use the statement to have the local and transmission dispositions changed simultaneously when moving a job/output queue entry from a local queue to the XMT queue using the PALTER ...,NODE=,DISP= command format. For a definition of local and transmission dispositions, refer to Appendix A, "VSE/POWER Disposition Codes," on page 497.

If a disposition is specified in the PALTER command (using the DISP operand), its usage varies according to the following rules:

1. If the job or output resides in one of the local queues and remains in this local queue after the PALTER command has been processed, the value of the DISP operand has been used for the local disposition.
2. If the job or output resides in the transmission queue and remains in the transmission queue after the PALTER command has been processed, the value of the DISP operand has been used for the transmission disposition.
3. If the job or output resides in the transmission queue but appears in one of the local queues after the PALTER command has been processed (due to the usage of the NODE operand), the value of the DISP operand has been used for the local disposition.
4. If the job or output resides in one of the local queues but appears in the transmission queue after the PALTER command has been processed (due to the usage of the NODE operand), the value of the DISP operand has been

used for the transmission disposition. The local disposition is preserved and as soon as the job or output is received at its final destination, the local disposition becomes effective again.

Using the SET PALTER=BOTHDISP statement, rule (4) from above is replaced by the following:

If the job or output resides in one of the local queues but appears in the transmission queue after the PALTER command has been processed, the value of the DISP operand has been used for both dispositions, for the transmission disposition as well as for the local disposition.

Note: Further information can be found in "Disposition in the Transmit Queue" in the *VSE/POWER Networking* manual.

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 VSE/POWER generation macro. One may even specify an NDT with a new local node name. For details see also *VSE/POWER Networking*, .

If the specification is in error, VSE/POWER issues a message, logs the statement, and continues startup processing; VSE/POWER uses, in this case, the NDT defined in the VSE/POWER generation macro.

RSCSROOM=DIST

This operand applies to output transmission through PNET to RSCS.

When output is transmitted to another node via PNET, a specified DIST= value other than blank will be copied to a field in the Job Header General section containing the ROOM information, thus overwriting it. RSCS uses the ROOM information as the CP distribution code if no distribution code is specified in the RSCS section of the Data Set Header Record. This offers the possibility to transfer the DIST= operand value from * $$ LST/PUN to VM/CP via PNET.

SECAC=NO|SYS

NO causes VSE/POWER Spool Access Protection not to be activated. This is the default value.

SYS makes the activation of VSE/POWER Spool Access Protection dependent on z/VSE security support. If z/VSE security was activated with the IPL command

```
SYS SEC=YES
```

Spool Access Protection will also be activated, otherwise not.

SECNODE=zonename

This operand applies to a secured VSE system for which the Access Control feature is activated. For `zonename` specify the name of the security zone. This name will override the name defined in the SECNODE operand of the VSE/POWER generation macro. For specification details on the SECNODE operand, see page 68.

Note: If the system is being warm-started, then care must be taken when specifying a different value than during the previous start-up. This will overwrite the old SECNODE value and can cause already existing

authenticated jobs to lose their VSE security authorization when they begin executing. See “Data Security Considerations” on page 11.

SEGCHECK=NO
This operand is used to force segmentation when using a NOOP (X’03’) Command Code in combination with a VSE/POWER JECL statement spooled as output data. This SET autostart statement should only be used when programs using this way of segmentation cannot be modified (as can the RJE Workstation Program) to do segmentation with the SEGMENT (IPWSEGM) macro.

SJECL=YES
The operand enables VSE/POWER to replace the alternate JECL prefix ‘..$$’ by the normal JECL prefix * $$. at the time when the job is processed. Consequently, VSE/POWER job streams can be cataloged into a VSE library from a VSE/POWER- controlled partition. See “Hiding JECL Statements at Read-in Time” on page 404 for details.

SKIP=YES
This operand causes VSE/POWER to insert a skip-to-channel 1 if the user’s output does not start with one. The operand, however, applies only to VSE/POWER controlled printers and not to external device-driving subsystems. See the SKIP operand of the PSTART command on pages 366 and 371 for details.

SYSID=system-id
The operand applies to a shared-spooling environment. It defines a new system ID.

For `system-id`, specify a value from 1 to 9.

If the specification is in error, VSE/POWER issues a message and cancels VSE/POWER startup.

WORKUNIT=PA
The operand causes VSE/POWER to process as parallel workunit whenever possible, provided the Turbo Dispatcher has been activated at IPL time. For further details, refer to “VSE/POWER Multiprocessor Support” on page 128.

Note: If the Turbo Dispatcher is not activated at IPL time, the WORKUNIT operand is simply ignored

1QZ2A=OFF
The operand causes VSE/POWER to suppress ’1QZ2A’ as prefix for messages from CICS or PSF. The messages have the prefix ’FROM device’ instead, and they are not highlighted and scroll off the console. If this autostart statement is not provided, the messages get prefix ’1QZ2A’ and remain highlighted at the console. Their console message-id is passed to CICS or PSF to remove the messages from the screen via the DOM support, when expected operator action has been taken. Otherwise the messages must be deleted manually.

1Q30D=YES
Specify this operand if at VSE/POWER abnormal termination time the operator should be prompted by message 1Q30D and asked if a dump is required or not (rather than having a formatted dump taken automatically). The operand provides:

- Compatibility for termination processing with releases 5.1.0 and earlier
- A means to lead VSE/POWER (in shared partition) into taking a dump of the VIO area in case the queue file is required for diagnosis.

- A means to take a Standalone Dump when message 1Q30D asks for a response while all spooled partitions are still alive, i.e., not yet cancelled by VSE/POWER.

1Q41I=NO
The operand applies to printing/punching output of locally executed jobs. When at execution time - also called output spooling time - the assigned printer or punch spooling device has been different from the printer or punch device used now at actual printing or punching time (PSTART LST|PUN,cuu). VSE/POWER warns you by message 1Q41I. This happens for every output entry being addressed. Use the SET 1Q41I=NO statement to suppress message 1Q41I completely, or use the PVARY MSG,1Q41I,NOCONS command to suppress the message on the console.

For further background on message 1Q41I, see "If Output Records are Ignored" on page 115.

1R33D=FLUSH|IGNORE
This operand applies to job execution. When a job is executed, VSE/POWER may detect an error in a VSE/POWER JECL statement supplied either directly in the job or by a SEGMENT macro call and prompt the operator by message 1R33D to correct the flagged statement on the console or to take other action. To prevent the 1R33D message from blocking further VSE/POWER message processing, the SET statement offers two ways to automatically respond to and suppress message 1R33D.

FLUSH
will issue warning message 1R33A and flush the job internally. The job will be retained in the RDR queue. For tracking of internally flushed jobs by Conditional Job Control, refer to the SET INTFLUSH=OPER statement ("SET: Setting VSE/POWER Startup Control Values" on page 473).

IGNORE
will issue warning message 1R33A and ignore the incorrect JECL statement. The job continues to run.

****LINE=n**
This is a conditional operand and takes effect only when a list task has been PSTARTed with the operand 'DLSEP' or runs under control of the 'SET DLSEP' autostart statement or when a Spool Access (SAS) or a Device Service (DST) task runs under control of 'SET DLSEPSAS'.

With this operand, you can influence the length of a separator page. 'n' must be specified in the range from 1 to 8 and defines how many perforation lines (starting with '****') are generated if the LST task has been started with operand 'DLSEP' or runs under control of the 'SET DLSEP'autostart statement or when a Spool Access (SAS) or a Device Service (DST) task runs under control of 'SET DLSEPSAS'. Refer to the DLSEP explanation in "Format 1: Processing Disk-Spooled Output" on page 359 or to the SET DLSEP or SET DLSEPSAS explanation in "SET: Setting VSE/POWER Startup Control Values" on page 473.

Example

For an example of how to use the SET statement, see page 103 and page 283.

DEFINE: Specifying 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 any (user-defined or native VSE/POWER) keyword operand is syntax-checked at job execution time as follows:

- First, the operand value is checked against the specification made in a corresponding DEFINE statement. If found, the keyword and its value are preserved in coded form for interpretation by other components when they retrieve a certain queue entry.
- If no corresponding DEFINE statement is found, the operand value is checked against the definitions of the 'native' VSE/POWER operands of the * $$ LST/PUN statements. If found, the operand value is preserved as a queue entry attribute for interpretation by VSE/POWER itself.

Exception: Only the PRMODE operand is processed both as 'user-defined' **and** as 'native' operand, provided that a valid DEFINE statement is present for PRMODE.

Warning: When a 'native' VSE/POWER keyword operand (such as FNO=) is introduced as a 'user-defined' operand by a DEFINE statement, its value is made available to other components only in coded form (which hopefully expect and process the coded format), but it is no longer available as a VSE/POWER attribute of a queue entry.

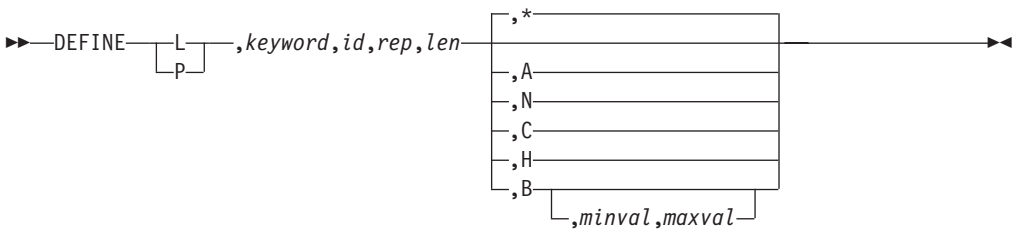
Newly defined operands might be required, for example, to support certain output devices, such as page mode printers. Currently, IBM-supplied component programs (like PSF for AFP) make use of this support. VSE/POWER passes the keyword values in coded form to the component, requesting the queue entry via the spool-access support interface. The component, in turn, interprets the coded values at its print time. Such a coded value is known as output parameter text block or OPTB. The layout is described in *VSE/POWER Application Programming*.

For every 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 every one of them.

If an output operand is valid for both the * $$ LST and the * $$ PUN statements, you must code a separate DEFINE statement for every one of them.

If a DEFINE statement is found invalid during VSE/POWER startup. The statement is displayed on the console and flagged by message 1Q09I.

Format of the Statement



L | P
Specify the type of output: L indicates that the operand is valid for an * $$ LST statement, while P indicates that it is valid for an * $$ PUN statement.

keyword
Specify the name of the keyword. It can be up to eight alphameric characters long.

id
Specify the registered keyword ID, as defined in the manual *Network Job Entry: Formats and Protocols* You can specify a hexadecimal value in the range of 1 - 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 either form:

```
keyword=value  or
keyword=(value)
```

Keywords with several values must be specified as:

```
keyword=(value1,value2,...valuen)
```

len
Specify the maximum length of the keyword value. It can range from 1 to 255 (a length of 0 is invalid). This length specification describes the maximum number of bytes to hold the keyword value in storage. Usually, this number is equal to the number of characters you specify for the keyword value.

There are, however, the following exceptions:

If the keyword value is to be converted into a binary field (see type B, below), you need at most four bytes of storage to contain the keyword value. Therefore, the length specification may range from 1 to 4 bytes only. See also the table below with the description of the maxval operand.

If, however, the keyword value is in hexadecimal representation (see type H, below), each two characters of the specified keyword value can be contained in one byte of storage. For example, if you specify keyword=01ABCF, three bytes are sufficient to contain this keyword value. Thus, if you use the hexadecimal representation, the number of characters of the specified keyword value always must be the multiple of two.

*** | A | N | C | H | B**
Specify the type of the keyword value, where:

***** The default indicates that the keyword value can contain:
- any character, except a blank or a comma, when the value is not enclosed by parenthesis
- any character except a comma, when the value is enclosed by parenthesis

A Indicates that the keyword value must be alphabetic.

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).

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.

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 the manual *Network Job Entry: Formats and Protocols* The keyword definition table shown in Table 15 shows only some of the registered keyword ids and other information needed to code a VSE/POWER DEFINE statement.

Table 15. Extract from Keyword Definition Table

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'
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
0022	1	4	THRESHLD	Integer: Range 1 - 99999999 Default: 1
3F00	1	1-128	CICSDATA	Alphameric or national Mapped by CICS

Table 15. Extract from Keyword Definition Table (continued)

ID	Max. Repet.	Length (dec.)	Keyword	Meaning
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 other spooling systems, for example, JES2/JES3.

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
```

For an example of a procedure with the DEFINE statement, see page 103 and page 283.

FORMAT: Specifying 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 completely (referred to as 'coldstart'). 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

NO
Specify 'NO' if none of the VSE/POWER files is to be formatted. This is a VSE/POWER **warm start**. It may also be used to extend either:

- the data file (see "Extending the Data File During Warm Start" on page 38), where the job queues are preserved and additional data file extent(s) formatted or
- the queue file (see "Re-Allocate Queue File During Warm Start" on page 41, where the queue records of the old(existing) queue file are transported to the extended(new) queue file, leaving the queue record linkage to the data file unchanged.

Q|D
Specify Q or D if both the VSE/POWER queue and data files are to be formatted. Q and D are synonyms; however, in all cases the full queue *and* data files are addressed. For the linkage of the queue and data files, refer to "The Size of the Queue File" on page 33.

Note: As soon as two or more data file extents must be formatted, the following combination of cold and warm start is strongly recommended to reduce the system down time:

1. first format the queue file and the first data file extent (and the account file, if required)
2. next reload your reader jobs and start processing
3. finally terminate/restart your system for a subsequent VSE/POWER warm start combined with the extension of the data file during warm start (for details see "Extending the Data File During Warm Start" on page 38). Then formatting is performed in parallel with the ongoing system spooling and job execution activities.

A Specify A if the VSE/POWER account file is to be formatted.

Example

For an example of how to use the FORMAT statement, see page 103 and page 283.

READER: Defining an Input-Spool Device

Use the statement to define the input-spool device for the static 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, see "PSTART: Starting a Task or a Partition" on page 359.

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

NO
Specify NO if no input-spool device is to be assigned to the static 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.

Examples

For an example of how to use the READER statement, see page 283.

For an example of how to display the spooled reader device of an active partition, see page 284.

PRINTERS: Defining Devices for List Output

The statement defines the printer(s) for spooling list output from the static partition that is being placed under control of VSE/POWER by a preceding PSTART command. To have spooled list output printed, start a list task by another PSTART command. For a description of the PSTART command, see "PSTART: Starting a Task or a Partition" on page 359.

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

Notes:

1 You can specify up to 14 printers for spooling list output.

NO
Specify NO if no printer is to be assigned to the static partition for spooling list output. If you specify also PUNCHES=NO, you indicate that a reader-only partition is to be started.

cuu
For cuu specify the channel and unit address 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.

The following ADD device type codes are accepted for spooled printers:
- 1403
- 1403U
- 1443
- PRT1
- 4248
- 5203
- 5203U
- 3800

A printer with AFP device type code is not accepted for spooling. For spooling of CPDS records intended for AFP printing via BAM or PIOCS refer to "Spooling of CPDS Data" on page 145.

Examples

For an example of how to use the PRINTERS statement, see page 283.

For an example of how to display the spooled printers of an active partition, see page 284.

PUNCHES: Defining Devices for Punch Output

The statement defines the punch device(s) for spooling punch output from the static partition that is being placed under control of VSE/POWER by a 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, see "PSTART: Starting a Task or a Partition" on page 359.

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

Notes:

1 You can specify up to 14 punch devices for spooling of punch output.

NO
Specify NO if no card punch is to be assigned to the static partition for spooling list output. If you specify also PRINTERS=NO, you indicate that a reader-only partition is to be started.

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.

Examples

For an example of how to use the PUNCHES statement, see page 283.

For an example of how to display the spooled punch device(s) of an active partition, see page 284.

Appendix A. VSE/POWER Disposition Codes

The disposition of a queue entry is user defined and requests VSE/POWER to handle a queue entry based on the specified disposition, namely:

- its *local disposition* (intended for processing in the RDR/LST/PUN queue) as specified by the DISP= operand (or its default) of the * $$ JOB/LST/PUN statement or the specified SPLDDP value for spool-access PUT-OUTPUT
- its *transmission disposition* (intended for processing in the XMT queue) as specified by the TDISP= operand (or its default) of the * $$ JOB/LST/PUN statement or the specified SPLOTDP value for spool-access PUT-OUTPUT.

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 explains 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	(Local only) Appendable. Spool data may be added to the job via spool-access support.
D	Process the job and delete it after processing. Default disposition.
H	Hold in queue until released.
K	Process the job and keep it in the queue after processing. (Default disposition for time event scheduling jobs that have to be processed more than once.)
L	Leave in queue until released.
X	(Local only) Hold until disposition is changed to D or K. Temporarily assigned by VSE/POWER when processing fails.
Y	(Local only) 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. Output queue entries may have also been set to a disposition of Y when ignored records were found and SET IGNREC=DISPY was specified in the VSE/POWER autostart procedure.
*	Indicates that a queue entry is being processed.

The following local 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.

If a queue entry is in execution state (DISP=*) or has a temporary *local* disposition of A, or X, or Y, VSE/POWER presents the original disposition in the ORGDP=field of a PDISPLAY...,FULL=YES request.

Figure 69 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.

Creation of a Queue Entry in Local Queues (RDR, LST, PUN): (See Note 5):

Specified Disposition	Disposition During Creation (Note 1)	Disposition After Creation — Successful Completion	Disposition After Creation — Unsuccessful Completion
===== DISP=D ==>	n/a	Queued, D	Lost or held with -A or X if output -H if job (see Note 3)
===== DISP=H ==>	n/a	Queued, H (See Note 2)	Lost or held with -A or X if output -H if job (see Note 3)
===== DISP=K ==>	n/a	Queued, K	Lost or held with -A or X if output -H if job (see Note 3)
===== DISP=L ==>	n/a	Queued, L (See Note 2)	Lost or held with -A or X if output -H if job (see Note 3)

Processing of a Queue Entry:

Disposition is			
===== DISP=D ==>	*	Deleted	Lost or held with X or Y (See Note 3)
===== DISP=H ==>	---------- n/a (See Note 2) ----------		
===== DISP=K ==>	*	Held, L	Held; may get X or Y (See Note 3)
===== DISP=K ==>	*	Held, K (See Note 4)	Held; may get X
===== DISP=L ==>	---------- n/a (See Note 2) ----------		

n/a = not applicable.

Figure 69. Overview of Local Disposition Assignments by IBM VSE/POWER

Notes:

1. To make queue entries visible in this state, use the PDISPLAY CRE command to show the contents of the Create Queue. For a general overview, see "Life Cycle of VSE/POWER Queue Entries" on page 30.
2. 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. Only the spool-access support GET-for browse request has unlimited access to dispatchable (D|H) or non-dispatchable (H|L|X|Y|A) dispositions. Browsing even provides access to entries currently being processed (*). Tracking of access is then managed by the Multiple Access Count, which is presented by the

'MACC=' field of a PDISPLAY ...,FULL=YES request. For a display example refer to page "Format 1-4: PDISPLAY LST,FULL=YES" on page 257.

3. To process a queue entry that has a disposition of A, H, X, or Y, use the PALTER command to change the entry's disposition to D or K.
4. This is only applicable for time event scheduling jobs that have to be processed more than once.
5. For creation of queue entries in the Transmit Queue (XMT) refer to the section "Disposition in the Transmit Queue" in *VSE/POWER Networking*.

Figure 70 maps the reasons for unsuccessful creation/processing against the disposition of the involved queue entry; it shows the new disposition code assigned by VSE/POWER where applicable. In these figures:

- 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		Transmit Queue
Reason for Failure	Local Read-in	SAS Sub-mission	Output Spooling	SAS Sub-mission	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 70. Mapping of Conditions to Dispositions for Unsuccessful Creation (Part 1 of 2)

Creation of a Queue Entry (Continued):

Reason for Failure	Reader Queue: Local Read-in	Reader Queue: SAS Sub-mission	Output Queue: Output Spooling	Output Queue: SAS Sub-mission	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	-

Figure 70. Mapping of Conditions to Dispositions for Unsuccessful Creation (Part 2 of 2)

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
If protected entry	-	-	Held, Y	-

SAS = Spool-access support interface

Figure 71. Mapping of Conditions to Dispositions for Unsuccessful Processing (Part 1 of 2)

Processing of a Queue Entry (Continued):

	Queue Entry Resides in			
		An Output Queue --- processed by ---		
Reason for Failure	Reader Queue (Execution)	List/Punch Writer	SAS Get Service	Transmit Queue (Send Function)
PCANCEL jobname:				
If D	Deleted	-	-	-
If K	Held, L	-	-	-
If K	Held, K (Note 1)	-	-	-
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
If K	Held, K (Note 1)	-	-	-
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
Note 1 = This is only applicable for time event scheduling jobs that have to be processed more than once.

Figure 71. Mapping of Conditions to Dispositions for Unsuccessful Processing (Part 2 of 2)

Appendix B. VSE/POWER Diagnostic and Service Aids

VSE/POWER provides dump and trace capabilities in order to facilitate trouble shooting if unforeseen problems occur during system operation. A dump is a snapshot of all VSE/POWER addressed storage areas at a desired point in time (usually immediately after a problem has occurred). A trace, on the other hand, is a series of specific storage area dumps produced whenever a certain internal function was processed. As described in the following sections, tracing can be turned on and off as desired.

For information on how to interpret dumps and traces, see "Analyzing Queue File and Data File in Dumps" on page 523.

VSE/POWER Dumps

You can request a dump of the spool files and of the partition in which VSE/POWER resides as described in the following section.

Requesting a Dump of the Spool Files

Use IPW$$DD to generate a dump of the queue record and DBLK groups belonging to a certain job. The program sends the dump to a printer or tape assigned to SYSLST.

Invoking IPW$$DD

Follow these steps to invoke IPW$$DD:

1. For dumping, make assignments for the spool files of your running VSE/POWER. In the shipped VSE/ESA system, the spool file assignments are contained in DTRPOWR.PROC of sublibrary IJSYSRS.SYSLIB, and you may just invoke this procedure. Or you may inquire about the currently existing assignments in the VSE/POWER partition (assume F1) by requesting the Job Control command 'LISTIO F1' from out of a PAUSE job of any other partition.
2. Use DLBL/EXTENT statements (same as for running VSE/POWER) for the files to be dumped. These statements need not be submitted if they are stored permanently in your system's label-information area. In the shipped z/VSE system, the label information of the spool files is contained in STDLABEL.PROC of sublibrary IJSYSRS.SYSLIB.
3. Invoke the program with // EXEC IPW$$DD.

Figure 72 on page 504 shows an example of the control statements needed to invoke IPW$$DD.

```
     // 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,DOSRES
2    // DLBL IJDFILE,'VSE.POWER.DATA.FILE'
2    // EXTENT SYS002,SYSWK1
     // EXEC IPW$$DD
     /*
3    // CLOSE SYSLST,cuu
```

Figure 72. Example of an IBM VSE/POWER File Dump Invocation

1 Assignments are as follows:

SYS000
For the account file

SYS001
For the queue file

SYS002
The first or only data-file extent

SYS003
The file's second extent, up to SYS016

SYSLST
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.

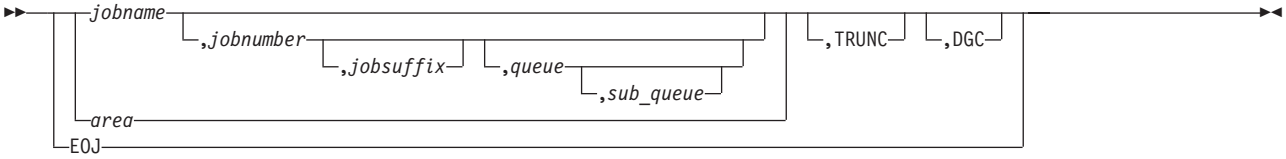
3 Needed only if SYSLST is assigned to a tape device.

Specifying Dump Options

When the program receives control, it prompts you at the console as follows:

```
DUMP FUNCTION=
```

The format of the response is as follows:



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 record and its occupied DBLK groups are to be dumped.

jobnumber
The VSE/POWER assigned job number, if this is of significance.

jobsuffix
The VSE/POWER assigned segment number (nnn), if only a certain output segment S=nnn of the named job should be dumped. For segment number examples, refer to "Format 1-4: PDISPLAY LST,FULL=YES" on page 257.

Note: Output queue entries produced by RBS segmentation (so that they have the same jobname and jobnumber but different jobsuffix S=001, S=002, etc.) can be selected by jobsuffix for dumping. If not selected explicitly, IPW$$DD will dump *all* segmented entries with the specified jobname and jobnumber, and the segment order is unpredictable.

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 and is accepted 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.

area
This option applies when a **total** dump of a certain VSE/POWER spool file is desired. Specify one of the following options:

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 (QCA, which is a logical part of the data file). For details see Table 23 on page 526.

TRUNC
Use this operand to truncate the output for this response after 5 pages, or 15 pages for area=Q.

DGC
Use this operand to dump only SEH and SER records to follow the DBLKGP chain. This operand is ignored for area dumps of types A, Q, and S.

When you intend to dump area D, you should first cease all spooling activity and then start IPW$$DD with SYSLST assigned to tape, in order to get a true snapshot of the DBLK groups of the data file.

EOJ
Use this operand to end processing of IPW$$DD.

Completion of IPW$$DD: Once the dump is completed, the program prompts you again as follows:

```
DUMP FUNCTION=
```

You can now enter a new set of specifications (see above) or end processing the dump program by entering EOJ.

Example: To get a dump of the queue record and its appended DBLK groups containing the spooled data of the list output entry named MYJOB with jobnumber 36, residing in the list queue enter

```
MYJOB,36,L
```

Format and Contents of a Generated Dump

For better understanding of the dump data, assume that the following VSE/POWER job resides in your reader queue:

```
* $$ JOB JNM=MYJOB,DISP=L,CLASS=0,PRI=5
* $$ LST DISP=H,CLASS=X
// JOB MYJOB
* ---- COMMENT LINE   1 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
* ---- COMMENT LINE   2 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
                       ....
                       ....
* ---- COMMENT LINE  24 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
* ---- COMMENT LINE  25 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
/&
* $$ EOJ
```

Figure 73. Statements of the Sample Job

When executing 'MYJOB', the list queue entry 'MYJOB' with jobnumber 36, for example, is created. When printed by a local list task it contains the following data records:

```
// JOB MYJOB                                                    DATE...
* ---- COMMENT LINE   1 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
* ---- COMMENT LINE   2 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
                       ....
                       ....
* ---- COMMENT LINE  24 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
* ---- COMMENT LINE  25 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
EOJ MYJOB                                                     DATE...
```

Figure 74. List Output of Executed Sample Job

If you expect different print output as presented by the VSE/POWER list task, or PSF, or OEM supported printers, or if you are in doubt about the printer operation codes going along with each printed line, you can look by IPW$$DD at the data which VSE/POWER has spooled for the list queue entry 'MYJOB'. The output of such a dump job will appear as follows:

```
// EXEC IPW$$DD
DUMP FUNCTION=MYJOB,36,L
Q-BLOCK      0      CH  09/24/05  l   l                 MYJOB     L  FEE  X5               20            H
Q-RECORD    30      ZN  FF6FF6FF0093009344444444444444444DEDDC44402D14CCC00EF0001000044440000FF000001000000C0000000000000
                    NM  09124105113C113C00000000000000004816200004303655007500 0C0001000000002000000B000000800000000000000
                   X'0000......08......10......18......20......28......30......38......40......48......50......58.....5F

                    CH                                             1                                   c        POWER271
                    ZN  0402444444444000000000000000000000000040700000F0000000000000000B423B51000000000001800010000DDECDFFF
                    NM  0204000000000000000000000000000000000092000010000000000000000876961A000000000009300080002766592 71
                   X'0060......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF

                    CH          POWER271                                        D
                    ZN  44444444DDECDFFF0000000000004444444400000000000000000000C000000000000000000000000000000000000000000
                    NM  0000000076659271000000000000000000000000000100000000000040000000000000000000000000000000000000000
                   X'00C0......C8......D0......D8......E0......E8......F0......F8....0100......08......10......18.....1F

                    CH
                    ZN  0000000000000000000000000000000000000000000000000000000000000000000000000000000000
                    NM  0000000000000000000000000000000000000000000000000000000000000000000000000000000000
                   X'0120......28......30......38......40......48......50......58......60......68.......

CYL    22 HEAD  0   CH      SEH:                $  P
RECORD-#       22   ZN  0C00ECC7FFFF00000001B42395B2D0000000000000000000000000000000000000000000000000000000000000000000
DBLK SIZE    1024   NM  0000258AFFFF4003000E8767ABC170000000000000000000100000000000000000000000000000000000000000000000
DBLK-# X'00001983' X'0000......08......10......18......20......28......30......38......40......48......50......58.....5F
             +++>       LL+-------------------------------------SEH-RECORD----------------------------------------------

                    CH
                    ZN  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                    NM  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                   X'0060......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF
             +++>       ----------------------------------------SEH-RECORD---------------------------------------------+

                    CH           m   M    0A  1             MYJOB                                   /0 hPOWER271        POWE
                    ZN  090F800009000D0002FC80F0000044444444DEDDC444444444440000000000000000B4226F28DDECDFFF44444444DDEC
                    NM  1C0F0000140004000401081100000000000048162000000000000000000000000000876B1088766592710000000007665
                   X'00C0......C8......D0......D8......E0......E8......F0......F8....0100......08......10......18.....1F
             +++>       LLPREFIX+-------------------------------JOB-HEADER-RECORD---------------------------------------

                    CH  R271        POWER271        POWER271
                    ZN  DFFF44444444DDECDFFF44444444DDECDFFF4444444444444444000000000000000044444444444444444444444444444
                    NM  9271000000007665927100000000766592710000000000000000000000000000000000000000000000000000000000000
                   X'0120......28......30......38......40......48......50......58......60......68......70......78.....7F
             +++>       ----------------------------------------JOB-HEADER-RECORD---------------------------------------

                    CH                                    f  L
                    ZN  4444444444444444000000000444444440B800D0044444444444444444000000000000000000004444444400004444444 4
                    NM  000000000000000000000000000000000C60030000000000000000000000000000000000000000000000000000000000 0
                   X'0180......88......90......98......A0......A8......B0......B8......C0......C8......D0......D8.....DF
             +++>       ----------------------------------------JOB-HEADER-RECORD---------------------------------------

                    CH
                    ZN  44444444444444440000444444444444444444444444444400000000444444444444444444444444444444444004000004444
                    NM  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                   X'01E0......E8......F0......F8....0200......08......10......18......20......28......30......38.....3F
             +++>       ----------------------------------------JOB-HEADER-RECORD---------------------------------------

                    CH                               8       0      POWER271                                  X     B
                    ZN  4444444444444444444444444444440F0F20000F000700DDECDFFF444444444444444444444444444444444000E00004C00
                    NM  000000000000000000000000000000080F00000000000007665927100000000000000000000000000000000000700000220
                   X'0240......48......50......58......60......68......70......78......80......88......90......98.....9F
             +++>       ---------------------------+LLPREFIX+---DATA-SET-HEADER-RECORD----------------------------------

                    CH                                                            f   5H                    BG
                    ZN  00004444444440000000000000000000000000000000000800044444444078004FC4444444444444444440000CC0000004444
                    NM  1000000000000000000000000000000000000000000000000000000000C60035800000000000000000000270000000000
                   X'02A0......A8......B0......B8......C0......C8......D0......D8......E0......E8......F0......F8.....FF
             +++>       ----------------------------------------DATA-SET-HEADER-RECORD----------------------------------

                    CH           -                                                                                    ,
                    ZN  0000000046104444444440000000000000000000004444444444440000000000F000000000000000000000000000000800004061
                    NM  00000000402000000000000000000000000000000000000000000000000001F0000000000000000000000000000090B000000B1
                   X'0300......08......10......18......20......28......30......38......40......48......50......58.....5F
             +++>       -----------------------------------------------------------------------------------+LLPREFIX+LLP
```

Figure 75. IPW$$DD Dump of List Output Entry of the Sample Job (Part 1 of 3)

```
                    CH       // JOB MYJOB                                                                DATE 09/24/2005, CLOCK
                    ZN  00000664DDC4DEDDC4444444444444444444444444444444444444444444444444444444444444CCEC4FF6FF6FFFF64CDDCD4
                    NM  900001101620481620000000000000000000000000000000000000000000000000000000000041350091241200 5B0336320
                   X'0360......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF
              +++>      REFIX(WITH END-OF-BLOCK FLAG)-----------//-JOB-STATEMENT----------------------------------------

                    CH  10/19/33
                    ZN  FF6FF6FF000000000000000000000000000000000000000000000000000000000000000000000000
                    NM  101191330000000000000000000000000000000000000000000000000000000000000000000000
                   X'03C0......C8......D0......D8......E0......E8......F0......F8.......
              +++>      -------+

CYL    22 HEAD  0   CH          * ---- COMMENT LINE   1 WITH A LENGTH OF DECIMAL 64 BYTES ---- *        MYJ00040
RECORD-#       23   ZN  050000005466664CDDDCDE4DCDC444F4ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE4666645444444444DEDFFFFF05000000
DBLK SIZE    1024   NM  08190000C0000003644553039550001069380103557380660453941306402835200000C0000000048100040081900 00
DBLK-# X'00001984' X'0000......08......10......18......20......28......30......38......40......48......50......58.....5F
              +++>      LLPREFIX+-------------------------------------------------------------------------------+LLPREFIX

                                                    ..........
                                                    ..........

                    CH          MYJ00130        * ---- COMMENT LINE  11 WITH A LENGTH OF DECIMAL 64 BYTES ---- *
                    ZN  44444444DEDFFFFF05100000 5466664CDDDCDE4DCDC44FF4ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE466664544444444
                    NM  0000000048100130081900 00C0000003644553039550011069380103557380660453941306402835200000C00000000
                   X'0360......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF
              +++>      ---------------+LLPREFIX(WITH END-OF-BLOCK FLAG)-----------------------------------------------

                    CH  MYJ00140
                    ZN  DEDFFFFF000000000000000000000000000000000000000000000000000000000000000000000000
                    NM  481001400000000000000000000000000000000000000000000000000000000000000000000000
                   X'03C0......C8......D0......D8......E0......E8......F0......F8.......
              +++>      -------+

CYL    22 HEAD  0   CH      SER:                 i  P                        -
RECORD-#       24   ZN  0C00ECD7000100000001B42398D6D000000000010001000000004610444444440000000000000000444444444444 0000
DBLK SIZE    1024   NM  0000259A00084003000E8767C9F870000001000500060001080040200000000000000000000000000000000000000000000
DBLK-# X'00001985' X'0000......08......10......18......20......28......30......38......40......48......50......58.....5F
              +++>      LL+-----------------------------------SER-RECORD-----------------------------------------------

                    CH
                    ZN  00000F000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                    NM  00001F000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                   X'0060......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF
              +++>      ----------------------------------------SER-RECORD-----------------------------------------------+

                    CH          * ---- COMMENT LINE  12 WITH A LENGTH OF DECIMAL 64 BYTES ---- *        MYJ00150
                    ZN  050000005466664CDDDCDE4DCDC44FF4ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE4666645444444444DEDFFFFF05000000
                    NM  08190000C0000003644553039550012069380103557380660453941306402835200000C0000000048100150081900 00
                   X'00C0......C8......D0......D8......E0......E8......F0......F8....0100......08......10......18.....1F
              +++>      LLPREFIX+-------------------------------------------------------------------  ... AND SO ON ...

                                                    ..........
                                                    ..........

CYL     8 HEAD  0   CH      SEH:   c            h  hP                       -
RECORD-#       25   ZN  0C00ECC7001800000001B4238508D000000000010001000000004610444444440000000000000000444444444444 0000
DBLK SIZE    1024   NM  0000258A00934003000E876985287000000100050006000208004020000000000000000000000000000000000000000000000
DBLK-# X'00000018' X'0000......08......10......18......20......28......30......38......40......48......50......58.....5F

                    CH
                    ZN  00000F000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                    NM  00001F000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                   X'0060......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF

                    CH          * ---- COMMENT LINE  21 WITH A LENGTH OF DECIMAL 64 BYTES ---- *        MYJ00240
                    ZN  050000005466664CDDDCDE4DCDC44FF4ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE4666645444444444DEDFFFFF05000000
                    NM  08190000C0000003644553039550021069380103557380660453941306402835200000C0000000048100240081900 00
                   X'00C0......C8......D0......D8......E0......E8......F0......F8....0100......08......10......18.....1F

                    CH  * ---- COMMENT LINE  22 WITH A LENGTH OF DECIMAL 64 BYTES ---- *        MYJ00250        * ---- C
                    ZN  5466664CDDDCDE4DCDC44FF4ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE4666645444444444DEDFFFFF0500000054666 64C
                    NM  C0000003644553039550022069380103557380660453941306402835200000C0000000048100250081900 0C0000003
                   X'0120......28......30......38......40......48......50......58......60......68......70......78.....7F

                    CH  OMMENT LINE  23 WITH A LENGTH OF DECIMAL 64 BYTES ---- *        MYJ00260        * ---- COMMENT L
                    ZN  DDDCDE4DCDC44FF4ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE466664544444444DEDFFFFF050000005466664CDDDCDE4D
                    NM  64455303955002306938010355738066045394130640283520000 00C0000000481002600819000 0C0000003644553 03
                   X'0180......88......90......98......A0......A8......B0......B8......C0......C8......D0......D8.....DF
```

Figure 75. IPW$$DD Dump of List Output Entry of the Sample Job (Part 2 of 3)

```
                 CH  INE  24 WITH A LENGTH OF DECIMAL 64 BYTES ---- *        MYJ00270        * ---- COMMENT LINE  25
                 ZN  CDC44FF4ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE4666664544444444DEDFFFFF050000005466664CDDDCDE4DCDC44FF4
                 NM  9550024069380103557380660453941306402835200000C000000004810027008190000C000000364455303955002 50
                X'01E0......E8......F0......F8....0200......08......10......18......20......28......30......38.....3F

                 CH  WITH A LENGTH OF DECIMAL 64 BYTES ---- *        MYJ00280        EOJ MYJOB
                 ZN  ECEC4C4DCDCEC4DC4CCCCDCD4FF4CEECE4666645444444444DEDFFFFF08000000CDD4DEDDC4444444444444444444444
                 NM  6938010355738066045394130640283520000000C000000004810028000110000561048162000000000000000000000
                X'0240......48......50......58......60......68......70......78......80......88......90......98.....9F

                 CH                                    DATE 09/24/2005, CLOCK 10/19/35, DURATION   00/00/02
                 ZN  444444444444444444444444444444444CCEC4FF6FF6FFFF64CDDCD4FF6FF6FF64CEDCECDD444FF6FF6FF030F4000
                 NM  000000000000000000000000000000000041350091241200 5B033632010119135B0449139650000010010208 4F0000
                X'02A0......A8......B0......B8......C0......C8......D0......D8......E0......E8......F0......F8.....FF
           +++>       LL+-------------------------------------------------------------------------------------+LLPREFIX

                 CH          0      q
                 ZN  030002000F00B42396BAB423BDE60000000000000000000000000000000000000000000000000000000000000000000
                 NM  00000C000000876787E087695B06000000000000000000888000000000000000000000000000000000000000000000000
                X'0300......08......10......18......20......28......30......38......40......48......50......58.....5F
           +++>       (WITH END-OF-DATA FLAG)---JOB-TRAILER-RECORD---+
                 CH
                 ZN  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                 NM  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                X'0360......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF

                 CH
                 ZN  0000000000000000000000000000000000000000000000000000000000000000
                 NM  0000000000000000000000000000000000000000000000000000000000000000
                X'03C0......C8......D0......D8......E0......E8......F0......F8.......

CYL     8 HEAD  0   CH                              CH          POWER271                                          D
RECORD-#       26   ZN  0500000044444444444444444444CC444444444DDECDFFF444444444444444444444444444444444444C0800000
DBLK SIZE    1024   NM  0919000000000000000000000000038000000000076659271000000000000000000000000000000000000040019000
DBLK-# X'00000019' X'0000......08......10......18......20......28......30......38......40......48......50......58.....5F

                                                .........
                                                .........

CYL     8 HEAD  0   CH      SER:                e   A
RECORD-#       27   ZN  0C00ECD7FFFF00000001B4238124C000000000000000000000000000000000000000000000000000000000000000000
DBLK SIZE    1024   NM  0000259AFFFF4003000E87695E611000000000000000000000000000000000000000000000000000000000000000000
DBLK-# X'0000001A' X'0000......08......10......18......20......28......30......38......40......48......50......58.....5F

                 CH
                 ZN  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                 NM  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                X'0060......68......70......78......80......88......90......98......A0......A8......B0......B8.....BF

                 CH
                 ZN  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                 NM  000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                X'00C0......C8......D0......D8......E0......E8......F0......F8....0100......08......10......18.....1F
```

Figure 75. IPW$$DD Dump of List Output Entry of the Sample Job (Part 3 of 3)

To identify the logical data records in the dump which correspond to the printed lines of Figure 74 on page 506, you must gain some knowledge about VSE/POWER spooling internals:

1. In this example, VSE/POWER has been generated with a DBLK size of 1024 bytes and a DBLKGP size of 3. The data records of 'MYJOB' are spooled in two Data Block Groups with Data Blocks '1983'-'1985' and '18'-'1A'.
2. The dump presents first the contents of the queue record (with internal decimal number 30) of the list queue entry 'MYJOB'. The queue record has a length of decimal 368 bytes as accompanied by the hexadecimal count scale of X'00'-X'16F' (for more internals, see "Analyzing Queue File and Data File in Dumps" on page 523). Each queue record byte is presented in the
   - 'CH' line in its printable format (also lowercase)
   - 'ZN' line with the zone-part of the byte
   - 'NM' line with the numeric part of the byte.

The queue record contains all attributes of the queue entry. For the meaning of selected fields you may assemble the macro IPW$DQR, which generates the queue record DSECT named QRDS.

3. Next, the dump presents the Data Blocks headed by their Cyl-Head-Rec numbers of the CKD disk chosen in this example. Data Blocks contain various record types- all starting with a two byte hexadecimal record length field that may be used to locate the subsequent record. The following records may be found:
   - SEH record, length X'00C0', first record in every first DBLK of a group. This record contains internal information (see figure Table 23 on page 526) and may be bypassed when searching user data.
   - SER record, length X'00C0', first record in every last DBLK of a group. This record contains internal information (see figure Table 22 on page 525) and may be bypassed when searching user data.
   - VSE/POWER control records that contain information also available in the queue record. The Control Records are:
     – Job Header (JHR)
     – Data Set Header (DSHR)
     – Job Trailer (JTR)

     They are in NJI standards format and carry queue entry information between networking nodes. The contents and layout is either described in *Network Job Entry Formats and Protocols* or you may assemble the macro

     ```
     IPW$DNR JHR=YES,DHR=YES,JTR=YES
     ```

     for the corresponding DSECTS.
   - Logical data records that contain user output records with their printer operation code.

Identification of Record Types: SEH/SER records can be identified by the 'SEH:'/'SER:' constant at record offset X'04'. All other records start with an eight byte prefix of the following layout:

Table 16. Layout of Data Record Prefix in DBLK ('LLPREFIX' in Sample Dump)

Displacement	Type	Contents of Field
00 - 01	Bin	Length of Data Record, including the 8 byte prefix
02	Bin	General Purpose Byte 1: X'00' normal record X'01' line print or card move data X'02' 3540 data record X'04' end-of-data, last record of queue entry X'08' unused X'10' end-of-block last record in DBLK X'20' end of 3540 data X'40' extended record
03	Bin	Command Code: X'00' for job input record X'xx' command code for list or punch output X'FF' for JHR, DHR, JTR records X'FE' for VSE/POWER end-of-page record

Table 16. Layout of Data Record Prefix in DBLK ('LLPREFIX' in Sample Dump) (continued)

Displacement	Type	Contents of Field
04	Bin	General Purpose Byte 2: X'80' job header record (JHR) X'40' job trailer record (JTR) X'20' data set header record(DSHR) X'10' CPDS data record (command code X'5A') X'08' unused X'04' fixed format message record X'02' ASA data record
05	Bin	General Purpose Byte 3: X'80' extended record - begin X'40' extended record - middle X'20' extended record - end
06 - 07	Bin	Extended Record Residual Length

With this knowledge you may now start to identify the printed lines of Figure 74 on page 506 in the dump of Figure 75 on page 507. Use as additional help the annotated (+++>) lines added in this manual below the scale line for your help.

1. Start in 1st DBLK ('1983') at X'0000' and find SEH record
2. Add SEH length '00C0' to SEH start, find JHR prefix at '00C0'
3. Add JHR length '019C' to JHR start, find DSHR prefix at '025C'
4. Add DSHR length '00F8' to DSHR start, find SKIP-CH1 at '0354'
5. Add SKIP-CH1 length '0009' to ...., find // JOB statement, etc.
6. Continue till last record(// JOB)in DBLK, noting the end-of-block flag
7. Go to 2nd DBLK and locate more data records till end-of-block
8. Go to 3rd DBLK, bypass SER record, locate data records
9. Reach last used DBLK ('18'), locate data records till
10. Finally JTR record found at '02FC' with the end-of-data flag

Note:

- VSE/POWER does not spool trailing blanks of a record, unless you specify BTRNC=NO in the RDR/LST/PUN JECL statement.
- Logical data records do not span more than DBLK unless they are bigger than a DBLK (then called 'extended' record).
- Printer command code records, in this example the next record after the Data Set Header Record, with operation code X'8B'=Skip-to-Channel1 are always equipped with one artificial blank data byte.
- In this example, the last two Data Blocks (number X'19' and X'1A') do not contain any user data, because the end-of-data-flag appeared in the job trailer of DBLK number X'18'. The unused DBLK's of the last DBLKGP of a queue entry contain:
  - hex zero when they are either non-SER DBLK's and have never been used since the VSE/POWER coldstart, or when they are the SER DBLK itself
  - unpredictable data of deleted queues, when they are non-SER DBLK's that have been used since VSE/POWER cold start or since data file extension.

Dump of the VSE/POWER Partition at Abnormal End

If a **VSE subtask** cancels, VSE/POWER automatically tries to create a formatted partition dump in the dump sublibrary of the VSE/POWER partition. Then, normal processing continues, the VSE subtask is re-attached when its support is required.

If the **main task** ends abnormally, a formatted dump is also created automatically. VSE/POWER establishes its own SYSDUMP=YES option and overwrites existing STDOPT SYSDUMP=NO settings. Depending on VSE/POWER operating in a private or shared partition, the storage copy of the queue file is part of this dump or not. For more details, refer to "Abnormal Ending of VSE/POWER" on page 187 when creation of the formatted dump fails, the operator can specify an alternate printer or tape device for the dump. VSE/POWER will then write a formatted partition dump to that device in SYSLST format. In case of tape, use the VSE/DITTO tape-to-printer function to print the tape contents. After the VSE/POWER dump is taken, all spooled partitions are cancelled by VSE/POWER, which also involves dumping. This flood of unnecessary dumps is reduced since VSE/POWER 6.1. A spooled partition dump is ordered only for a certain partition if an execution reader/writer task serving this partition has caused the abnormal termination of VSE/POWER.

PSW Mode and Access Registers

The PSW displayed in message 1Q2CI is made available by the supervisor whenever an abnormal task termination occurs and the STXIT macro has been used. A bit within the PSW indicates that the PSW is in extended control (EC) mode format. The Instruction Length Code (ILC) of the hidden basic control (BC) mode PSW is displayed together with message 1Q2CI.

The PSW used for processing by the supervisor is always in EC mode format and can be found in the STXIT save area after the general registers. At the end of the STXIT save area, the 16 access registers can be found. If VSE/POWER terminates abnormally, the information for the maintask can be found at the beginning of module IPW$$AT.

Dump of the VSE/POWER Partition at Program Request

Programming Interface Information

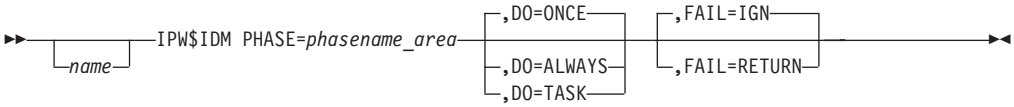
It is often desirable to generate a dump of the VSE/POWER partition without cancelling VSE/POWER. This allows a problem investigation without operation interruption. The macro IPW$IDM generates an Idump while halting the internal tasking of VSE/POWER. It takes a snapshot dump of the VSE/POWER partition including the module IPW$$NU of the system GETVIS area and files it as an Idump into the dump sublibrary defined for the VSE/POWER partition.

You can use the IPW$IDM macro when:

- A user written exit has to be debugged
- Customer problems have to be traced by Field Support.

Note: Consider that a call of IPW$IDM always returns control for a parallel (PA) work unit, even when the call came from a non-parallel (NP) work unit of a user exit.

Using IPW$IDM



To use offline debugging support, proceed as follows:
1. Call IPW$IDM
2. Note the console message referring to the previous Idump (VSE/POWER typically uses message 1QZ0I)

PHASE=*phasename-area*

This mandatory operand identifies the dump requesting phase in message 1Q2JI. For a

- User exit routine, specify the label in your program of an eight byte area that contains the exit phase name, left justified with trailing blanks.
- VSE/POWER module, specify the label of the storage descriptor area as generated by the macro IPW$DSD.
- VSE/subtask code residing in a VSE/POWER module, specify:

  PHASE=LTSB
  in case librarian subtask (LS) in module IPW$$AS

  PHASE=DTSB
  in case dump subtask (DS) in module IPW$$AS

  PHASE=STSB
  in case service subtask (AS) in module IPW$$AS

  PHASE=TISB
  in case timer subtask (TI) in module IPW$$TI.

Note: Instead of the PHASE operand, you can use the PHASEAD=phasename-area-address operand. It specifies the label of a four-byte field that contains the address of the phase-name area. The conventions are those of the PHASE operand.

DO=ONCE | ALWAYS | TASK

This optional operand may be used to reduce the number of Idumps created for the same functional situation, in case the coded call of IPW$IDM processed repeatedly.

Specify `TASK` if one dump per VSE/POWER task should be taken whenever the process flow enters this call for macro IPW$IDM.

Specify `ALWAYS` if a dump should be taken whenever the process flow enters the call for macro IPW$IDM.

Specify `ONCE`, which is the default, if a dump should be created only at the first call of IPW$IDM, after VSE/POWER has been initialized. If a further dump request is desired later during the same VSE/POWER session, use the PLOAD EXIT command for user routines or the PLOAD PHASE command for VSE/POWER modules, to regain the first time effect for dumping.

FAIL=IGN | RETURN

Use this optional operand to get feedback information, in case the Idump request has failed.

Specify IGN, which is the default, if any Idump failure should be ignored.

Specify RETURN if an Idump failure should be returned to the program that has called IPW$IDM. Upon macro return, the leftmost byte of register 11 may be flagged X'80*vvvvvv*', indicating that the Idump request has failed. It is recommended to test and reset the high order bit of register 11 (RB) as, for example, in the following code:

```
IPW$IDM PHASE=EXITNAME,FAIL=RETURN
LTR     RB,RB         IDUMP SUCCESSFUL ?
LA      RB,0(,RB)     CLEAR HIGH ORDER BYTE
BNM     LABELOK       ..YES, CONTINUE
```

Macro inclusion is ignored by an ASSEMBLER MNOTE in the following modules:

1. IPW$$ID, for all code areas (due to recursive entry)
2. IPW$$NU, for selected parts (due to register and other conventions)
3. IPW$$I1, for all code areas (due to no TCB created)
4. IPW$$I2, entry part only (until TCB chain build).

Register Conventions and Return Information: Coding an IPW$IDM request preserves **all** caller registers. Their actual contents at the point of macro call may be identified from the printed Idump together with the PSW, which points to the return address behind the macro call. The PSW condition code setting at IPW$IDM call time is not preserved.

In all cases, the code location of the IPW$IDM request and the name of the calling phase may be identified from message 1Q2JI. This message is displayed after the Idump has been taken.

Considerations when used by user exit:

1. In order to guarantee correct macro expansion, global ASSEMBLER variables starting with '&$...' must not be used.
2. Access-register mode must be set off before calling IPW$IDM.
3. Addressability for the VSE/POWER Task Control Block and the Common Address Table must have been established before macro IPW$IDM is called:

```
USING  TCDS,RB      MAKE TCB ADDRESSABLE
USING  PADS,RA      MAKE CAT ADDRESSABLE
```

Considerations when used by VSE/POWER modules:

1. Access-register mode must be set off before calling IPW$IDM. Remember that the 'Validate Data Area' routine of IPW$$NU is entered in access-register mode already.
2. Macro inclusion used in IPW$$AT does not reflect the PSW and the registers of the macro call time, but of the failing event that has called the abnormal termination routine. Message 1Q2JI is suppressed, because the failure has been documented by message 1Q2CI or 1Q2DI before.

Obtaining the Location and Status of a Requested Idump

The sequence of messages displayed on the console indicate the request location and status of an Idump.

If the macro is called by a **VSE/POWER task** (for example the execution reader task of partition BG), the console shows the following sequence of messages:

```
0S09I AN IDUMP MACRO WAS ISSUED
0S30I DUMP STARTED. MEMBER=XXXXXXXX.DUMP IN SUBLIB=YYYYYY.ZZ
1Q2JI IDUMP 'dumpname' FROM X'VVVVVV' BY PWRTASK='TCB E BG FEC'
      IN IPW$$XRE(ZZZZZZ)
```

If the macro is called by a **VSE/POWER VSE subtask** (for example the Library Service Subtask), the console shows the following sequence of messages:

```
0S09I AN IDUMP MACRO WAS ISSUED
0S30I DUMP STARTED. MEMBER=XXXXXXXX.DUMP IN SUBLIB=YYYYYY.ZZ
1Q2JI IDUMP 'dumpname' FROM X'WWWWWW' BY SUBTASK='LS'
      IN IPW$$AS(ZZZZZZ)
```

If the macro is called by **IPW$$AT** due to abnormal termination of a **VSE/POWER VSE subtask** (assume Asynchronous Service Subtask), the console shows the following sequence of messages:

```
1Q2CI PSW=070D000......, ... ,SUBTASK=AS     (existing message)
0S09I AN IDUMP MACRO WAS ISSUED
0S30I DUMP STARTED. MEMBER=XXXXXXXX.DUMP IN SUBLIB=YYYYYY.ZZ
```

If the macro is called by **IPW$$AT** due to abnormal termination of the **VSE/POWER maintask**, the console shows the following sequence of messages:

```
1Q2CI PSW=070D000......      (existing message)
1Q30D ABNORMAL VSE/POWER TERMINATION, DUMP REQUIRED? (YES/NO)
<---- YES
(1Q2ED ....
 <---- NO)
0S09I AN IDUMP MACRO WAS ISSUED
0S30I DUMP STARTED. MEMBER=XXXXXXXX.DUMP IN SUBLIB=YYYYYY.ZZ
```

Idump Error Messages

If the Idump request fails, because for example no dump library has been defined for the VSE/POWER partition, the following messages appear on the console:

```
0S09I AN IDUMP MACRO WAS ISSUED
0S33I LIBDEF STATEMENT IS MISSING FOR THE DUMP LIBRARY
1QC5I DUMP PROCESSING FAILED, RC=04
1Q2JI IDUMP '--------' FROM X'VVVVVV' BY PWRTASK='TCB E BG FEC'
      IN IPW$$XRE(ZZZZZZ)
```

If IPW$IDM is called by the VSE/POWER Maintask from the abend exit routine (that means, VSE/POWER is about to be cancelled), a failing Idump request results in

```
0S09I AN IDUMP MACRO WAS ISSUED
0S33I LIBDEF STATEMENT IS MISSING FOR THE DUMP LIBRARY
1QC5I DUMP PROCESSING FAILED, RC=04
1QC5D TO DUMP TO PRINTER OR TAPE, SPECIFY (CUU/NO)
```

End of Programming Interface Information

VSE/POWER Task Dispatching Trace

To aid in trouble shooting, a task dispatching trace operand can be used with the PSTART and PSTOP commands. Once the task trace is enabled, every VSE/POWER task dispatching event is recorded in wraparound fashion in the trace area. With the PVARY command, tracing may be enabled or disabled.

Note: To provide for a permanent task trace, z/VSE systems include in the VSE/POWER startup procedure a PSTART TASKTR,ENAB,10 command when shipped.

Requesting a Task Dispatching Trace

1. To activate a trace for VSE/POWER task dispatching events, enter:

   ```
   S TASKTR,,6
   ```

   This starts filling a trace area of 6KB with records in wraparound mode.
2. To get the trace output, have INFO ANALYSIS print possible Idump(s) taken after tracing has been started or print the dump taken at abnormal end.
3. To stop tracing temporarily, enter

   ```
   V TASKTR,DISAB
   ```
4. To resume tracing again, enter

   ```
   V TASKTR,ENAB
   ```
5. To inquire about task tracing, enter

   ```
   D TRINFO
   ```

   This displays, for example, the following

   ```
   1R461   TASKTR:  BEGIN=0035A360 END=0035BA80 NEXT=0035A980 ENAB
   ```

   The display has the following meaning:

   BEGIN
   Address of the trace area header

   END Address of the last possible entry within the task trace area

   NEXT Address of the next trace entry to be used, which is identified in storage by '<<<< LAST ENTRY' or 'TASKTRC DISABLED'.

   ENAB Tracing status (may also be DISAB or FULL,ENAB or FULL,DISAB)
6. To stop tracing and return the trace area to the storage pool, enter

   ```
   PSTOP TASKTR
   ```

   Note: This command deletes all previously collected trace information.

Format of the Trace Output

The trace area has a minimum and default size of 2KB and resides in real storage. Every trace entry is 128 (X'80') bytes long. Thus, the default trace area holds 16 trace entries.

The trace area begins at an address stored in field CATTRA, which is located at address X'520' in the VSE/POWER nucleus. The first 32 bytes of the trace area contain a header with the following information:

Table 17. Trace Header Information

Address (hex)	Content
0 - F	"DISPATCH TRACE"

Table 17. Trace Header Information (continued)

Address (hex)	Content
10 - 13	Address of first trace entry
14 - 17	Address of last trace entry
18 - 1B	Address of last used entry during tracing
1C - 1F	Total number of bytes of trace area including the header
20 - 3F	″-START-OF-TASK----TRACE-ENTRIES-″

Then follow the trace entries showing a history of the last n dispatched tasks and their status at dispatch time. Thus a malfunction may be identified which otherwise is often no longer visible at system breakdown.

Layout of an Entry in the Task Dispatching Trace

The following table shows the layout of a task dispatching trace entry.

Table 18. Layout of a Task Dispatching Trace Entry

Displacement	Type	Contents of Field
0 - 7	Char	Task ID and cuu
8 - B	Bin	TCB address
C	Char	Function trace byte
D	Char	Termination byte
E	Char	1st byte of task selection field
F		Reserved
10 - 18	Bin	Flag bytes 2 - 10
19 - 1F	Bin	Log. interface & function request bytes
20 - 57	Bin	Registers 12 - 9
58 - 5F	Bin	STCK value
60 - 6F	Bin	Access registers 1, 6, 7, 8
70 - 76	Bin	Flag bytes 11 - 17
77 - 7F		Reserved

For INFO/ANALYSIS, the trace area is described by information contained in section 6 of the system record of a VSE/POWER dump.

Layout of a Short Entry in the Task Dispatching Trace

The following table shows the layout of a short task dispatching trace entry, which is produced only when the task trace is started with the option 'FULL' or option 'QUEUE'. The short task trace entries are 16 (X'10') bytes long and are interspersed among the normal entries.

Table 19. Layout of a Short Task Dispatching Trace Entry

Displacement	Type	Contents of Field
0 - 7	Char	'TCB O.K.' or 'CLASS OK'
8 - E	Bin	Leftmost 7 bytes of 8-byte STCK value
F	Char	'W' for first trace entry after VSE/POWER leaves wait state

Commands and Aids for Exceptional Situations

This section aids the system administrator dealing with exceptional cases in which

- the standard commands do not function as desired, or
- VSE/POWER does not accept any further operator command
- a replaced phase is to be activated in a running system
- A shared link is to be verified by command/message exchange.

Regaining Control over Printer, Punch, or Tape Devices

When printer, punch, or tape devices are used by VSE/POWER tasks, while

- Hardware or software failures occur that leave the device still assigned to the VSE/POWER partition but no longer addressable by other VSE/POWER tasks, or
- VSE/POWER resource shortages disallow the task to continue or to be stopped orderly, thus still owning the device

You can, in most cases, regain control over the printer, punch, or tape device using the following commands:

```
PSTOP cuu,UNASSGN
```

For details, refer to “Format 11: Releasing a Physical Unit from VSE/POWER Control” on page 385.

```
PSTOP cuu,FORCE
```

For details, refer to “Format 12: Stopping a VSE/POWER Task Pending in Internal Wait” on page 386.

No VSE/POWER Command Accepted

In very rare cases, when a command has been entered that does not complete its desired function and no further VSE/POWER command is accepted, there are several reasons for the command processor task not to respond:

- Command task waits for real or virtual storage
- Command task waits for a VSE/POWER logical resource which is owned by another task
- Command task itself has entered a tight instruction loop
- Command task cannot continue because another task has entered a tight instruction loop
- VSE/POWER abnormal termination processing has started already.

In order to analyze such symptoms, you may use the operator communication support of z/VSE by entering the attention routine command:

```
MSG part,DATA=powercmd
```

part
the partition ID of the partition in which VSE/POWER is running

powercmd
One of the following VSE/POWER commands:

```
D A
D TASKS
D SPDEV
D SPDEVT
D AUSTMT
```

Additional operands for the above display commands may be used as described for the corresponding PDISPLAY command formats.

For a sample display, refer to "Format 16-2: MSG F1,DATA=D TASKS" on page 288.

Notes:

1. This operator communication support may still be used after VSE/POWER abnormal termination (messages 1Q2CI or 1Q2DI) and before VSE/POWER came to an end (when VSE/POWER still dumps to the dump sublibrary or waits for a response to message 1QC5D). In this case, especially the command D TASKS may be helpful for IBM to analyze the problem.
2. If you assume that VSE/POWER is 'looping', you should use repeatedly the command D TASKS and check if each time the same task has a state of 'R' (=running). If so, use additional selection operands to display the running task only and follow up the 'loop' code-area using the displayed 'PSW=addresses'.
3. In case you assume a problem and have found a task with state 'R' (=running), you should display the task control block. This display can be achieved by using the attention routine command:

   ```
   SHOW part,addr.300
   ```

   part The partition ID of the partition in which VSE/POWER is running

   addr The address of the task control block (found under header TCBADR) and 300 is a length sufficient for the task control block. Note that the period must be entered as shown above.
4. When a VSE/POWER VSE/Subtask or an OEM driven VSE/Subtask of the VSE/POWER partition has entered an instruction loop, the 'MSG part,...' command does **not** gain control to present the desired display. Instead, the command is merely accepted by Attention Routine 'READY'.

Validity Check of VSE/POWER Queue File

The display BIGGEST command scans the total VSE/POWER queue file and at the same time flags those queue entries by message 1Q2MI, whose queue identification field has been destroyed, that is, *this command may be used to verify the validity of a given queue file.*

Replacing an IPW$$xx Phase Dynamically

You may be asked by your IBM representative to catalog a VSE/POWER IPW$$xx processing phase with local fixes applied to your libraries and then to load and activate the phase for your currently running z/VSE system without terminating VSE/POWER by the PEND command. The PLOAD PHASE command can be used to load and activate a phase.

PLOAD PHASE Command

The PLOAD PHASE command loads and activates a new version of nearly all IPW$$xx(x) processing phases which represent the functional support of the VSE/POWER program product. The following phase names are excluded from PLOAD'ing:

```
IPW$$AS, IPW$$AT, IPW$$BM, IPW$$DD, IPW$$DP,
IPW$$I1, IPW$$I2, IPW$$I3, IPW$$I4, IPW$$I5,
IPW$$I7, IPW$$ID, IPW$$IN, IPW$$IP, IPW$$NS, IPW$$NU,
IPW$$LM, IPW$$SN, IPW$$TI, IPW$$TQ, IPW$$TV,
IPW$$T1, IPW$$XH, IPW$$XM
```

When loading, the command scans the library search chain of the VSE/POWER partition for the desired phase and places it into the VSE/POWER partition Getvis area. Because repetitive PLOAD requests for the same phase do not release the storage area of the last active phase, you should monitor the space consumption in the partition Getvis area.

The PLOAD PHASE request completes with one of the following messages:

1R88I If loading and enabling is successful

1Q7AI If no more partition Getvis space is available

1Q15I If phase not found in search chain

1R52I If phase out of restriction list requested

1R5CI If specified phase name is not suitable for the current environment. For further details, see the subject message in *z/VSE Messages and Codes, Volume 1*

If no reply message is given, an internal load failure has occurred. Note that long running tasks continue to process in the existing (old) version of the phase. Only when they are stopped and started again, the new phase is entered. Partitions, for example, must be PSTOP'ed and PSTART'ed again before a new PLOAD'ed IPW$$XRE execution reader phase is used for processing.

Format of the Command

PHASE
Code the operand as shown to identify loading of VSE/POWER processing phases

IPW$$xx | IPW$$xxx
Specify the name of the VSE/POWER phase with its 2 digit xx or 3 digit xxx suffix. Nearly all VSE/POWER module names match their phase names, except for the modules IPW$$AM, IPW$$DM, and IPW$$QM, which comprise the following phases:

```
IPW$$AM with IPW$$PA,IPW$$GA,IPW$$SA
IPW$$DM with IPW$$GD,IPW$$PD,IPW$$DS
IPW$$QM with IPW$$AQ,IPW$$DQ,IPW$$FQ,IPW$$NQ,
             IPW$$Q1,IPW$$RQ,IPW$$SQ
```

Make sure to always specify phase names.

Command/Message Passing Between Sharing Systems

Systems sharing common VSE/POWER queue/data file must by definition appear as one networking node, since it is the common queue file which actually represents the node. Hence, central operators on the different SYSIDs cannot use networking services to exchange commands or messages. They may display and

manipulate the entries of the commonly shared queue file, but they cannot influence any task operating on another sharing system.

However, through the Queue Control Area (see QCA in Table 20 on page 523 and Table 21 on page 524), a limited way exists to exchange commands and messages between sharing systems. With successful exchange of messages, you can verify that two systems, such as SYSID 1 and SYSID 2, have access to the *same* shared queue and data file.

Command Passing by PXMIT Command

The PXMIT command allows to specify a target SYSID to which the enclosed command is to be passed via the QCA for execution. The command format appears as follows:

PXMIT SYSID=n,power-command

SYSID=n
For n, specify the identification (1-9) of a sharing VSE/POWER system to which the command should be delivered for execution.

The command is rejected for the local SYSID and is rejected in general on non-shared running systems.

power-command
Specify any VSE/POWER command as you would enter it from the central operator console. Find allowed/disallowed commands in the 'NET' column of the "Authorization Table for the Central Op. of a Remote Node" in the *VSE/POWER Networking* manual. Any "(1)" job access limitation mentioned therein is not effective for VSE/POWER commands passed via the QCA using the 'PXMIT SYSID=...' command.

```
Example 1: Assuming the SYSID=1 central operator wants to trigger a
           PDISPLAY of the VSE/POWER queues on the central operator
           console of SYSID=3, he would enter:

                   X SYSID=3,D ALL

Example 2: Assuming the SYSID=1 central operator wants to flush
           job RUNXXX (with jobnumber 17) currently executing
           on SYSID=3, then he would use the PCANCEL command
           and enter:

                   X SYSID=3,C RUNXXX,17

           resulting in the following messages on the central operator
           console of SYSID=3:

              1R59I FOR SYSID=1 , EXECUTING COMMAND: C RUNXXX,17
              0V16I JOB RUNXXX CANCELED. REQUEST FROM VSE/POWER.
              1S78I JOB TERMINATED DUE TO OPERATOR CANCEL.
```

This version of the PXMIT command may be submitted by the central operator, RJE users, and X-partition users (CTLSPOOL and SAS-CTL) without any restriction. Independent of the type of submitter, commands will be accepted on the other SYSID and will be stated on the central op. console by message 1R59I (see Example 2).

Any console display lines resulting from execution of a PDISPLAY command do not travel back to the command originator (as with PNET), but they appear on the central operator console of the executing system (SYSID 3 in Example 1).

Any command rejected for execution (such as the PEND command) is stated by message 1RA7I (this message too does not travel back to the command originator):

```
1RA7I PEND COMMAND NOT ALLOWED ON NODE POWSHR
```

A received command on any SYSID from SYSID=1, for example, will identify the originating SYSID (thereby overwriting the originating node name, which is the same for all sharing PNET systems) by message:

```
1R59I FOR SYSID=1 , EXECUTING .......
```

When the command originator is an X-partition user with user-id 'XTOOL', for example, message 1R59I will appear as:

```
1R59I FOR SYSID=1(XTOOL), EXECUTING .......
```

Message Passing by PBRDCST Command

The PBRDCST command allows specifying a target SYSID to which the message is to be passed via the QCA. The command format appears as follows:

PBRDCST nodeid,userid,SYSID=n,'message-text'

nodeid
For nodeid, specify the destination node if your message is to be sent to another node in the network. Specify an asterisk (*) if the message is directed to a user at your own node, or if networking is not supported on your system at all.

userid For userid, specify 0 or R000 to reach the central operator, R001 - R250 to reach an RJE operator, and another userid according to the definition of a VSE/ICCF user on page 194.

SYSID=n
For n, specify the identification (1-9) of a sharing VSE/POWER system to which the message should be delivered when the final 'nodeid' has been reached. SYSID may be seen as a sub-qualification of 'nodeid'.

```
Example 1: Assuming both PNET or NON-PNET shared systems;
           you may send a message from SYSID=1
           to central operator of SYSID=3 by

               B *,0,SYSID=3,'MESSAGE FROM SYSID1'

Example 2: Assuming PNET link between node POW1 and shared
           node POWSHR with SYSID=1 and SYSID=3 connected
           as follows:
              POW1-----pnet-----POWSHR(1)-Q-POWSHR(3)

           you may send a message from POW1 to the central
           operator of SYSID=3 - which currently has
           no PNET link - by:

               B POWSHR,0,SYSID=3,'MESSAGE FROM POW1'
```

The extended version of the broadcast command may be submitted by the central operator, RJE users, and X-partition users (CTLSPOOL and SAS-CTL) without any restriction.

Restrictions

1. There is no guarantee for immediate delivery of messages/commands to another sharing system - namely when the recipient idles. When no VSE/POWER task wants to access the shared queue file in write mode until after t3=60 sec(s) (default), the recipient is forced to look into the shared queue

file for any passed message or command. Being in test mode on an idling SYSID awaiting commands/messages to be read out from the QCA, you can trigger write access to the shared queue file by entering 'PDISPLAY Q', for example. This command locks the DMB (Master Record), i.e., asks for queue file write access.

2. When many messages/commands are passed to another sharing system, it may happen that they are not received in the same sequence as they were entered at the sending system.
3. Commands/messages directed to sharing SYSIDs currently not initialized (started) are kept in the QCA and are delivered immediately at VSE/POWER startup. When such preserved commands/messages for SYSID=5, for example, should be removed from the QCA, then use the following command:

   PRESET QCA,5

Analyzing Queue File and Data File in Dumps

You may be asked to provide a VSE/POWER IPW$$DD dump, a VSE/POWER partition dump, or a standalone dump. To enable using such dumps to analyze spooling related problems, this section describes the structure of the VSE/POWER queue file and data file.

Queue Records

VSE/POWER maintains the attributes of reader/list/punch queue entries in queue file records. The queue records are numbered 0,1,2,...,n and are grouped in queue record blocks (numbered 0,1,2,...,m) in the queue file. Each queue record block has a size of 12 KB and contains 32 queue records. Each queue record has a length of 368 bytes and is housed in a compartment of 384 bytes. Queue records 0 and n ('I' and 'D' records) are used for internal purposes, and records 1,2,...,n-1 are 'usable' queue records.

Data Blocks

VSE/POWER stores spooled data into data blocks (DBLKs) in a data file. They are bundled into DBLK groups (DBLKGPs). Size values for both, DBLKs and DBLKGPs, may be specified in the POWER generation macro. DBLKs are numbered from 0-n. Adjacent numbers are put together into DBLKGPs. If, for example, the DBLKGP size is 4, then DBLKs 0-3 form a DBLKGP, 4-7 another DBLKGP, 8-11 the next group, and so on.

Free DBLKGP Subchains

During cold start, all DBLKGPs are collected into 8 free DBLKGP subchains of equal size. In each subchain, the groups are forward linked via the next-group-pointer of the Spool Environmental Record (SER), which is the heading part of the **last** DBLK of every group, called SER-DBLK (numbers 3,7,11,... in above example).

The SER record has the following layout:

Table 20. Layout of SER in a Group of a Free DBLKGP Subchain

Displacement (dec)	Type	Contents of Field
0 - 1	Bin	Length of SER record: X'00C0'
2 - 3	Bin	Reserved
4 - 7	Char	Storage Descriptor: C'SER:'

Table 20. Layout of SER in a Group of a Free DBLKGP Subchain (continued)

Displacement (dec)	Type	Contents of Field
8 - 11	Bin	Next-group-pointer: number of first DBLK of next DBLKGP X'FFFFFFFF', if last group in free subchain
12 - 13	Bin	DBLK size established at VSE/POWER cold start
14 - 15	Bin	DBLKGP size established at VSE/POWER cold start
16 - 19	Bin	Owner of DBLKGP: X'FFFFFFFF'(free) when in the free subchain.
20 – 27	Bin	StoreClock value, identified by flag
28	Char	C'F' ... StoreClock set by data file format
29 – 31	Bin	Reserved
32 – 191	Bin	Meaningless field, all hex zero.

The remaining bytes of a SER-DBLK are always X'00' when its DBLKGP resides in a free subchain.

The heading part of the **first** DBLK (numbers 0,4,8,... in above example) of every group contains the Spool Environment Header (SEH) record with the same field structure as the SER record. SEH records are only required for backward chaining of **used** DBLK groups. In free DBLK groups they have merely a placeholder function.

The SEH record has the following layout:

Table 21. Layout of SEH in a Group of a Free DBLKGP Subchain

Displacement (dec)	Type	Contents of Field
0 - 1	Bin	Length of SEH record: X'00C0'
2 - 3	Bin	Reserved
4 - 7	Char	Storage Descriptor: C'SEH:'
8 - 11	Bin	Previous-group-pointer (0,1,...,n): - X'FF...FF' (dummy), when group never used - 'undefined', when group used and returned
12 - 13	Bin	DBLK size established at VSE/POWER cold start
14 - 15	Bin	DBLKGP size established at VSE/POWER cold start
16 — 19	Bin	Owner of DBLK: — X'FFFFFFFF'(free) when group unused since cold start — 'undefined', when group used once and returned to free DBLKGP subchain.
20 – 27	Bin	StoreClock value, identified by flag
28	Char	C'F' ... StoreClock set by data file format
29 – 31	Bin	Reserved
32 - 191	Bin	Meaningless field: - all hex zero when group never used - 'undefined', when group used and returned

Non-SER-DBLKs (numbers 1,2,5,6,... in the above example) are all X'00' immediately after a cold start. If they are used once and returned to the free subchain, they may contain any data. Using the diagnosis tool IPW$$DD

specifying DUMP FUNCTION=D, all DBLKGPs of the data file are printed in collating sequence, including the free DBLKGPs described above.

Used DBLKGPs

When VSE/POWER collects spooled data for a queue entry, DBLKGPs are obtained sequentially from the free DBLKGP subchains (flagging the SER StoreClock Flag with C'A') to build the data chain of a queue entry. Again, the SER records are used to link the DBLKGPs in a forward chain. The SER owner field, however, contains the internal queue record number to which this DBLK group belongs.

The queue record itself contains a pointer (number of) to the first DBLK of the first DBLKGP and another pointer to (number of) the first DBLK of the last DBLKGP belonging to the queue entry. It also contains a count field showing the number of DBLKGPs owned. See also Figure 2 on page 30.

Important SER fields are then:

Table 22. Layout of SER in last DBLK of a 'used' DBLKGP

Displacement (dec)	Type	Contents of Field
0 - 1	Bin	Length of SER record: X'00C0'
2 - 3	Bin	Reserved
4 - 7	Char	Storage Descriptor: C'SER:'
8 - 11	Bin	Next-group-pointer (0,1,...,n): - number of first DBLK of next DBLKGP - X'FFFFFFFF', if last group of queue entry (not used for QCA DBLKGPs)
12 - 13	Bin	DBLK size established at VSE/POWER cold start
14 - 15	Bin	DBLKGP size established at VSE/POWER cold start
16 - 19	Bin	Owner of DBLKGP: — X'00000001' - X'0001869E' = group belongs to corresponding queue record —X'00000000' = group belongs to QCA.
20 – 27	Bin	StoreClock value, identified by flag
28	Char	C'P'= StoreClock set by IPW$$PD Put Data C'Q'=StoreClock set by QCA Slot Manager
29 – 31	Bin	Reserved
32 — 43	Bin	Highest Page/Line/Record Count of current DBLKGP, zero in last group.
44 — 47	Bin	Ascending number of DBLKGPs (1...n) belonging to the current queue entry, zero in last group.
48 — 119	Bin	3800 printer setup information of current DBLKGP, zero in last group
120 — 127	Bin	Reserved
128 — 191	Bin	Reserved

In addition to the SER-based forward chain, 'used' DBLK groups are also linked together by the SEH-based backward chain. Both chains facilitate fast restarting from an arbitrary page/record position to any other position within the data of a queue entry.

The SEH record has the following layout:

Table 23. Layout of SEH in first DBLK of a 'used' DBLKGP

Displacement (dec)	Type	Contents of Field
0 - 1	Bin	Length of SEH record: X'00C0'
2 - 3	Bin	Reserved
4 - 7	Char	Storage Descriptor: C'SEH:'
8 - 11	Bin	Previous-group-pointer (0,1,...,n): - number of first DBLK of previous DBLKGP - X'FF...FF', if first group of queue entry (not used for QCA DBLKGPs)
12 - 13	Bin	DBLK size established at VSE/POWER cold start
14 - 15	Bin	DBLKGP size established at VSE/POWER cold start
16 — 19		Owner of DBLKGP: — X'00000001' - X'0001869E' = group belongs to corresponding queue record — X'00000000' = group belongs to QCA
20 – 27	Bin	StoreClock value, identified by flag
28	Char	C'P'= StoreClock set by IPW$$PD Put Data C'Q'=StoreClock set by QCA Slot Manager
29 – 31	Bin	Reserved
32 - 43	Bin	Highest Page/Line/Record Count of previous DBLKGP, zero in first group
44 - 47	Bin	Ascending number of DBLKGPs (1...n) belonging to the current queue entry
48 - 119	Bin	3800 printer setup information of previous DBLKGP, zero in first group
120 — 127	Bin	Reserved
128 — 191	Bin	Reserved

Using the diagnosis tool IPW$$DD specifying

- DUMP FUNCTION=jobname,jobnumber,queue, you may print the queue record of the specified queue entry and the chain of the DBLKGPs belonging to it.
- DUMP FUNCTION=S, you may print the master record and the chain of DBLKGPs belonging to the QCA.

Note: The Queue Control Area (QCA) is used to house

- non queue entry related control information exchanged between sharing systems.
- data of extended checkpoints.

All DBLKGPs chained to the QCA (identified by owner = X'00000000') do not make use of the SEH/SER-based backward/forward chaining, instead they rely upon private chaining pointers established behind the 32-byte header of the SER record.

Returning DBLKGPs to the Free Subchain

When a queue entry is deleted, its queue record is returned to the free queue record chain. Its DBLKGPs are returned one-by-one to the free DBLKGP subchains

flagging the SER Store Clock Flag with C'D', where they appear again (as described under "Free DBLKGP Subchains" on page 523).

DBLKGP Tracing

During allocation and deallocation of DBLKGPs, internal trace functions permanently verify the consistency of the free and used DBLKGP chains. Detected failures are reported by the console messages 1QFAA, 1QFBA, 1QFCA, or 1QFDA followed by 1QF8I, for example, and self repair takes place by dropping inconsistent unused DBLKGPs. At the same time, the statistics status report line

```
DBLK-GPS LOST DUE TO I/O OR LOGIC ERROR       nnn GROUPS
```

indicates the number of groups lost due to logic error.

The same statistic counter also is increased when a Read/Write I/O request for any DBLK fails and is reported by the messages 1Q6GA, 1Q6HA, 1Q6KA, or 1Q61I, for example, followed by the message 1QF8I and possibly by termination of the subject VSE/POWER task.

When VSE/POWER abends during deallocation of DBLKGPs of a queue entry, they may be lost. The subsequent recovery increases also the statistics counter displayed in the 'LOST' message shown above.

The internal traces also verify the overall structure of the logical records within any DBLK when being read in from the data file. If any logical record length of 0 is detected, the subject VSE/POWER task is terminated and is reported by the message 1Q6LA/1Q61I. The queue entry is preserved by the message 1Q6JI.

Data Ownership Tracing

Whenever entries of the RDR/LST/PUN/XMT queues are processed by corresponding tasks and their data are read from the data file, VSE/POWER verifies whether the first or subsequent DBLKGPs chained off a queue entry really belong to this entry. This trace method is an extremely fast and precise way to detect data corruption. In case an error is detected, the queue entry is flagged by message 1Q6UA/1Q6VA and saved as indicated by message 1Q6JI, and the corresponding processing task is terminated, requiring the operator to restart.

Layout of VSE/POWER Tapes

VSE/POWER has two tape types:

- Spool tapes. These are produced by POFFLOAD BACKUP|PICKUP|SAVE or `* $$ LST/PUN DISP=T` JECL statement.
- SYSIN tapes. See the note below.

Either tape type may be:

- IBM standard unlabeled (single volume)
- IBM standard labeled (single or multivolume – see note below).

The tape layouts are diagrammed in the following figures.

Notes:

1. SYSIN labeled tapes may also be multi-file as well as multi-volume for native SYSIN tape processing (not illustrated here).
2. VSE/POWER labeled tapes use standard labels at the tape beginning and end. However, the data between the tape labels are non-standard, using tape marks (TM) to separate spool entries.

Queue Entries on Spool Tape

When written to tape, queue entries of any queue (RDR,LST,PUN, or XMT queue) have the following format:

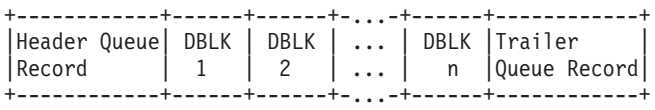


Figure 76. Layout of a Queue Entry on Tape

The queue records contain all attributes of the corresponding queue entry, the data blocks (DBLK 1,2,...) contain VSE/POWER control records and user data records in VSE/POWER logical record format. Queue entries are preceded and separated by a single tape mark (TM), and the logical end-of-tape is indicated by a double tape mark, as presented by "Layout of Spool Tapes" on page 530.

On IBM 9346 or 3592 WORM tapes 2 forms of trailing queue entries may be found without a 'Trailer Queue Record' but terminated by 2 tape marks instead. Such 'invalid' queue entries are bypassed by the VSE/POWER tape reading functions. The following figures show the 'invalid' queue entry tape layout.

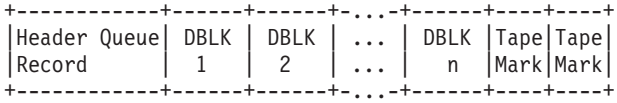


Figure 77. Layout of an 'Invalid' Queue Entry on 9346 or 3592 Tape with DBLK data

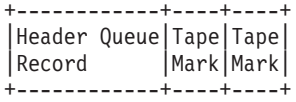


Figure 78. Layout of another 'Invalid' Queue Entry on 9346 or 3592 Tape without DBLK data

The queue record length depends on the release level of VSE/POWER:

VSE/POWER Release	Number of Bytes
4.1	240
5.1	240
5.2.x	256
6.1.x	256
6.3	256
6.4	368
6.5	368
6.6	368
6.7	368
7.1	368

The data block length is defined by the DBLK operand of the POWER generation macro. This length may vary between

- 1,000 - 12,288 bytes for releases up to VSE/POWER 5.2
- 1,000 - 65,024 bytes for releases VSE/POWER 6.1 and following.

The Data Block Group (DBLKGP) size is transparent on tape; this means DBLKs are not headed by SEH/SER control records. Due to the standard format, these tapes can be read in again and can be processed by the

- POFFLOAD LOAD/SELECT function for RDR/LST/PUN/XMT entries
- Spool output tape Printing/Punching function for LST/PUN-queue entries.

The size of DBLKs on tape can be different from the DBLK size of the reading system, because VSE/POWER processes the DBLKs according to logical records contained therein.

Layout of Spool Tapes

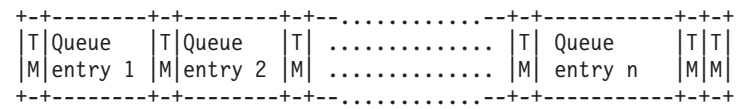


Figure 79. Layout of an Unlabeled Spool Tape

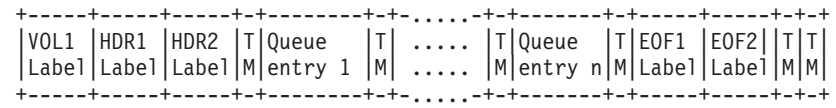
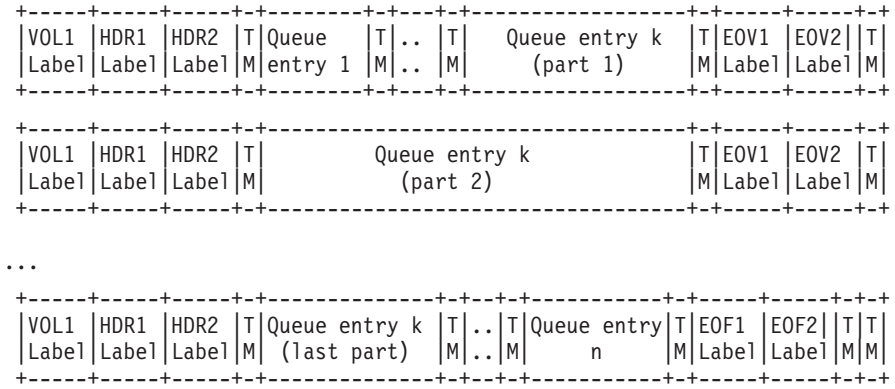


Figure 80. Layout of a Labeled Single Volume Spool Tape



where a continued queue entry has the following layout:

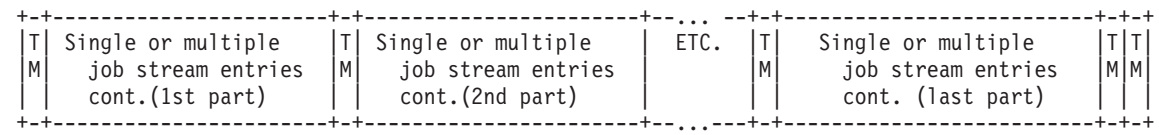
Part 1:

Part 2:

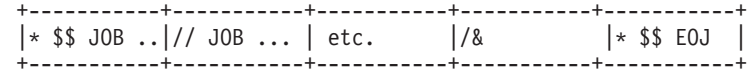
Last part:

Figure 81. Layout of a Labeled 3-Volume Spool Tape

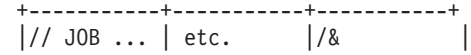
Layout of SYSIN Tapes



where a job stream entry has the following layout:



or:



Note - If the final job on the tape is not complete then it will be closed up with a /& or * $$ EOJ statement, set to DISP=HOLD, and the operator will be notified by the message 1Q87I.

Figure 82. Layout of an Unlabeled SYSIN Tape

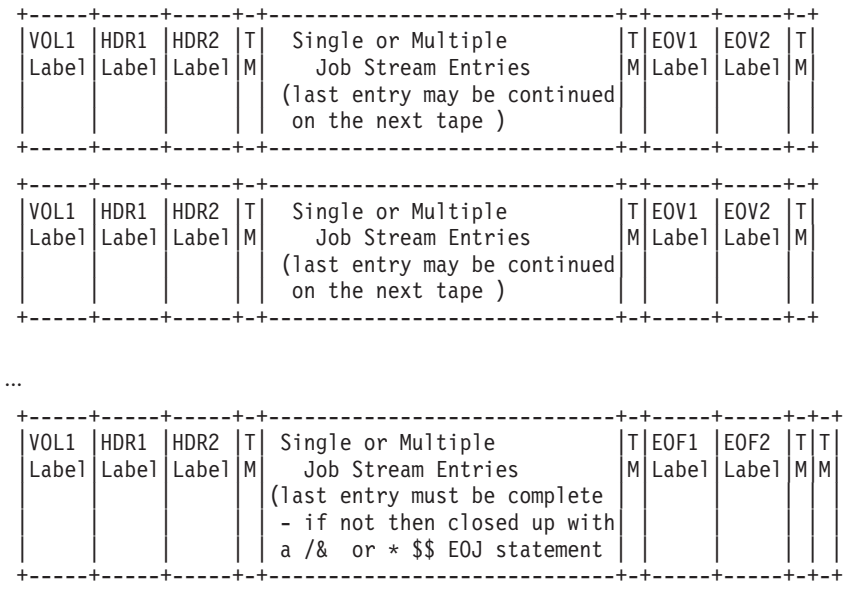


Figure 83. Layout of Labeled Multi-Volume SYSIN Tapes

Appendix C. RJE, SNA Data Compaction

Compaction as implemented for the VSE/POWER RJE/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 and count them.
2. Use Figure 84, 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. Master characters are those that occur most often in the data stream. You decide how many of those are to be considered, according to this list.

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 84. 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 85 on page 534.

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 85. Example of a Compact-Code Character Set

In the first column of Figure 84 on page 533, 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 84 on page 533. 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 84 on page 533). 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
```

3. 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, see "PCPTAB Generation Macro" on page 90.

This is all you have to know 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 every 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. Every 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 ID 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. 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 terminal.

The table TRTC contains zeros at the displacements corresponding to every 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 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'.

Every 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 86 on page 538. For the TRTC table, a dash represents any nonzero value.

For the XLTC table, a dash represents any value, because such a character does 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 86. IBM 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 terminal. There the string is re-translated into the original hexadecimal code, and then passed to the printer.

Glossary

This glossary includes terms and definitions related primarily to IBM VSE/POWER. If you do not find the term you are looking for, refer to the index of this book or view the *IBM Glossary of Computing Terms* at

`http://www.ibm.com/ibm/terminology`

The glossary includes definitions with:

- Symbol *, which denote entries from the *IBM Glossary of Computing Terms*.
- Symbol (A) from the *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036.
- Symbols (I) or (T) from the *ISO Vocabulary - Information Processing* and the *ISO Vocabulary - Office Machines,* developed by the International Organization for Standardization, Technical Committee 97, Subcommittee 1. Definitions of published sections of the vocabularies are identified by symbol (I) after definition; definitions from draft international standards, draft proposals, and working papers in development by the ISO/TC97/SC1 vocabulary subcommittee are identified by symbol (T) after definition, indicating final agreement has not yet been reached among participating members.

The following cross-references are used:

- Contrast with. This refers to a term that has an opposed or substantively different meaning.
- Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the dictionary.
- Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.
- See. This refers the reader to multiple-word terms that have the same last word.
- See also. This refers the reader to related terms that have a related, but not synonymous, meaning.

access control. A function of VSE that ensures that the system and the data and programs stored in it can be accessed only by authorized users in authorized ways.

access register (AR). A hardware register that a program uses to identify an address space or a data space. Each processor has 16 ARs, numbered 0 through 15, which are paired one-to-one with the 16 general-purpose registers (GPRs).

account file. A disk file maintained by VSE/POWER containing accounting information generated by VSE/POWER and the programs running under VSE/POWER.

address. 1.The location in the storage of a computer where data are stored. 2.In data communication, the unique code assigned to every device or workstation connected to a network.

addressing mode (AMODE). A program attribute that refers to the address length that a program is prepared to handle on entry. Addresses may be either 24 bits or 31 bits in length. In 24-bit addressing mode, the processor treats all virtual addresses as 24-bit values; in 31-bit addressing mode, the processor treats all virtual addresses as 31-bit values. Programs with an addressing mode of ANY can receive control in either 24-bit or 31-bit addressing mode.

address space. A range of up to two gigabytes of contiguous virtual storage addresses that the system creates for a user. Unlike a data space, an address space contains user data **and** programs, as well as system data and programs, some of which are common to all address spaces. Instructions execute in an address space (not a data space). Contrast with data space.

address space control (ASC) mode. The mode (determined by the PSW) that tells the system where to find referenced data. It determines how the processor resolves address references for the executing programs. z/VSE supports two types of ASC modes:

1. In **primary** ASC mode, the data that a program can access resides in the program's own (primary) address space. In this mode, the system uses the contents of general-purpose registers to resolve an address in the address space; it does not use the contents of the access registers (ARs).
2. In **access register (AR)** ASC mode, the data that a program can access may reside in an address space other than the primary or in a data space. In this mode, the system uses both a general-purpose register (GPR) and the corresponding access register together to resolve an address in another address space or in a data space. Specifically, the AR contains a value, called an ALET, that identifies the address space or data space that contains the data, and the GPR contains a base address that points to the data within the address space or data space.

*** adjacent nodes.** Two nodes connected without any intermediate nodes. (T)

*** alternate tape.** A tape drive to which the operating system switches automatically for tape read or write operations if the end of the volume has been reached on the originally used tape drive.

*** alternate route.** A secondary or backup route that is used if normal routing is not possible.

AMODE. Addressing mode.

application program. A program written for or by a user that applies directly to the user's work.

AR. Access register.

AR (access register) mode. If a program runs in AR mode, the system uses the access register/general-purpose register pair to resolve an address in an address space or data space. Contrast with *primary mode*. See also *address space control (ASC) mode*.

ASC mode. Address space control mode.

*** ASCII.** American National Standard Code for Information Interchange.

ASI (automated system initialization) procedure. A set of control statements which specifies values for an automatic system initialization.

assemble. To translate a program from assembler language into object code.

assembler. A computer program used to assemble. Synonymous with *assembly program*.

assembler language. A programming language whose instructions are usually in one-to-one correspondence with machine instructions and allows to write macros.

attention routine. A routine of the system that receives control when the operator presses the Attention key. The routine sets up the console for the input of a command, reads the command, and initiates the system service requested by the command.

*** automated system initialization (ASI).** A function that allows control information for system startup to be cataloged for automatic retrieval during system startup.

*** autostart.** A facility that starts up VSE/POWER with little or no operator involvement.

*** background partition.** An area of virtual storage in which programs are executed under control of the system. By default, the partition has a processing priority lower than any of the existing foreground partitions.

*** backup copy.** A copy, usually of a file or a library member, that is kept in case the original file or library member is unintentionally changed or destroyed.

binary synchronous communication (BSC). Method of telecommunication using binary synchronous line discipline. Contrast with *SDLC*.

Basic Security Manager (BSM). The component of VSE that provides basic security support.

binary synchronous communication (BSC). Method of telecommunication using binary synchronous line discipline. Contrast with *SDLC*.

block. Usually, a block consists of several records of a file that are transmitted as a unit. But if records are very large, a block can also be part of a record only. See also *control block*.

BSC. Binary synchronous communication.

BSD. Used with tracing of the BSD/C (Berkeley C language) socket interface of TCP/IP for VSE/ESA.

buffer. An area of storage temporarily reserved for input or output operations; an area into which data is read or from which data is written. Synonymous with I/O area.

byte. Eight adjacent binary digits that are operated upon as a unit and that constitute the smallest addressable unit of information within a computer system. Normally, it represents a stored character.

card punch. A device that punches holes into a card to represent data.

card reader. A device that reads or senses holes in a punched card, transforming the data from hole patterns to electrical signals. (I) (A)

catalog. 1. A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (I) (A) 2. To store a library member such as a phase, module, or book in a sublibrary.

*** cataloged procedure.** A set of control statements placed in a library and retrievable by name.

channel-to-channel attachment (CTCA). A function that allows data to be exchanged (1) under the control of VSE/POWER between two virtual VSE machines running under VM or (2) under the control of VTAM between two processors.

CICS. Customer Information Control System.

class. In VSE/POWER, a means of grouping jobs that require the same set of resources.

cold start. Initialization of all queues. All information from the queues before the cold start is lost.

*** compile.** To translate a source program into an executable program (object program). See also *assembler*.

compiler. A program used to compile.

conditional job control. The capability of the job control program to process or to skip one or more statements based on a condition that is tested by the program.

control block. An area within a program or a routine defined for the purpose of storing and maintaining control information.

control program. A program to schedule and supervise the running of programs in a system.

CTCA. Channel-to-channel attachment.

*** data block (DBLK).** In VSE/POWER, the unit of transfer for spooling job input and job output.

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.

data file. A VSE/POWER file to hold input and output program data needed or generated by programs under control of VSE/POWER.

data set header record. In VSE/POWER abbreviated as DSHR, alias NDH or DSH. An NJE control record either preceding output data or, in the middle of input data, indicating a change in the data format.

*** DBLK.** Data block.

default value. A value assumed by the program when no value has been specified by the user.

data space. A range of up to two gigabytes of contiguous virtual storage addresses that a program can directly manipulate through ESA/370 instructions. Unlike an address space, a data space can hold only user data; it does not contain shared areas, system data or programs. Instructions do not execute in a data space, although a program can reside in a data space as non-executable code. Contrast with address space.

*** device address.** 1. The identification of an input/output device by its channel and unit number. 2. In data communication, the identification of any device to which data can be sent or from which data can be received.

device service task. In VSE/POWER, a task that processes requests for transfer of output to be printed or punched by a device owned by a program in another partition.

*** device type code.** The four- or five-digit code to be used for defining an I/O device to a computer system.

direct link. Connection between two adjacent nodes in a network.

disk operating system residence volume (DOSRES). The disk volume on which the system sublibrary IJSYSRS.SYSLIB is located including the programs and procedures required for system startup.

disk sharing. An option that lets independent computer systems use common data on shared disk devices.

disposition. A means of indicating to VSE/POWER how a job input or output entry is to be handled: according to its local disposition in the RDR/LST/PUN queue or its transmission disposition when residing in the XMT queue. A job may, for example, be deleted or kept after processing.

DOSRES. Disk operating system residence volume.

DSH(R). Data set header record.

dump. 1. Data that has been dumped. (I) (A) 2. To write at a particular moment some contents of storage to another data medium for the purpose of safeguarding or debugging the data. (T)

dynamic partition. A partition created and activated on an 'as needed' basis that does not use fixed static allocations. After processing, the occupied space is released. In other words, a partition dynamically defined by VSE/POWER when a VSE/POWER job is begun in the partition and then de-allocated when this job completes. Contrast with *static partition*.

*** dynamic partition balancing.** A VSE facility that allows the user to specify that two or more or all partitions of the system should receive about the same amount of time on the processor.

end node. Destination node for a job or output.

Enterprise Systems Architecture (ESA). See *ESA/370* and *ESA/390*.

ESA mode. Such a supervisor will run on Enterprise Systems Architecture processors (ESA/370 and ESA/390) and provides support for multiple virtual address spaces, the channel subsystem, and more than 16MB of real storage.

ESA/370. IBM Enterprise Systems Architecture/370. The extension to the IBM System/370 architecture which includes the advanced addressability feature that provides access registers.

ESA/390. IBM Enterprise Systems Architecture/390. The latest extension to the IBM System/370 architecture which includes the advanced addressability feature and advanced channel architecture.

exit. A routine, normally user-supplied, that receives control from the system when a certain event occurs (abnormal-end exit, for example).

extended addressability. 1. See *31-bit addressing*. 2. The ability of a program to use virtual storage that is outside the address space in which the program is running. Generally, instructions and data reside in a single address space - the primary address space. However, a program can have data in address spaces other than the primary or in data spaces. (The instructions remain in the primary address space, whilst the data can reside in another address space or in a data space.) To access data in other address spaces, a program must use access registers (ARs) and execute in access register mode (AR mode).

External Security Manager (ESM). A priced vendor product that can provide extended functionality and flexibility compared to that of the Basic Security Manager (BSM), which is part of VSE.

EZASMI macro. An assembler application programming interface for TCP/IP services that addresses the BSD/C interface of TCP/IP for VSE/ESA through the LE (Language Environment) C support.

*** FCB.** Forms control buffer.

file. A named set of records stored or processed as a unit. (T) Synonymous with *data set*.

flush. To stop processing of a current queue entry immediately and to start on the next one.

*** foreground partition.** A space of virtual storage in which programs are executed under control of the system. By default, a foreground partition has a higher processing priority than the background partition.

*** forms control buffer (FCB).** In the 3800 Printing Subsystem, a buffer for controlling the vertical format of printed output.

general-purpose register. A register, usually explicitly addressable, within a set of registers that can be used for different purposes.

generate. To produce a computer program by selecting subsets of standardized code under the control of parameters. (A)

generation. See *macro generation*.

*** GETVIS space.** Storage space within a partition or the shared virtual area, available for dynamic allocation to programs.

GPR. General-purpose register.

hard-copy file. A system file on disk, used to log all lines of communication between the system and the operator at the system console, to be printed on request.

interactive partition. An area of virtual storage for the purpose of processing a job that was submitted interactively via VSE/ICCF.

I/O (input/output). See *input* and *output*.

IPL. Initial program load.

JCL. Job control language.

JECL. Job entry control language.

JHR. Job header record.

job accounting. A system function that lists how much every job step uses of the different system resources.

job accounting interface. A function that accumulates accounting information for each job step, to be used for charging the users of the system, for planning new applications, and for supervising system operation more efficiently.

*** job accounting table.** An area in the supervisor where accounting information is accumulated for the user.

job catalog. A catalog made available for a job by means of the file name IJSYSUC in the respective DLBL statement.

job control language (JCL). A language that serves to prepare a job or each job step of a job to be run. Some of its functions are: to identify the job, to determine the I/O devices to be used, set switches for program use, log (or print) its own statements, and fetch the first phase of each job step.

job control statement. A particular statement of JCL.

job entry control language (JECL). A control language that allows the programmer to specify how VSE/POWER should handle a job.

job header record (JHR). As VSE/POWER parameter JHR, alias network job header or NJH. An NJE control record preceding a job entry in the input or output queue.

job step. One of a group of related programs complete with the JCL statements necessary for a particular run. Every job step is identified in the job stream by an EXEC statement under one JOB statement for the whole job.

job stream. The sequence of jobs as submitted to an operating system.

job trailer record (JTR). As VSE/POWER parameter JTR, alias NJT. An NJE control record terminating a job entry in the input or output queue and providing accounting information.

JTR. Job trailer record.

KB. Kilobyte. 1024 bytes of storage.

*** librarian.** The set of programs that maintains, services, and organizes the system and private libraries.

library. See *VSE library*.

*** library block.** A block of data stored in a sublibrary.

*** library directory.** The index that enables the system to locate a certain sublibrary of the accessed library.

*** library member.** The smallest unit of data to be stored in and retrieved from a sublibrary.

link-edit. To create a loadable computer program by having the linkage editor process compiled (assembled) source programs.

list task. A VSE/POWER task to control writing of spooled output to a printer.

*** load.** To bring a program phase from a library into virtual storage to run it.

*** logging.** The recording of data about specific events.

logical unit (LU). 1. A name used in programming to represent an I/O device address. 2. In SNA, a port through which a user accesses the SNA network, a) to communicate with another user and b) to access the functions of the SSCP. An LU can support at least two sessions -- one with an SSCP and one with another LU -- and may be capable of supporting many sessions with other LUs.

LU. Logical unit.

macro (instruction). 1. In assembler programming, a user-invented assembler statement that causes the assembler to process a set of statements defined previously in the macro definition. 2. A sequence of VSE/ICCF commands defined to cause a sequence of certain actions to be performed in response to one request.

macro definition. A set of statements and instructions that defines the name of, format of, and conditions for generating a sequence of assembler statements and machine instructions from a single source statement.

macro expansion. See *macro generation*.

macro generation. An assembler operation by which a macro instruction gets replaced in the program by the statements of its definition. It takes place before assembly. Synonymous with *macro expansion*.

*** main task.** The main program within a partition in a multiprogramming environment.

master console. In z/VSE, one or more consoles that receive all system messages, except for those that are directed to one particular console. Contrast with the *user console* which receives only those messages that are specifically directed to it, for example messages issued from a job that was submitted with the request to echo its messages to that console. The operator of a master console can reply to all outstanding messages and enter all system commands.

*** MB.** Megabyte.

Mb. Megabit.

*** megabyte (MB).** 1,024KB or 1,048,576 bytes of storage.

*** member.** The smallest unit of data that can be stored in and retrieved from a sublibrary.

message. 1. In VSE, a communication sent from a program to the operator or user. It can appear on a console, a display terminal or on a printout. 2. In telecommunication, a logical set of data being transmitted from one node to another.

*** microcode.** 1. A code written using the instructions of a specific instruction set and implemented in a part of storage that is not program-addressable. 2. To design write, and test one or more micro instructions.

*** migrate.** To move to a changed operating environment, usually to a new release or version of a system.

mode. A method of operation.

NCP. Network Control Program.

NDT. Network definition table.

NetView. An IBM licensed program to monitor a network, manage it, and diagnose its problems.

network. 1. An arrangement of nodes (data stations) and connecting branches. 2. The assembly of equipment through which connections are made between data stations.

network address. In SNA, an address, consisting of subarea and element fields, that identifies a link, link station, or NAU. Subarea nodes use network addresses; peripheral nodes use local addresses. The boundary function in the subarea node to which a peripheral node is attached transforms local addresses to network addresses and vice versa.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

network definition table (NDT). In VSE/POWER networking, the table where every node in the network is listed.

networking. Making use of the services of a network program.

network job entry (NJE). 1. A facility for transmitting jobs (JCL and in-stream data sets), SYSOUT data sets, (job-oriented) operator commands and operator messages, and job accounting information from one computing system to another. 2. A facility that provides access to batch computing facilities from other host systems. It enables users to transfer work and data throughout a distributed network of batch computing facilities. (″NJE″ is not a part of ″Systems Network Architecture (SNA)″, but is an application layer which uses SNA, BSC and CTC transmission facilities.) 3. The JES2 program product implementation of the NJE Protocol.

* **node.** 1. In SNA, an end point of a link or junction common to several links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. 2. In VTAM, a point in a network defined by a symbolic name. Synonymous with *network node*.

* **operating system.** Software that controls the running of programs; an operating system may provide services such as resource allocation, scheduling, input/output control, and data management. (I) (A)

* **operator command.** A statement to a control program, issued via a console or terminal. It causes the control program to provide requested information, alter normal operations, initiate new operations, or end existing operations.

OPTB. 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. 1. In a virtual storage system, the unit of code or data or both which is transferred between processor storage and the PDS as needed for processing. 2. To transfer pages between processor storage and the page data set.

page data set (PDS). One or more extents of disk storage in which pages are stored when they are not needed in processor storage.

page fault. A program interruption that occurs when a program page marked ″not in processor storage″ is referred to by an active page.

* **page fixing.** Marking a page so that it is held in processor storage until explicitly released. Until then, it cannot be paged out.

page frame. An area of processor storage that can contain a page.

page-in. The process of transferring a page from the PDS to processor storage.

page I/O. Page-in and page-out operations.

page-out. The process of transferring a page from processor storage to the PDS.

* **page pool.** The set of page frames available for paging virtual-mode programs.

partition. A division of the virtual address area available for running programs. See also *dynamic partition, static partition*.

* **partition balancing, dynamic.** A VSE facility that allows the user to specify that two or more or all partitions of the system should receive about the same amount of time on the processor.

PDS. Page data set.

* **physical record.** The amount of data transferred to or from auxiliary storage. Synonymous with *block*.

PNET. Programming support available with VSE/POWER; it provides for the transmission of selected jobs, operator commands, messages, and program output between the nodes of a network.

POWER. See *VSE/POWER*.

* **pregenerated operating system.** An operating system such as z/VSE that is shipped by IBM mainly in object code. IBM defines such key characteristics as the size of the main control program, the organization and size of libraries, and required system areas on disk. The customer does not have to generate an operating system.

primary mode. If a program runs in primary mode, the system resolves all addresses within the current (primary) address space. Contrast with *AR (access register) mode*. See also *address space control (ASC) mode*.

private area. The part of an address space that is available for the allocation of private partitions. Its maximum size can be defined during IPL. Contrast with *shared area*.

* **private partition.** Any of the system's partitions that are not defined as shared. See also *shared partition*.

procedure. See *cataloged procedure*.

* **processing.** The performance of logical operations and calculations on data, including the temporary retention of data in processor storage while this data is being operated upon.

processor. The hardware component that interprets and executes instructions. (I) (A)

processor storage. The storage contained in one or more processors and available for running machine instructions. Synonymous with *real storage*.

profile. A description of the characteristics of a user or a computer resource.

prompt. To issue messages to a terminal or console user, requesting information necessary to continue processing.

punch. 1. To make holes in some data medium according to a signal code and thus save data on that medium. 2. A machine (output device) to punch 80-column punch cards.

* **punch card.** A card into which hole patterns can be punched; normally, it is characterized by 80 columns and 12 rows of punch positions.

* **queue.** 1. A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a network. 2. To arrange in, or form, a queue.

QCA. Queue Control Area

Queue Control Area (QCA). In VSE/POWER, an area of the data file which may contain:

- Extended checkpoint information
- Control information for a shared environment.

queue file. A disk file maintained by VSE/POWER that holds control information for the spooling of job input and job output.

queue record. A record in the queue file containing descriptive information about a job or job output.

real address. The address of a location in processor storage.

* **real address area.** In VSE, the area of virtual storage where virtual addresses are equal to real addresses.

* **real address space.** In a VSE system operating in 370 mode, the address space whose addresses map one to one to the addresses in processor storage.

real mode. In VSE, a processing mode in which a program may not be paged. Contrast with *virtual mode*.

real storage. See *processor storage*.

* **remote job entry (RJE).** Submission of jobs through an input unit that has access to a computer through a data link.

* **restore.** To write back on disk data that was previously written from disk to an intermediate storage medium such as tape.

RJE. Remote job entry.

* **routine.** Part of a program, or a sequence of instructions called by a program, that may have some general or frequent use. (I) (A)

* **routing.** The assignment of the path by which a message will reach its destination.

RSCS. remote spooling communications subsystem.

* **run.** 1. A performance of one or more jobs. (I) (A) 2. A performance of one or more programs. (I) (A)

3. To cause a program or job to be performed.

SAS. See *spool access support*.

schedule. To select a program or task for getting control over the processor.

SDLC. Synchronous data link control.

segmentation. In VSE/POWER, a facility that breaks list or punch output of a program into segments so that printing or punching can start before this program has finished generating such output.

shared area. An area of storage that is common to all address spaces in the system. z/VSE has two shared areas:

1. The shared area (24 bit) is allocated at the start of the address space and contains the supervisor, the SVA (for system programs and the system GETVIS area), and the shared partitions.
2. The shared area (31 bit) is allocated at the end of the address space and contains the SVA (31 bit) for system programs and the system GETVIS area.

* **shared partition.** In VSE, a partition allocated for a program (VSE/POWER, for example) that provides services for and communicates with programs in other partitions of the system's virtual address spaces.

* **shared spooling.** A function that permits the VSE/POWER account file, data file, and queue file to be shared among several computer systems with VSE/POWER.

* **shared virtual area (SVA).** In VSE, a high address area that contains a list system directory list (SDL) of frequently used phases, resident programs shared between partitions, and an area for system support.

SNA. Systems Network Architecture.

SNA network. The part of a user-application network that conforms to the formats and protocols of SNA.

Spool Access Protection. An optional feature of VSE/POWER that restricts individual spool file entry access to user IDs that have been authenticated by having performed a security logon.

*** spool access support (SAS).** A function of VSE/POWER that allows user programs or subsystems running on VSE to access the spool files of VSE/POWER.

*** spool file.** 1. A file that contains output data saved for later processing. 2. One of three VSE/POWER files on disk: queue file, data file, and account file.

*** spooling.** The use of disk storage as buffer storage to reduce processing delays when transferring data between peripheral equipment and the processor of a computer. In VSE, this is done under the control of VSE/POWER.

*** SSL (Secure Sockets Layer).** A security protocol that provides communication privacy. SSL enables client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corp. and RSA Data Security, Inc.

SSP. System Support Program.

startup. The process of performing IPL of the operating system and of getting all subsystems and application programs ready for operation.

static partition. A partition, defined at IPL time and occupying a defined amount of virtual storage that remains constant. Contrast with *dynamic partition*.

station. 1. One of the input or output points of a network that uses communication facilities; for example, the telephone set in the telephone system or the point where the business machine interfaces with the channel on a leased private line. 2. One or more computers, terminals, or devices at a particular location.

statistical data recorder (SDR). A feature that records the cumulative error status of an I/O device on the system recorder file.

sublibrary. In VSE, a subdivision of a library. Members can only be accessed in a sublibrary.

sublibrary directory. An index for the system to locate a member in the accessed sublibrary.

submit. A VSE/POWER function that passes a job to the system for processing.

*** subsystem.** A secondary or subordinate system or program, usually capable of operating independently of, or asynchronously with, the operating system.

subtask. A task that is initiated by the main task or by another subtask.

*** supervisor.** The part of a control program that coordinates the use of resources and maintains the flow of processor operations.

SVA. Shared virtual area.

Synchronous Data Link Control (SDLC). A discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or non-switched links. The configuration of the link connection may be point-to-point, multipoint, or loop.

SYSRES. System residence volume.

*** system console.** A console, usually equipped with a keyboard and display screen for control and communication with the system.

*** system file.** In VSE, a file used by the operating system, for example, the hard-copy file, the recorder file, the page data set.

system logical unit. A logical unit available primarily for operating system use. See also *logical unit name*.

system residence volume (SYSRES). The disk volume on which the system sublibrary is stored and from which the hardware retrieves the initial program load routine for system startup.

system sublibrary. The sublibrary that contains the operating system. It is stored on the system residence volume (SYSRES).

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

System Support Program (SSP). An IBM licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

*** task.** The basic unit of synchronous program execution. A task competes with other tasks for system resources such as processing time and I/O channels.

TCP/IP. Transmission Control Protocol/Internet Protocol. Also, an access facility to the Internet as granted through the 'TCP/IP for VSE/ESA' partition.

*** throughput.** 1. A measure of the amount of work performed by a computer system over a given period of time, for example, jobs per day. (I) (A) 2. In data communication, the total traffic between stations per unit of time.

time event scheduling support. In VSE/POWER, the time event scheduling support offers the possibility to schedule jobs for processing in a partition at a predefined time once or repetitively. The time event scheduling operands of the * $$ JOB statement are used to specify the desired scheduling time.

trace. 1. To record a series of events as they occur.

2. A record of specified events during the run of a program. 3. A program to produce such a record.

track group. In VSE/POWER, the basic organizational unit of a file for CKD devices.

UCB. Universal character set buffer.

* **UCS.** Universal character set.

* **unattended mode.** A mode in which no operator is present or in which no operator station is included at system generation.

unattended node support. In VSE, a set of functions allowing one or more VSE systems to run without an operator being present. The systems are connected to a single central host.

universal character set (UCS). A printer feature that permits the use of a variety of character arrays.

universal character set buffer (UCB). A buffer to hold UCS information.

user console. In z/VSE, a console that receives only those system messages that are specifically directed to it. These are, for example, messages that are issued from a job that was submitted with the request to echo its messages to that console. Contrast with *master console*.

user exit. A programming service provided by an IBM software product that may be requested during the execution of an application program for the purpose of transferring control back to the application program upon the later occurrence of a user-specified event.

* **utility program.** 1. A program in general support of computer processes, for example, a

virtual address area. The virtual range of available program addresses.

virtual address space. In VSE, a subdivision of the virtual address area available to the user for the allocation of private (non-shared) partitions.

virtual disk. A range of up to two gigabytes of contiguous virtual storage addresses that a program can use as workspace. Although the virtual disk exists in storage, it appears as a real FBA disk device to the user program. All I/O operations directed to a virtual disk are intercepted and the data to be written to, or read from, the disk is moved to or from a data space. Like a data space, a virtual disk can hold only user data; it does not contain shared areas, system data or programs. Unlike an address space or a data space, data is not directly addressable on a virtual disk. To manipulate data on a virtual disk, the program has to perform I/O operations.

* **virtual I/O area (VIO).** An extension of the page data set; used by the system as intermediate storage, primarily for control data.

* **virtual machine.** A functional simulation of a computer system and its associated devices.

Virtual Machine/Enterprise Systems Architecture (VM/ESA). An IBM program providing operating system support. Among other services, it manages the resources of a single computer so that multiple computing systems seem to exist. Each of those virtual machines is the functional equivalent of an IBM computer system.

* **virtual mode.** The operating mode of a program which may be paged.

* **virtual partition.** In VSE, a division of the dynamic area of virtual storage.

virtual storage. Addressable space image for the user from which instructions and data are mapped into processor storage locations.

VM/ESA. Virtual Machine/Enterprise Systems Architecture.

volume. A data carrier that is mounted and demounted as a unit, for example, a reel of tape or a disk pack. (I) Some disk units have no demountable packs. In that case, a volume is the portion available to one read/write mechanism.

VSE (Virtual Storage Extended). A system that consists of a basic operating system (VSE/Advanced Functions) and any IBM-supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware it controls form a complete computing system. Its current version is called z/VSE.

VSE/Advanced Functions. The basic operating-system component of z/VSE.

VSE library. A collection of programs in various forms and storage dumps stored on disk. The form of a program is indicated by its member type such as source code, object module, phase, or procedure. A VSE library consists of at least one sublibrary which can contain any type of member.

VSE/POWER. A program primarily used to spool input and output. The program's networking functions enable a z/VSE system to exchange files with or run jobs on another remote processor. It is part of VSE

Central Functions, a base program of z/VSE. The acronym POWER stands for 'Priority Output Writers, Execution Processors, and Input Readers'.

VSE/SP Unique Code. A program that provides productivity support such as the Interactive Interface and workstation-related support. It is part of VSE Central Functions, a base program of z/VSE.

wait for run subqueue. In VSE/POWER, a subqueue of the reader queue with dispatchable jobs ordered in execution start time sequence.

workunit. A set of instructions that is processed from the point of selection by the Turbo Dispatcher until the next interrupt. Parallel work units (PA) are customer applications in a batch or online environment. Non-parallel work units cannot be processed in parallel, for example most system services except VSE/POWER.

z/VSE. See VSE.

31-bit addressing. Provides addressability for address spaces of up to 2 gigabytes. (The maximum amount of addressable storage in previous systems was 16 megabytes.).

Index

Special characters

Numerics

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

Readers' Comments — We'd Like to Hear from You

IBM z/VSE
VSE Central Functions
VSE/POWER
Administration and Operation
Version 7 Release 1

Publication No. SC33-8247-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

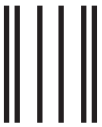
Company or Organization

Phone No.

Readers' Comments — We'd Like to Hear from You
SC33-8247-00

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

File Number: S370/S390-34
Program Number: 5686-066

Printed in USA

SC33-8247-00

Spine information:

IBM z/VSE
VSE Central Functions

VSE/POWER Administration and Operation

Version 7 Release 1

SC33-8247-00