SY33-8576-1
File No. S370-36

Systems

DOS/VS POWER/VS
Logic Part 1

**Program Number 5745-SC-PWR
Release 34**

IBM

Second Edition (April 1977)

This edition includes the support of RJE, SNA for the 3790 Communication System and various quality improvements. Changes and additions to the text and illustrations are indicated by a vertical line to the left of the change.

This edition, as amended by Technical Newsletter SN33-9240, applies to version 5, release 34 of the Disk Operating System/Virtual Storage, DOS/VS, and to any subsequent versions and releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography,* GC20-0001, for the editions that are applicable and current.

Note: *For the availability dates of features and programming support described in this manual, please contact your IBM representative or the IBM branch office serving your locality.*

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

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to *IBM Germany, Publications Dept., Schwertstrasse 58, D-7032 Sindelfingen, Germany.* Comments become the property of IBM.

Technical Newsletter

This Newsletter No.	SN33-9240
Date	November 28, 1977
Base Publication No.	SY33-8576-1
File No.	S370-36
Prerequisite Newsletters	None

DOS/VS POWER/VS Logic, Part 1

This Technical Newsletter, a part of the independent component release (ICR) of support for the IBM 3800 Printing Subsystem under Release 34 of the IBM Disk Operating System/Virtual Storage, DOS/VS, and DOS/VS POWER/VS, provides replacement pages for your publication. Information contained on these pages applies only if the ICRs are installed on your system. You need not insert the pages if they are not installed. These replacement pages remain in effect for subsequent DOS/VS releases unless specifically altered. Pages to be replaced are:

Cover - Charts
21-24
27-38
45-48 (46.1, 46.2, 48.1 added)
50.1-52 (50.1, 50.2 added)
55-58
69-70
73-74
93-106 (100.1, 104.1 added)
109-124
129-136.1 (136.1 added)
143-144
147-150.1 (150.1 added)
155-160
165-168
171-174
181-182
187-188
201-204
207-208.4 (208.1-208.4 added)
211-212
215-216
221-224
227-228
231-232
255-256
259-260

A technical change to the text or to an illustration is indicated by a vertical line to the left of the change.

Summary of Amendments

Changes to the system are summarized under "Summary of Amendments" preceding the Contents.

For a complete list of publications that support the DOS/VS IBM 3800 Printing Subsystem ICR, and the DOS/VS POWER/VS IBM 3800 ICR, see the *DOS/VS IBM 3800 Printing Subsystem Programmer's Guide*, GC26-3900.

Note: Please insert this page in your publication to provide a record of changes.

IBM Germany, Publications Department, Schwertstrasse 58, D-7032 Sindelfingen, Germany

Printed in U.S.A.

This Manual . . .

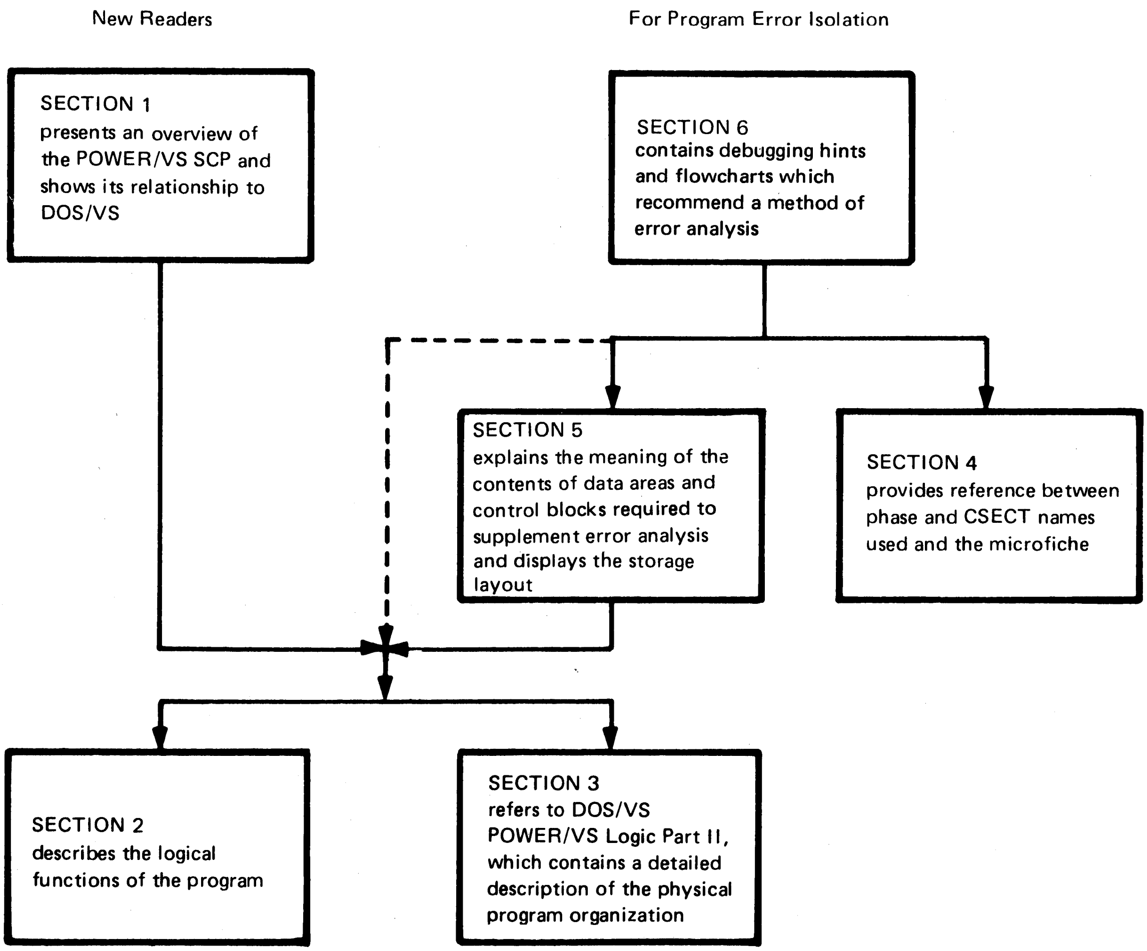
which should be used together with *DOS/VS POWER/VS Logic Part 2,* SY33-8577, contains information about the internal logic of POWER/VS. To use it effectively, the reader should be familiar with the concepts and facilities of POWER/VS as they are described in the following IBM DOS/VS manuals:

Introduction to DOS/VS, GC33-5370
DOS/VS POWER/VS Installation Guide and Reference, GC33-6048
DOS/VS POWER/VS Work Station User's Guide, GC33-6049

RJE,SNA users should also be familiar with the VTAM concepts and facilities as they are described in:

VTAM Concepts and Planning, GC27-6998
VTAM Macro Language Reference, GC27-6995

The manual is divided into six sections designed to be used as follows:


New Readers
For Program Error Isolation
SECTION 1 presents an overview of the POWER/VS SCP and shows its relationship to DOS/VS
SECTION 6 contains debugging hints and flowcharts which recommend a method of error analysis
SECTION 5 explains the meaning of the contents of data areas and control blocks required to supplement error analysis and displays the storage layout
SECTION 4 provides reference between phase and CSECT names used and the microfiche
SECTION 2 describes the logical functions of the program
SECTION 3 refers to DOS/VS POWER/VS Logic Part II, which contains a detailed description of the physical program organization

Summary of Amendments

INDEPENDENT COMPONENT RELEASE OF IBM 3800 PRINTING SUBSYSTEM SUPPORT

Technical Newsletter SN33-9240 documents changes to support the IBM 3800 Printing Subsystem under DOS/VS POWER/VS.

RELEASE 34

This revised edition of the Logic Manual documents extended support for Remote Job Entry with SNA terminal support (RJE,SNA) using the 3790 Communication System with RJE Facility, plus quality improvement items.

RELEASE 33

Technical Newsletter SN33-9207 documents new support for the internal reader/writer (PUTSPOOL, GETSPOOL, and CTLSPOOL) and improvements in performance, handling, and accounting.

RELEASE 32 (RJE,SNA)

This edition of the Logic Manual documents new support for Remote Job Entry with SNA terminal support (RJE,SNA).

Manual usability has been improved through dividing the Logic Manual into two parts. Part 1 describes the overall logic, data area layouts, a directory of the modules and macros and messages, and other general logic information. Part 2 (SY33-8577) contains the detailed logic description of each POWER/VS module.

Contents

Figures

Charts

Section 1: Introduction

This section contains an overview of the POWER/VS System Control Program. It is organized as follows:

Purpose and Functions of POWER/VS. A general description of POWER/VS and the way its major functions are performed under DOS/VS.

Communication with POWER/VS. A summary of the POWER/VS Operator Command Language (POCL) and the Job Entry Command Language, which allow the user to control POWER/VS operations. The format of the messages issued by POWER/VS is also explained.

Environmental Requirements. This chapter gives a tabular presentation of the programming requirements for the various functions of POWER/VS. Then the basic organization of the POWER/VS partition is shown, with its storage requirements. A section entitled Hardware Support lists the machines and devices which are supported by POWER/VS.

Purpose and Functions of POWER/VS

POWER/VS (Priority Output Writers, Execution Processors and Input Readers/Virtual Storage) is a System Control Program which is an automatic spooling processor and priority scheduler running under the control of the DOS/VS supervisor. The POWER/VS SCP occupies a virtual partition in which it is initiated and can service from one to four partitions of a lower dispatching priority. Input to supported partitions is first spooled onto intermediate disk storage. When the supported partition commences execution, I/O requests to reader devices are intercepted and satisfied from intermediate storage via I/O data areas in the POWER/VS partition. The partition in which the POWER/VS SCP is initiated is referred to as the POWER/VS partition. Output requests to list and punch devices are also intercepted, with the output being stored in output data areas of the POWER/VS partition and later transferred to disk. Printing and punching of the output from disk is carried out when requested by the operator. Under the control of POWER/VS, programs may be executed in either real or virtual mode.

Three major operations are performed under POWER/VS control:

Read User job information is read from a reader device and spooled to intermediate storage (DASD). The PUTSPOOL macro interface can be optionally used to submit a job stream from the user's buffer area to intermediate storage. The job is executed under the control of the POWER/VS execution processor to meet user program requests.

List List output generated by the user program is spooled to intermediate storage (DASD or magnetic tape) before being transferred to a list device, normally a line printer.

Punch Punch output generated by the user program is spooled to intermediate storage (DASD or magnetic tape) before being transferred to a punch device, normally a card punch. The GETSPOOL macro interface can be optionally used to request retrieval of printer output.

Intermediate storage therefore contains user input and output data spooled to and from it under the control of POWER/VS.

POWER/VS SYSTEM VERSIONS

The user can generate several different versions, depending on the parameters specified | in the POWER macro, and optional PLINE, PRMT, and PCPTAB macros.

The partition in which the POWER/VS SCP is initiated is referred to as the POWER/VS partition.

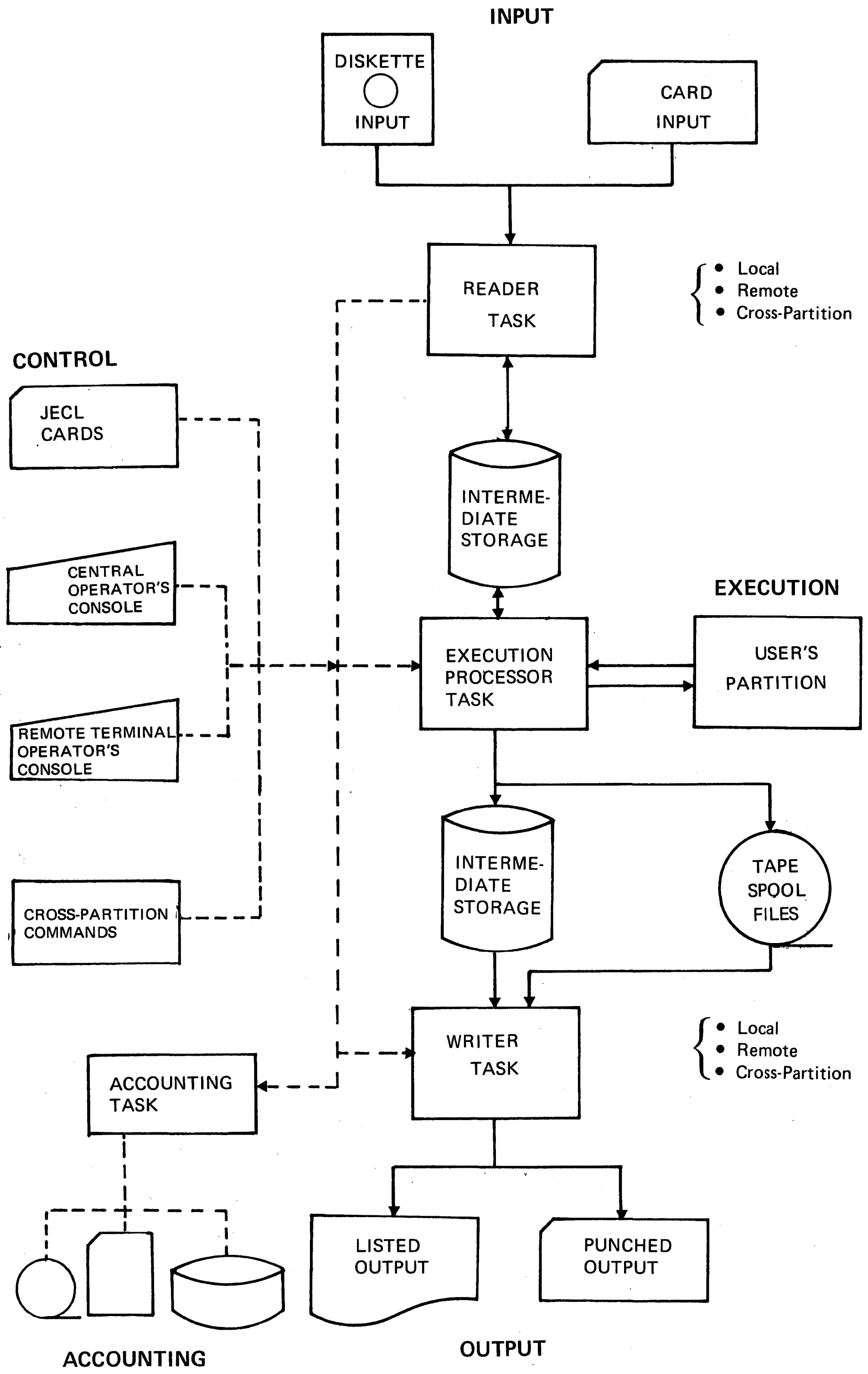


Figure 1.1. The major functions of POWER/VS, data flow, and operator communication

POWER/VS TASKS

In order to perform the required operations concurrently, POWER/VS is structured into a series of asynchronously executed tasks. All necessary task support is provided within the POWER/VS system and does not presuppose Multitasking Support within the DOS/VS supervisor.

MAIN TASK. Initiates and terminates POWER/VS execution and controls the execution of all other POWER/VS tasks by means of a private multitasking mechanism.

COMMAND TASK. Handles all commands submitted by the system operator and initiates and terminates other POWER/VS tasks.

WAIT TASK. Transfers the POWER/VS partition to and from the wait state to meet system requirements.

RJE,BSC LINE MANAGER TASK. Controls all line activities with remote terminals. The task is attached when the first line is started and detached when the last line is stopped.

RJE,SNA MANAGER TASK. Controls the activation of transmission processing to and from a remote SNA work station on a demand basis. The task is attached when the central operator issues a PSTART RJE,SNA command. The SNA manager also attaches a DOS/VS subtask in which the interface with the VTAM access method is opened.

RJE,SNA LOGON TASK NO. 1. Initializes session work areas and does validity checking of logon request.

RJE,SNA LOGON TASK NO. 2. Establishes a session between POWER/VS and a remote SNA work station.

RJE,SNA LOGOFF TASK. Terminates a session between POWER/VS and a remote SNA work station.

RJE,SNA MESSAGE TASK. Sends messages to a remote SNA work station.

THE SPOOL MANAGER TASK. Controls the activation and deactiviation of the internal reader task and the spool/command manager list task. The task is attached during POWER/VS initialization when SPOOL=YES is specified in POWER macro and detached at POWER/VS termination.

READ TASKS. (2) Perform the first part of the read operation and transfer information from a peripheral reader to intermediate direct access storage. The operator may call for concurrent execution of as many read tasks as he has physical readers available. Each read task is therefore associated with a specific reader.

RJE,BSC READ TASKS. (2) Perform the read operation on an RJE,BSC line. The central operator may call for concurrent execution of as many RJE,BSC Read Tasks as he has RJE,BSC lines available. Each RJE,BSC Read Task has the standard name 'RDR' assigned to it. Different RJE,BSC Read Tasks are further distinguished by suffixing the line address to this standard name.

RJE,SNA READ TASKS. Perform the read operation from a remote SNA work station.

INTERNAL READER TASK. Performs the read operation for the PUTSPOOL POWER/VS cross-partition communication macro interface.

EXECUTION READ TASKS. (1)(2) Perform the second part of the read operation and transfer information from intermediate direct access storage to meet the read requests of the user program. The operator may call for concurrent execution of as many execution read tasks as he has partitions under POWER/VS control. Each execution read task is therefore associated with a specific partition.

EXECUTION LIST TASKS. (1)(3) Perform the first part of the list function and transfer information from the user program to intermediate direct access storage. The operator may call for concurrent execution of as many execution list tasks as he has partitions under POWER/VS control. Each execution list task is therefore associated with a specific partition.

EXECUTION PUNCH TASKS. (1)(3) Perform the first part of the punch function and transfer information from the user program to intermediate direct access storage. The operator may call for concurrent execution of as many execution punch tasks as he has partitions under POWER/VS control. Each execution punch task is therefore associated with a specific partition.

LIST TASKS. (3) Perform the second part of the list operation and transfer information from intermediate direct access storage to the printer. The operator may call for concurrent execution of as many list tasks as he has physical printers available. Each list task is therefore associated with a specific printer.

RJE,BSC LIST TASKS. (3) Perform the list operation on an RJE,BSC line. Only the remote operator may call for concurrent execution of as many RJE,BSC List Tasks as he has physical printers available at the terminal. Each RJE,BSC List Task has the standard name 'LST' assigned to it. Different RJE,BSC List Tasks are further distinguished by suffixing the line address and an identifier for the physical printer to this standard name.

RJE,SNA LIST TASKS. Perform the list operation to a remote SNA work station.

SPOOL/COMMAND MANAGER LIST TASK. Perform the list retrieval (GETSPOOL) and the command invokation (CTLSPOOL) for the POWER/VS cross-partition communication macro interface.

PUNCH TASKS. (3) Perform the second part of the punch function and transfer information from intermediate direct access storage to the punch. The operator may call for concurrent execution of as many punch tasks as he has physical punches available. Each punch task is therefore associated with a specific punch.

RJE,BSC PUNCH TASKS. (3) Perform the PUNCH operation on an RJE,BSC line. Only the remote operator may call for concurrent execution of as many RJE,BSC Punch Tasks as he has physical punches available at the terminal. Each RJE,BSC Punch Task has the standard name 'PUN' assigned to it. Different RJE,BSC Punch Tasks are further distinguished by suffixing the line address and an identifier for the physical punch to this standard name.

RJE,SNA PUNCH TASKS. Perform the punch operation to a remote SNA work station.

ACCOUNT TASK. Supports the POWER/VS job accounting option (together with DOS/VS JAI). It gives the user the option to either save the account file on another medium (tape, disk, cards) or delete the account file. The contents and format of the account records are not checked or changed by this task.

STATUS TASK. The purpose of this task is to scan the Queue file and print the status report on SYSLOG, a line printer, or a terminal printer.

Notes:

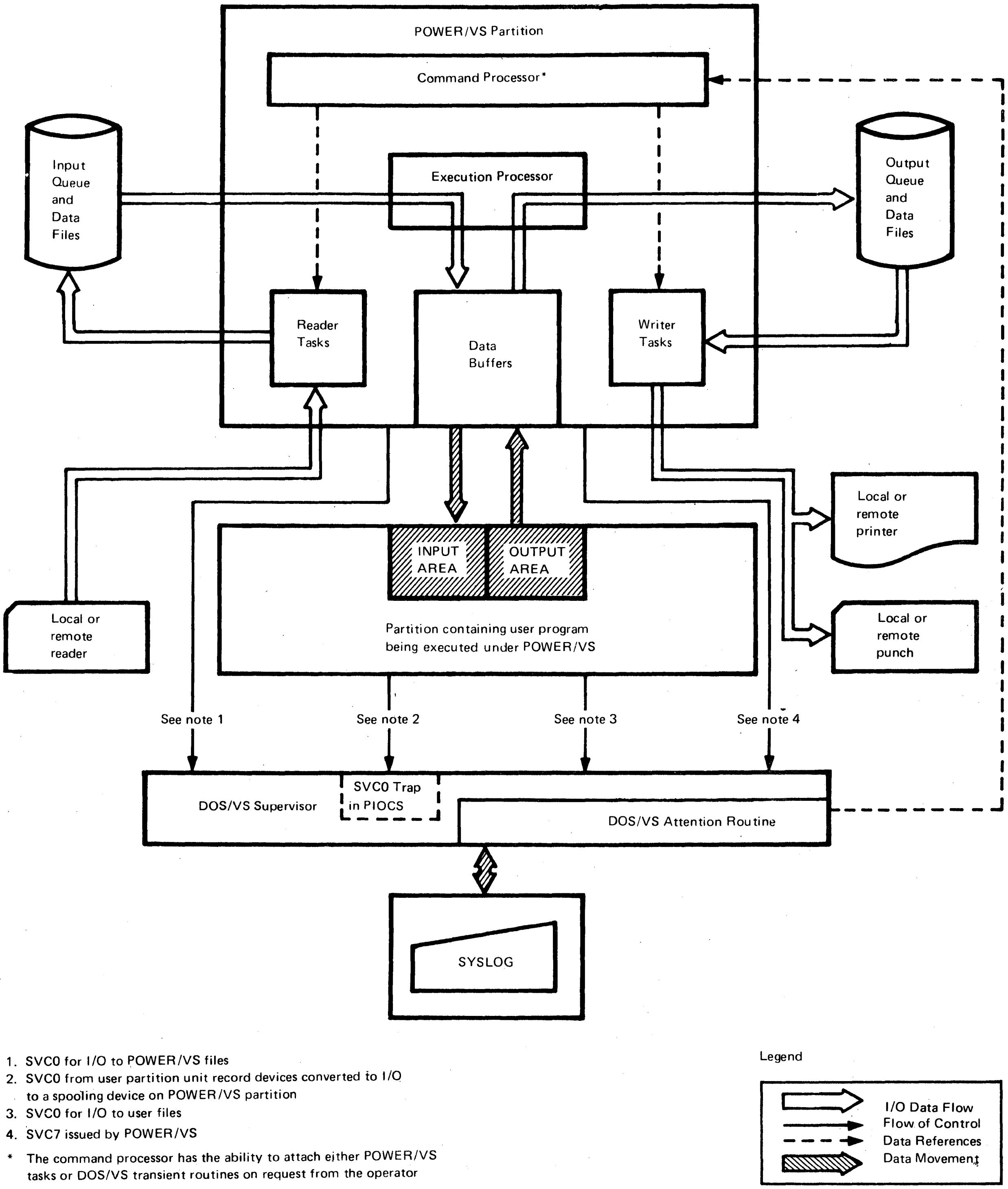
1. Execution Read Tasks, Execution List Tasks, and Execution Punch Tasks are collectively referred to as EXECUTION PROCESSOR TASKS.

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

3. Each Read Task is divided into two parts, PHYSICAL READER (PR and/or ER), which performs the device-dependent functions related to data collection from a specified device or family of devices, and LOGICAL READER (LR), which performs the logical functions related to entering input data into the POWER/VS Data File and inserting a new Queue Entry into the correct position in the POWER/VS Queue File. These two parts are linked by means of a high-level logical record interface.

4. Similarly, each List Task is divided into PHYSICAL LIST (PL), or PHYSICAL PUNCH (PP), and LOGICAL WRITER (LW). The physical part of a task performs device dependent functions for printer or punch, respectively. Logical Write retrieves data from the List Queue or the Punch Queue, as required. In each case a high-level logical record interface is defined to connect the two parts of the task.

Figure 1.2 shows the relationship of the user program to the POWER/VS partition and tasks and to the DOS/VS supervisor.



1. SVC0 for I/O to POWER/VS files
2. SVC0 from user partition unit record devices converted to I/O to a spooling device on POWER/VS partition
3. SVC0 for I/O to user files
4. SVC7 issued by POWER/VS

* The command processor has the ability to attach either POWER/VS tasks or DOS/VS transient routines on request from the operator

Figure 1.2. Relationship between POWER/VS, DOS/VS, and the program running under the control of POWER/VS

POWER/VS DIRECT ACCESS FILES

Any of the following direct access devices may be used to contain the POWER/VS files:

2314/2319
3330/3333
3340
3350

The input stream provided by the user to POWER/VS is broken up into a series of discrete jobs, each with its own identifying name, assigned by the user, and sequence number, assigned by the POWER/VS system at the time the job enters the system. Each job is described by summary records in direct access storage which form a Read Queue Entry.

The DOS/VS // JOB and /& statements function as job delimiters if no JECL is used. Otherwise, a job is delimited by the JECL * $$ JOB and * $$ EOJ statements.

List and punch output generated by a program executed under control of POWER/VS is similarly described by summary records which form a List Queue Entry and a Punch Queue Entry respectively. A program may create multiple List and Punch Queue Entries for a job either

- by using multiple LST/PUN statements (data-driven output segmentation), or
- by usage of the Segmented Output Feature (count-driven output segmentation), or
- by usage of the LFCB macro (program-driven output segmentation).

The three types of Queue Entry Record are logically assembled into three corresponding queues. The Reader Queue contains Read Queue Entries; the List Queue contains List Queue Entries; and the Punch Queue contains Punch Queue Entries. These three queues service all of the partitions running under POWER/VS control.

The Reader Queue is also referred to as the Input Queue; the term Output Queue is also used to refer to the List Queue and the Punch Queue. Like the reader and writer tasks, the POWER/VS queues may be referenced as RDR, LST, or PUN.

A summary of all the POWER/VS Queues is maintained in the form of the POWER/VS Master Record on the Queue file. This record provides the POWER/VS System with a 'warm start' capability.

DATA FILE

Figure 1.1 shows the transfer of program input and output records to and from intermediate storage under the control of POWER/VS. Program data spooled to intermediate storage constitutes the POWER/VS data file.

The user may assign from one to five separate extents of direct access storage to the data file, using logical units SYS002 to SYS006 within the POWER/VS partition. Each of the extents (with the file name IJDFILE) must start on a cylinder boundary and must contain an integral number of cylinders.

Track Group

The space assigned to the data file is managed by dividing it into units called 'track groups'. The track group is the basic organizational unit of the data file, and consists of an integral number of tracks. The track group size may vary from a single track to an entire cylinder. An appropriate track group size is calculated by POWER/VS at initialization time on the basis of the amount of disk storage made available. However, the size can be specified in the TRACKGP parameter of the POWER macro (refer to DOS/VS System Generation).

Data Block Size

This is the length of physical records written to the data file. Data block size is specified by the DBLK parameter of the POWER macro. The maximum data block size that may be specified is 2008 bytes, and the minimum is 544 bytes (refer to DOS/VS POWER/VS Installation Guide and Reference, GC33-6048). If it is omitted or 0, and a cold start is performed, the value assumed is based on the DASD assigned to the POWER/VS data file.

A minimum DBLK size of 608 bytes is needed to support the printer device type 3211 which needs 592 bytes for the UCB (512 bytes for UCB plus 80 bytes message area) and for control information POWER/VS needs 16 bytes.

Each page, restricted by the storage management, has 2048 bytes and is controlled by a 24 bytes Page Control Block (PCB) and a 8 bytes Buffer Control Word (BCW). The BCW is located at the begin and the end of each buffer making a total of 24 + 8 + 8 = 40 control bytes. This leaves 2008 bytes and this is the maximum DBLK size.

QUEUE FILE

In order to control data spooled on the data file, POWER/VS requires a queue file assigned as a separate extent (with the file name IJQFILE) to the logical unit SYS001. Each record in the file has a length of 152 bytes. The queue file contains two types of records; a Master Record (MR) of which there is only one, and Queue Records (QR) of which there can be 1024.

Queue Records

The input stream provided by the user to POWER/VS is broken up into a series of discrete jobs, each with its own identifying name, assigned by the user, and with a sequence number, assigned by the POWER/VS system at the time the job enters the system.

An input job stream, together with its data, is written to one or more track groups on the data file. The seek address of each track group is kept in a queue record on the queue file. When a track group becomes full a new queue record is constructed until the complete input job has been written on the data file.

The collection of queue records thus constructed is called a queue set, and this set, together with its associated track group(s), constitutes a queue entry. Figure 1.3 illustrates these relationships.

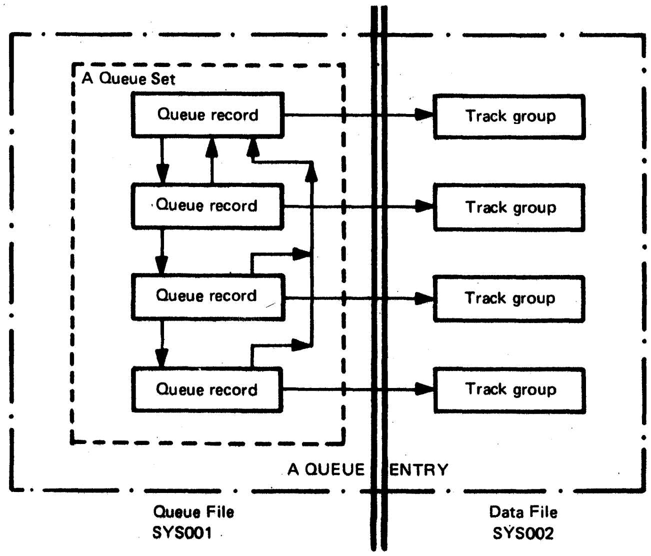


Figure 1.3. Relationship between a Queue Set, Queue Records, and a Queue Entry

The illustration shows a job which required four track groups on the data file.

ACCOUNT SUPPORT

If the DOS/VS supervisor has been generated to support the job accounting interface (JAI) option, which is to be used by programs under the control of POWER/VS, the following three conditions must be fulfilled:

- JA=YES or JA=(n1,n2,n3,n4,n5) must be specified in the FOPT macro during generation of the DOS/VS supervisor.
- ACCOUNT=YES must be specified in the POWER macro. Bit 0 of the POWER/VS flag byte (Partition COMREG, displacement decimal 164 or hexadecimal A4) will be on if POWER/VS supports accounting.
- DLBL and EXTENT statements (with the file name IJAFILE) must be provided, and a single extent must be assigned to SYS000 for the POWER/VS account file.

ACCOUNT FILE

Accounting information is collected both from the POWER/VS partition and from the DOS/VS JAI for partitions under the control of POWER/VS, and is merged within the Account File together with full identifying information. The user need not provide his own data collection routine, since POWER/VS collects all the data included in the user portion of the job accounting interface table.

The normal POWER/VS collection of accounting information is used for PUTSPOOL/GETSPOOL processing.

The POWER/VS account file may contain six types of records:

Read account record - created for each read queue entry entered into POWER/VS.

List account record - created for each list queue entry processed by a list task.

Punch account record - created for each punch queue entry processed by a punch task.

Execution account record- created during program execution on the basis of one record per job step.

Line account record - created for each RJE,BSC user session. When the session is terminated, the SIGNOFF processor writes the account record to the account file.

SNA account record - created for each RJE,SNA user session. When the session is terminated, the SNA LOGOFF processor writes the account record to the account file.

POWER/VS OPERATOR COMMAND LANGUAGE (POCL)

POWER/VS provides an Operator Command Language (POCL) which allows the central system operator and the remote terminal operator to communicate with the system. Four types of command are provided:

- Task Management Commands allow the operator to initiate and terminate POWER/VS Tasks (except spool management tasks).

  Task management commands are only applicable to the RJE writer task. The reader task is started by the central operator when he brings up the line. Its operation is controlled by the system.

- Queue Management Commands allow the operator to display and modify the contents of POWER/VS Queue Entries.

  Queue management commands are only applicable to jobs that are submitted by or routed to the same remote ID as the one issuing the command.

- List Control Commands allow the operator to perform certain device-dependent operations on line printers.

- Terminal Control Commands allow the remote operator to initiate and terminate POWER/VS RJE tasks.

Figure 1.4 shows the abbreviated and extended command codes available to the central operator. Figure 1.5 lists the commands to be used by the remote terminal operator.

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

For further details about these commands, refer to DOS/VS Operating Procedures.

Figure 1.4. Central operator commands

Type	Command	Function
Terminal control	LOGON SIGNON SIGNOFF LOGOFF	start an RJE,SNA user session start an RJE,BSC user session terminate an RJE,BSC or RJE,SNA user session terminate an RJE,SNA user session
Task management	START STOP FLUSH RESTART GO	start a writer task or message queuing stop a writer task or message queuing flush an active job restart a writer task reactivate a writer task
Queue management	DISPLAY ALTER DELETE RELEASE BRDCST	display job status alter job attributes delete a job or a message release a job transmit a message
List control	SETUP	print page layout

For further details, refer to DOS/VS Operating Procedures.

Figure 1.5. Remote terminal operator commands

JOB ENTRY CONTROL LANGUAGE (JECL)

POWER/VS provides a JECL to assist the user in delimiting jobs to the system and to allow him to specify special requirements that may apply to particular jobs. JECL supplements but does not replace the Job Control Language (JCL) provided by DOS/VS itself. The JCL statements required for normal DOS/VS system operation are also required when operating under POWER/VS.

JECL Statements	Function
* $$ CTL	specifies a default input class
* $$ JOB	indicates the beginning of a job and provides handling information
* $$ EOJ	indicates the end of a job
* $$ RDR	inserts a diskette file into the input stream
* $$ LST * $$ PRT	provides handling information for printed output
* $$ PUN	provides handling information for punched output
* $$ SLI	inserts data from a sublibrary into the job stream
* $$/*	indicates the end of a job step (used for the SLI statement only)
* $$/&	indicates the end of a job (used for the SLI statement only)
* $$ DATA	inserts data into a book in a source statement library

For details refer to DOS/VS System Control Statements.

Figure 1.6. Job Entry Control Language (JECL) statements

FORMAT OF POWER/VS OPERATOR MESSAGES

Messages sent by the POWER/VS system to SYSLOG, SYSLST, or to a terminal may have the following formats:

```
1QnnI or 1RnnI or 1VnnI  information-type message
1QnnD                    decision-type message
1QnnA or 1RnnA or 1VnnA  action-type message
```

where:

```
Q  = POWER/VS message indicator (general)
R  = POWER/VS message indicator (for messages issued by RJE,BSC and command processor
     tasks)
V  = POWER/VS message indicator (for messages issued by RJE,SNA tasks)
nn = message identification number.
```

I-type messages are for the operator's information only; no response is required. Processing continues normally. I-type messages for remote terminals are stored in the list queue.

D-type messages require an immediate reply from the operator. The console keyboard is unlocked for the operator to reply, and the system waits for the operator's reply. For systems running under DOS/VS Release 34 SCP, the console is not available to the rest of the system, and hence, decision-type messages will only be issued in situations where an operator decision are absolutely required. This limitation does not apply, however, for users running with DOS/VS Advanced Functions (available as a licensed program). See Advanced Functions DOS/VS, General Information manual, GC33-6048. D-type messages are never sent to remote terminals.

A-type messages require some action from the operator. A-type messages for remote terminals are directly displayed on the remote printer. The task issuing the message is put in the wait state. It may be reactivated

- centrally by:
  - PGO command. This command is normally entered by the operator after he has taken the appropriate action.
  - PFLUSH command to discontinue the job.
- remotely by:
  - RESTART command to continue the job.
  - FLUSH command to discontinue the job.

Most messages contain a task identifier to identify the task issuing the message. For reader/writer or RJE tasks, this identifier has the format ttttt, where ttttt stands for RDR, PUN, or LSTn. For execution processors, the identifier has the format pp, where pp stands for BG or Fn, where "n" stands for the partition number.

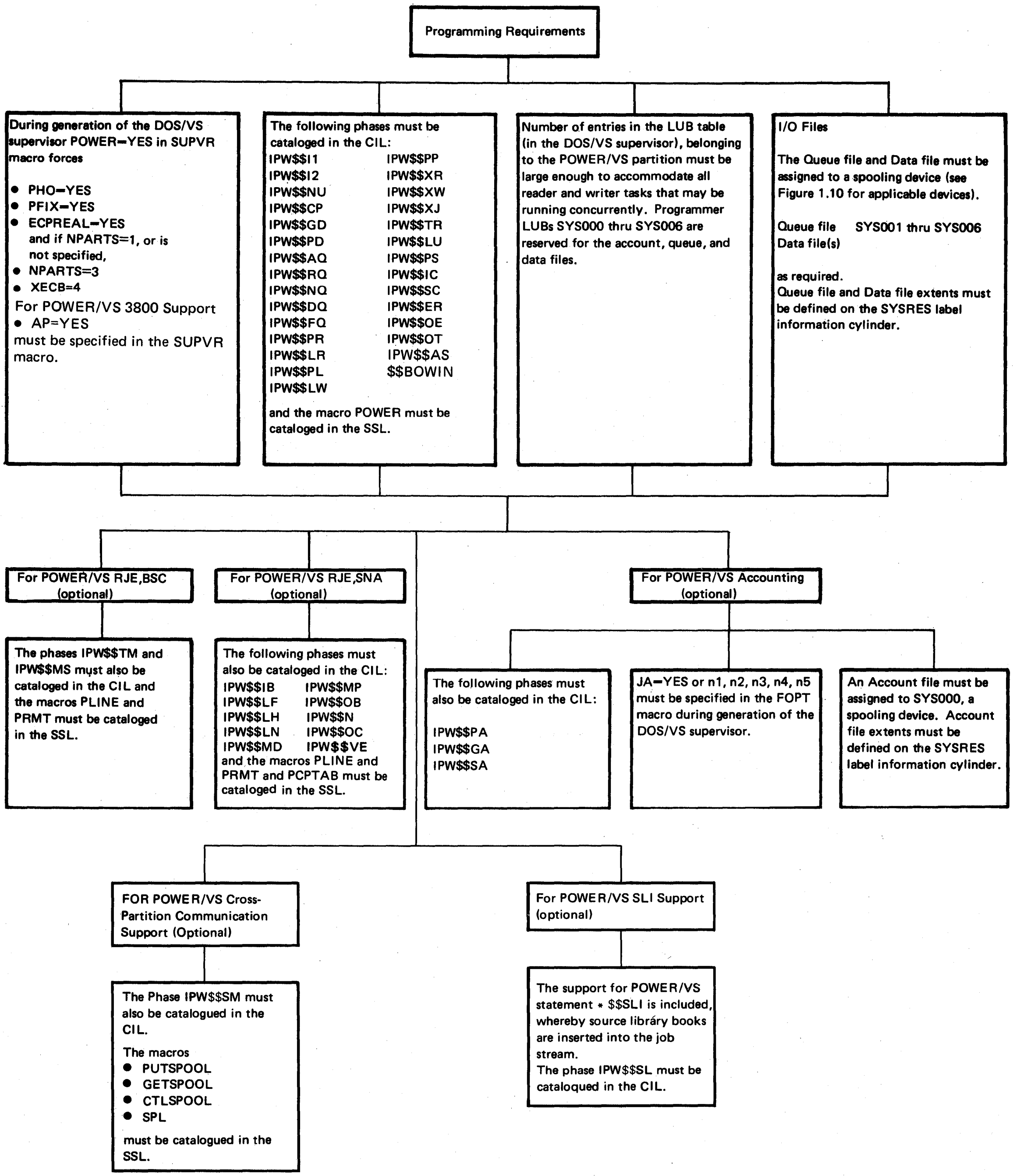
There are four types of messages:

- Initialization messages
- Execution processor and reader/writer task messages
- Operator command messages
- Remote (RJE) messages.

All messages, issued by POWER/VS, are listed in Section 4.

Environmental Requirements

PROGRAMMING REQUIREMENTS



Programming Requirements

During generation of the DOS/VS supervisor POWER—YES in SUPVR macro forces

- PHO—YES
- PFIX—YES
- ECPREAL—YES and if NPARTS=1, or is not specified,
- NPARTS=3
- XECB=4

For POWER/VS 3800 Support

- AP=YES

must be specified in the SUPVR macro.

The following phases must be cataloged in the CIL:

IPW$$I1	IPW$$PP
IPW$$I2	IPW$$XR
IPW$$NU	IPW$$XW
IPW$$CP	IPW$$XJ
IPW$$GD	IPW$$TR
IPW$$PD	IPW$$LU
IPW$$AQ	IPW$$PS
IPW$$RQ	IPW$$IC
IPW$$NQ	IPW$$SC
IPW$$DQ	IPW$$ER
IPW$$FQ	IPW$$OE
IPW$$PR	IPW$$OT
IPW$$LR	IPW$$AS
IPW$$PL	$$BOWIN
IPW$$LW	

and the macro POWER must be cataloged in the SSL.

Number of entries in the LUB table (in the DOS/VS supervisor), belonging to the POWER/VS partition must be large enough to accommodate all reader and writer tasks that may be running concurrently. Programmer LUBs SYS000 thru SYS006 are reserved for the account, queue, and data files.

I/O Files

The Queue file and Data file must be assigned to a spooling device (see Figure 1.10 for applicable devices).

Queue file SYS001 thru SYS006
Data file(s)

as required.
Queue file and Data file extents must be defined on the SYSRES label information cylinder.

For POWER/VS RJE,BSC (optional)

The phases IPW$$TM and IPW$$MS myst also be cataloged in the CIL and the macros PLINE and PRMT must be cataloged in the SSL.

For POWER/VS RJE,SNA (optional)

The following phases must also be cataloged in the CIL:

IPW$$IB	IPW$$MP
IPW$$LF	IPW$$OB
IPW$$LH	IPW$$N
IPW$$LN	IPW$$OC
IPW$$MD	IPW$$VE

and the macros PLINE and PRMT and PCPTAB must be cataloged in the SSL.

For POWER/VS Accounting (optional)

The following phases must also be cataloged in the CIL:

IPW$$PA
IPW$$GA
IPW$$SA

JA—YES or n1, n2, n3, n4, n5 must be specified in the FOPT macro during generation of the DOS/VS supervisor.

An Account file must be assigned to SYS000, a spooling device. Account file extents must be defined on the SYSRES label information cylinder.

FOR POWER/VS Cross-Partition Communication Support (Optional)

The Phase IPW$$SM must also be catalogued in the CIL.

The macros

- PUTSPOOL
- GETSPOOL
- CTLSPOOL
- SPL

must be catalogued in the SSL.

For POWER/VS SLI Support (optional)

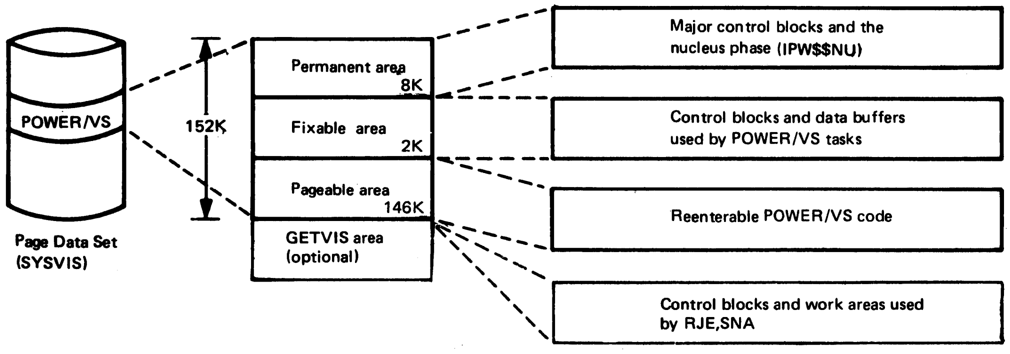
The support for POWER/VS statement * $$SLI is included, whereby source library books are inserted into the job stream.
The phase IPW$$SL must be cataloqued in the CIL.

Figure 1.7. Programming requirements for POWER/VS

STORAGE REQUIREMENTS AND ALLOCATIONS

Virtual Storage

The virtual address space of the POWER/VS partition consists of three major areas as shown in Figure 1.8.


POWER/VS
Page Data Set (SYSVIS)
152K
Permanent area 8K
Fixable area 2K
Pageable area 146K
GETVIS area (optional)
Major control blocks and the nucleus phase (IPW$$NU)
Control blocks and data buffers used by POWER/VS tasks
Reenterable POWER/VS code
Control blocks and work areas used by RJE,SNA

Minimum size of virtual partition (no optional support) = 142K plus size of real partition

Figure 1.8. Basic organization of the POWER/VS partition

- The permanent area, consisting of three pages (8K), is permanently fixed (PFIXed) in the real partition during initialization of POWER/VS. These pages are freed (PFREEed) only when POWER/VS is terminated.
- The fixable area consists of pages that are fixed when a task is started and freed when they are no longer required for the completion of this task.
- The pageable area consists of pages that are allowed to be paged out whenever DOS/VS requires additional real storage.
- The GETVIS area contains control blocks and work areas used by RJE,SNA or for printer setup processing.

Real Storage

The minimum real address space must be equal to the size of the permanent area (6K) plus the fixable area. The size of the fixable area (minimum = 4K) depends on the size of the data buffers, which is determined by the value specified in the DBLK parameter of the POWER macro during DOS/VS supervisor generation. (The minimum size of a data buffer = 544 bytes, and the maximum size = 2008 bytes.)

For a description of how to calculate POWER/VS storage requirements refer to DOS/VS POWER/VS Installation Guide and Reference, GC33-6048.

HARDWARE SUPPORTED

Machine Requirements

Any Model of System/370 supported by DOS/VS.

Devices Supported

Readers	Printers[4]	Punches	Spooling Devices	Terminals
1442	1403	1442	2400 series[1] [2]	2770[3]
2501	1443	2520	3400 series[2]	2780
2520	3203	2540	2314/2319	3771•
2540	3211	2560	3330/3333	3773[6]
2560[5]	3800	3525	3340	3774•
3504	5203	5425	3350	3775[6]
3505				3776•
3525				3780[3]
3540				3790[7]
5425•				

Figure 1.10. Devices supported by POWER/VS

Notes:

[1] The IBM 2495 tape cartridge reader does not belong to this series.

[2] Only as list and punch output devices.
For 7-track tape units the data convert feature is required.

[3] The following I/O devices are supported by RJE,BSC.

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

Teleprocessing control units supported by RJE,BSC are:

IBM 2701 Data Adapter Unit with SDA (Type 2)
IBM 2703 Transmission Control Unit
IBM 3704 Communications Adapter in 2703 emulation mode
IBM 3705 Communications Adapter in 2703 emulation mode
Integration Communication Adapter for the Models 115, 125, and 135.

Restrictions:

- TP connections must be point-to-point on switched or non-switched lines.
- Multipoint connections are not supported.
- Terminals and control units having the multipoint line control or multipoint data link control features are prohibited. (Connecting such a terminal or control unit to the POWER/RJE,BSC system will cause continuous error recovery processing.)

[4] The Universal Character Set Buffer (UCB) and Forms Control Buffer (FCB) features are supported by POWER/VS. The execution processor will accept UCB and FCB load requests from the various supported partitions for appropriate action at list time. On encountering an FCB load command, the execution processor will update the internal buffer representation to reflect the new buffer.

5 POWER/VS will support the 2560 and 5425 devices as SYSRDR, SYSIPT, and SYSPCH, and will handle 96 column input and output records for the 5425. The following functions are included:

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

However, there are some restrictions:

- POWER/VS cannot update or interpret a card input file
- Read column binary is not supported
- No program-controlled stacker selection on input.
- If a 2560 Punch or 5425 Punch is used, then hopper 1 must be used for input. Hopper 2 must be used for output. When no separate cards are used hopper 1 may also be used for output.

6 The following I/O devices are supported by RJE,SNA.

IBM Magnetic Diskette Storage (3774, 3775, 3776)
IBM 2502 Card Reader (3774, 3775, 3776)
IBM 3501 Card Reader (3771, 3774, 3775, 3776)
IBM 3521 Card Punch (3771, 3774, 3775, 3776)
IBM 3784 Printer (3774)

POWER/VS does not control SDLC lines. The SNA terminals may be connected to VTAM and the NCP on any communication media supported by VTAM.

7 The 3790 support is limited to that comprising the 3790 RJE Facility.

Section 2: Method of Operation

This section describes in general the way in which POWER/VS operates. It outlines the logical structure of the POWER/VS SCP, presenting overviews of all internal operations and indicating the relationships between the various tasks and routines.

The following topics are discussed:

Code organization explains the POWER/VS code and storage structure and lists the internal macros.

Initialization and Termination gives an overview of the phases that handle start-up and shut-down of POWER/VS processing.

POWER/VS multitasking explains the principles of task selection, of starting a task, and of terminating a task.

Reader, execution processor, and writer tasks shows the data flow through the spooling process, and highlights the work done by the various phases related to these tasks.

Functions explains the functions involved in the spooling process, and specifically the queue and data file processing technique.

Services describes the routines of the nucleus phase.

Command processor gives an overview of command processing; how the command processor is invoked, and what actions are taken.

POWER/VS job accounting describes the account functions and the save account task.

Remote job entry highlights the essentials of RJE,BSC and RJE,SNA.

Appendages lists the routines in the nucleus phase that are extensions of the DOS/VS system control programs.

Code Organization

STORAGE STRUCTURE

The address space of the virtual POWER/VS partition is composed of three major areas, each containing an integral number of pages.

The permanent area consists of the first four pages of the POWER/VS virtual address area. These are fixed in real storage as soon as the POWER/VS system begins execution, and remain fixed till the system is terminated. They contain the nucleus phase, which consists of major control tables and routines which do not tolerate paging activity. See Figures 5.1 and 5.2.

The fixable area consists of the second group of pages within the POWER/VS virtual address area. These provide the necessary address space for dynamically-structured control areas and for data buffers used by the POWER/VS tasks. The size of this area depends on the amount of real address space that the user has assigned to the POWER/VS partition. The pages within the area are fixed when reserved for specific task use and freed when no longer required. At any point in time certain pages within the area will be fixed while others are free; the necessary page fixing and freeing is controlled by the storage management service of POWER/VS. See Figure 5.3.

The pageable area consists of the remaining pages within the POWER/VS virtual address area. These contain the remaining phases of the POWER/VS code and may be paged at any time the system requires additional real storage. The size of this area depends on the particular POWER/VS phases required for system execution; this in turn depends upon the execution options selected by the user (accounting, RJE, reader exit, cross-partition spool manager support, and SLI facility).

The page frames within real processor storage that are occupied by POWER/VS at any time are divided into two groups:

- The first group of page frames have been obtained from within the DOS/VS page pool and contain pages of code from the POWER/VS pageable area which are currently being referenced for instruction execution. Page frames within this group remain part of the page pool and are susceptible to system paging.
- The second group of page frames have been withdrawn from the DOS/VS page pool and contain, firstly, the pages of the POWER/VS permanent area and, secondly, those pages of the POWER/VS fixable area which have been fixed in real storage by the POWER/VS storage management service.

Note 1: The pages of the POWER/VS fixable area which have not been fixed in real storage by POWER/VS do not occupy real storage in any sense.

Note 2: For RJE,SNA operations or for 3800 Printer setup processing the DOS/VS GETVIS area in the POWER/VS partition is used to allocate work areas and control blocks. Refer to Figure 2.24F for a description of the RJE,SNA control blocks and work areas and to Section 5, "Layout of the Storage and Data Areas." The work areas are allocated by the SNA processors and freed when no longer needed. In case of 3800 Printer setup processing the work area is allocated by the SETPRT logic module and freed when no longer needed.

CODE STRUCTURE

The code of POWER/VS consists of External Routines, Internal Routines, Functions, Services, and Appendages.

External routines provide task support at the highest level of the system. Each external routine consists of a single phase which is physically located in the POWER/VS pageable area.

The following external routines are provided:

IPW$$ER	3540 Diskette Reader
IPW$$IB	RJE,SNA Inbound Processor
IPW$$MP	RJE,SNA Message Processor
IPW$$LF	RJE,SNA Logoff Processor
IPW$$LH	RJE,SNA Lgon Processor No. 1
IPW$$LN	RJE,SNA Logon Processor No. 2
IPW$$OB	RJE,SNA Outbound Processor
IPW$$OC	Outbound Compaction Manager
IPW$$PL	Physical List
IPW$$PP	Physical Punch
IPW$$PR	Physical Reader
IPW$$PS	Print Queue Status
IPW$$SA	Save Account
IPW$$SM	Internal Reader Spool Command Manager
IPW$$SN	RJE,SNA Manager
IPW$$TM	RJE,BSC Routines
IPW$$VE	VTAM Exit Routines

Internal routines provide task support at a level below external routines, which communicate with them by means of the Interface Macro Instructions to be described below. Each internal routine consists of a single phase which is physically located in the POWER/VS pageable area.

The following internal routines are provided:

IPW$$LR	Logical Reader
IPW$$LW	Logical Writer
IPW$$XR	Execution Reader
IPW$$XW	Execution Writer.

Functions provide support for operations common to two or more routines; they are to be regarded as high-level subroutines capable of concurrent execution, and are invoked by means of the Function Macro Instructions to be described below. Each function consists of a single phase which is physically located in the POWER/VS pageable area.

The following functions are provided:

IPW$$RQ	Reserve Queue Record
IPW$$AQ	Add Queue Set to chain
IPW$$NQ	Get Next Queue Set from chain
IPW$$DQ	Delete Queue Set from chain
IPW$$FQ	Free Queue Set storage
IPW$$PD	Put Data Record
IPW$$GD	Get Data Record
IPW$$SL	Get Source Statement Library Record
IPW$$PA	Put Account Record
IPW$$GA	Get Account Record
IPW$$IC	Invoke Command Processor
IPW$$SC	Scan Reader JECL statement
IPW$$XJ	Scan Execution JECL statement
IPW$$LU	Update LUB and PUB tables
IPW$$MD	Message definition
IPW$$MS	Message Handler
IPW$$OE	3540 Diskette Open
IPW$$OT	Tape Open.
IPW$$AS	Asynchronous service

Services provide support for operations common to many routines and functions; they are to be regarded as low-level subroutines capable of concurrent execution, and are invoked by means of the Service Macro Instructions to be described below. Each service is coded as a separate segment; all of these segments are however physically located within the nucleus phase (IPW$$NU) which forms the POWER/VS permanent area.

The following services are provided:

Task Management
Resource Management
Storage Management
Message Service
Disk and Tape Service
Timer Service
Validation Service.

Appendages provide code which, though physically present in the nucleus phase (IPW$$NU), is logically part of the DOS/VS supervisor or of some other DOS/VS component. Appendages may reference and update POWER/VS tables and data areas but may not invoke any POWER/VS routine, function, or service, and may not be invoked by them.

The following appendages are provided:

Page Fault Appendages
Attention Interface Appendage
RJE Channel End Appendage
Hot Reader Appendage
SVC 0 Appendage
SVC 90 and SVC 91 Appendage.

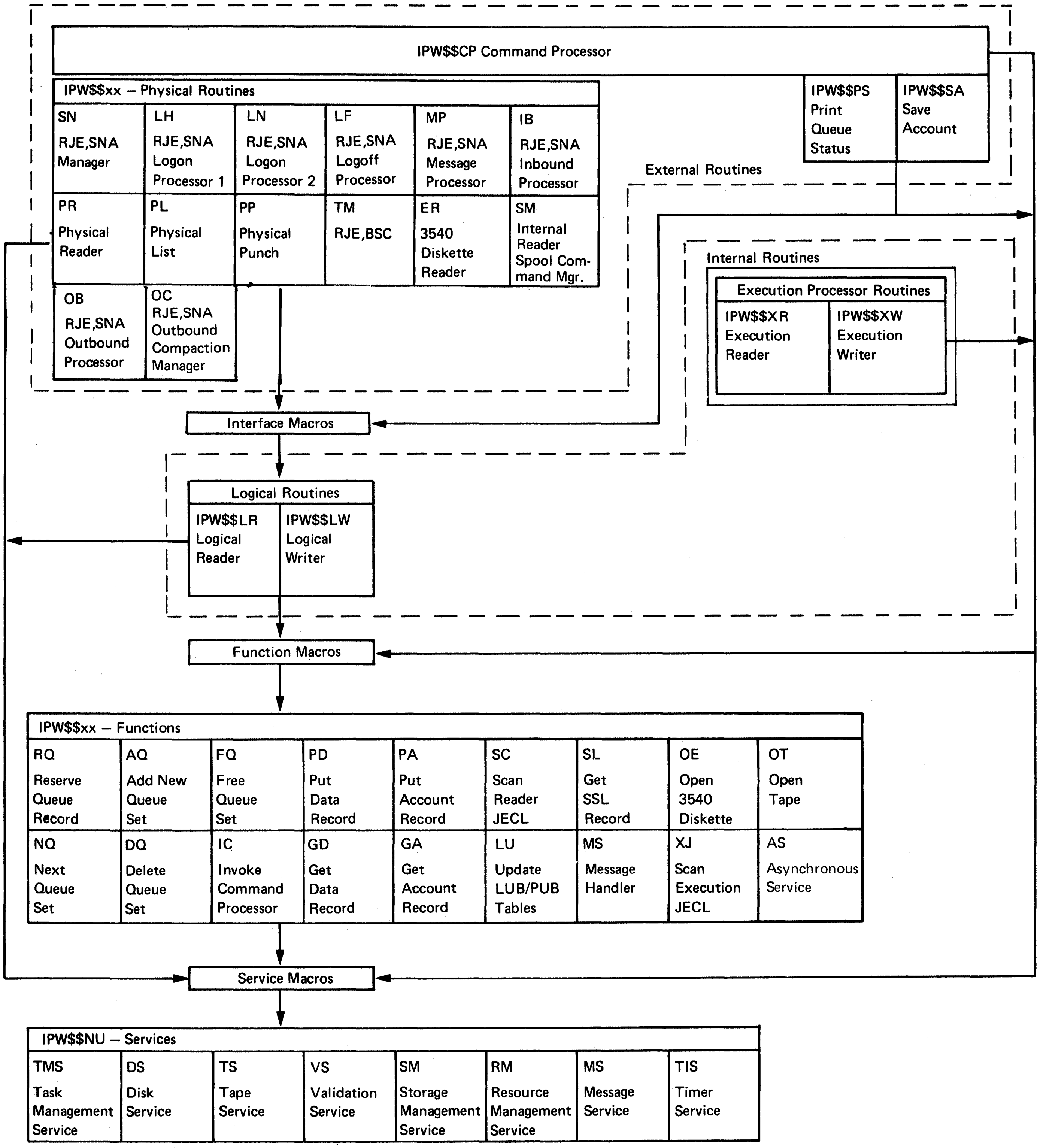


Figure 2.1. Hierarchic organization of calling sequence and interfaces between the POWER/VS routines

Note that appendages and initiator/terminators are not part of this hierarchy.

INTERNAL MACRO INSTRUCTIONS

Communication between external routines, internal routines, functions, and services is performed by means of POWER/VS internal macro instructions (see Figure 2.1). Macro instructions are also provided to define the format of common tables and data areas, and to perform other miscellaneous functions.

There are five types of POWER/VS internal macro instructions:

Interface Macros
Function Macros
Service Macros
Definition Macros
Miscellaneous Macros.

Interface Macros	Purpose
IPW$OLI	open logical interface
IPW$CLI	close logical interface
IPW$PLR	put logical record
IPW$GLR	get logical record

Function Macros	Purpose
Queue management	
IPW$RQS	reserve queue record
IPW$AQS	add queue set to chain
IPW$GQS	get next queue set from chain
IPW$DQS	delete queue set from chain
IPW$FQS	release queue set
Data management	
IPW$PDR	put data record
IPW$GDR	get data record
Account management	
IPW$OAF	open account file
IPW$CAF	close account file
IPW$PAR	put account record
IPW$GAR	get account record
Other functions	
IPW$IAS	invoke asynchronous service
IPW$ICP	invoke command processor
IPW$IOC	invoke outbound compaction manager (IPW$$OC)
IPW$SRJ	scan reader JECL statement
IPW$SXJ	scan execution JECL statement
IPW$ULP	update LUB and PUB tables
IPW$GSL	get source statement library record
IPW$RMS	request message service
IPW$OEF	open diskette file
IPW$OTP	open tape

Service Macros	Purpose
Task management (TM)	
IPW$ATT	attach new task
IPW$DET	detach current task
IPW$WFI	wait for initiation
IPW$WFO	wait for operator
IPW$WFL	wait for locked resource
IPW$WFM	wait for list posting
IPW$WFQ	wait for class table posting
IPW$WFC	wait for single posting
IPW$WFS	wait for storage posting
IPW$WFD	wait for dispatch
IPW$FCH	fetch transient (wait for PTA/LTA)
Resource management(RM)	
IPW$RSR	reserve resource
IPW$RLR	release resource
Storage management (SM)	
IPW$RSW	reserve work space
IPW$RLW	release work space
Message service (MS)	
IPW$WTO	write to operator
IPW$WTR	write to operator with reply
Disk service (DS)	
IPW$WTQ	write queue record
IPW$RDQ	read queue record
IPW$WTD	write data block
IPW$RDD	read data block
Tape service (TS)	
IPW$WTT	write tape record
IPW$RDT	read tape record
IPW$CTT	execute tape control
Timer service (TIS)	
IPW$RDC	read (TOD) clock
Validation service (VS)	
IPW$VDA	validate data area addresses

Definition Macros	Purpose
IPW$DAC	define account control block (ACB)
IPW$DAB	define asynchronous service anchor block (ASWS)
IPW$DBC	define buffer control area (BCA)
IPW$DCB	define command control block
IPW$DCO	define compaction control block (COCB)
IPW$DCT	define class table entry
IPW$DCW	define channel command word
IPW$DDE	define device entry
IPW$DDR	define data record format
IPW$DDV	define device table
IPW$DGN	define generation table (GNB)
IPW$DLC	define line control block (LCB)
IPW$DLR	define logon request control block (LRCB)
IPW$DLU	define logical unit control block (LUCB)
IPW$DMC	define module control block (MCB)
IPW$DMD	define message definition module
IPW$DMM	define message control block (MSCB)
IPW$DMS	define RJE (BSC and SNA) message control block (MMB)
IPW$DPA	define permanent area (CAT)
IPW$DPC	define page control block (PCB)
IPW$DPD	define partition control block (PDB)
IPW$DPW	define physical work space (PWS)
IPW$DRM	define SNA remote control block (RMCB)
IPW$DQC	define disk management block (DMB)
IPW$UQR	define queue record (QRA)
IPW$DSC	define storage control block (SCB)
IPW$DSL	define SLI work space (SLW)
IPW$DSR	define service request block (SRB)
IPW$DSU	define SNA unit control block (SUCB)
IPW$DSV	define save area
IPW$DTB	define tape control block (TBB)
IPW$DTC	define task control block (TCB)
IPW$DTE	define task control block extension area
IPW$DWA	define SNA work area (WACB)

Miscellaneous Macros	Purpose
Entry exit	
IPW$SAV	save caller registers
IPW$RET	restore registers and return to caller
IPW$ALN	align to storage boundary
IPW$EQU	establish equates

Initialization and Termination

INITIALIZATION OF POWER/VS

The initiation of POWER/VS comprises three phases:

- User-generated phase (POWER/username)
- IPW$$I1
- IPW$$I2.

Job control (EXEC statement processor) fetches the first of these three phases, which contains a small loader routine and a generation table. These are assembled from the generation macros POWER, PLINE (optional) and PRMT (optional). There can be as many of these generation table phases in the core image library as there are different versions of POWER/VS needed by user.

The loader routine in front of the generation table loads phase IPW$$I1 behind the generation table and gives control to it (see Figure 2.3).

IPW$$I1 checks the environment in which POWER/VS has to be executed to ensure that the following conditions are satisfied:

- Only one POWER/VS program active at one time.
- SYSLOG must be assigned to a console device.
- POWER/VS must be a main task.
- Run mode must be virtual.
- DOS/VS supervisor must have JAI support if POWER/VS supports job accounting.
- DOS/VS supervisor must have POWER/VS support.
- Remote terminal ID's are validated for RJE,SNA.
- Size of GETVIS area is checked for RJE,SNA.
- Line addresses are validated for RJE,BSC.
- Size of pageable area is checked.
- Real storage allocation must be at least 10K; if more than 128K, then 128K is forced.

If IPW$$I1 finds the environment to be satisfactory, it loads the next initiator phase, IPW$$I2, into the pageable area and gives control to it (see Figure 2.3).

IPW$$I2 is the main initiator phase that actually loads and starts up POWER/VS:

- Saves the generation table in itself.
- Loads the POWER/VS nucleus (8K permanent area: PFIX 4 pages).
- Initializes control blocks in nucleus and relocates various addresses. Generation table information is moved to various control blocks.
- Loads all required pageable phases behind itself in the pageable area (for required IPW$$xx phases, refer to "Programming Requirements" in Section 1) using a Local Directory List.
- Calls $$BPOWIN to change PSW key to zero.

- Initiates a TCB chain, to enable itself to run on as a task and make use of functions and services provided by the nucleus. This TCB chain is also used as a base for the task selection mechanism to attach more TCBs (tasks).

  The spool manager master TCB will be at the end of the chain when the POWER/VS cross-partition communication interface option is specified in the POWER generation macro.

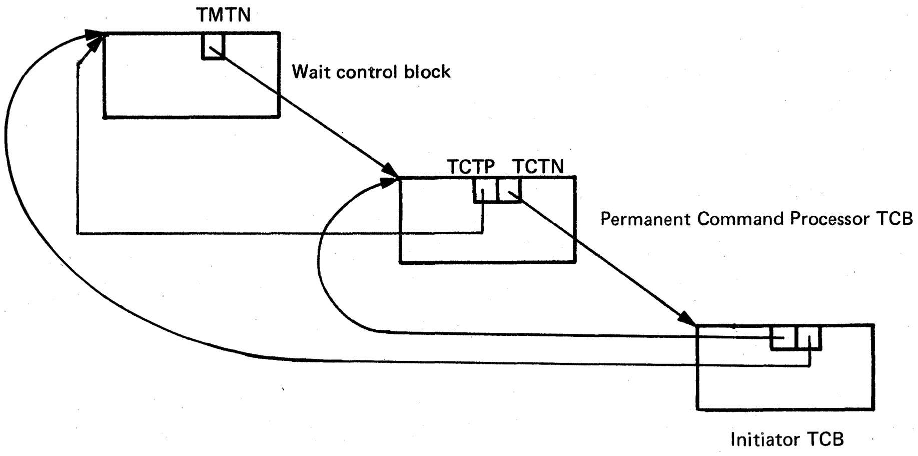
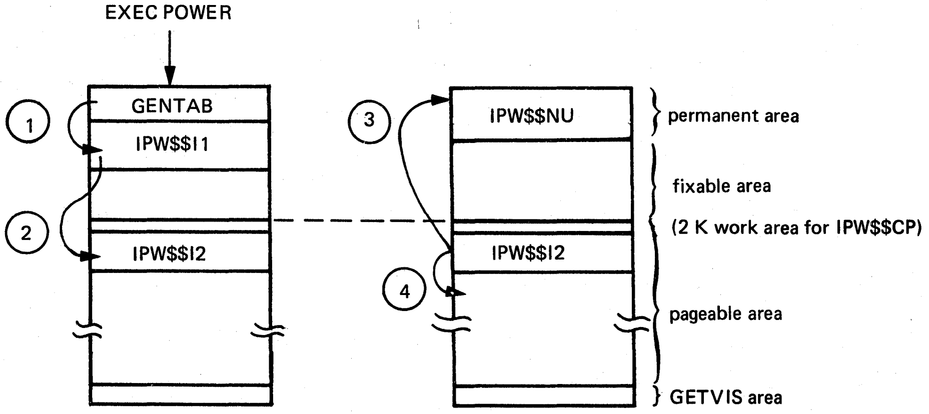


Figure 2.2. Initial Task Selection (TCB chain)

- If SYSIPT is assigned to a unit record device, Autostart control cards are read from it. In this case IPW$$I2 acts as a physical reader and links to IPW$$LR to write the data to the spooling device. Then it uses IPW$$LW to retrieve this data for processing of the statements.
- Opens POWER/VS direct access files using DTFPH.
- Processes FORMAT= statement (optional formatting of the POWER/VS direct access files).
- At the end of initialization, a status report is printed (if SYSLST is assigned), an "initialization complete" message is issued, and SYSRDR, SYSPCH, SYSLST, SYSIPT are unassigned in the POWER/VS partition. The initiation TCB is posted to be inactive and control is given to POWER/VS task selection.



[1] GENTAB loads IPW$$I1
[2] IPW$$I1 loads IPW$$I2 (start of pageable area)
[3] IPW$$I2 loads IPW$$NU (start of virtual partition)
[4] IPW$$I2 lcads other POWER/VS code (see Figure 1.7).

Figure 2.3. Initiation Logic

TERMINATION OF POWER/VS

POWER/VS is normally terminated by the PEND command. All active tasks are allowed to continue until they finish processing the current queue entry. Deactivation is handled by each of the tasks, after the command processor (IPW$$CP) has set a termination code ("S", "E", or "F") in their TCBs. In case of an I/O error, POWER/VS can also be terminated by the IPW$$TR phase (see "Abnormal Termination of POWER/VS").

After all supported partitions have been released, the POWER/VS partition is restored for normal DOS/VS operation.

The detach routine of task management actually gives control to the terminator routine within IPW$$I2, which executes as a subtask.

The terminator routine performs the following functions:

- It closes POWER/VS files (writes master record back to queue file, and writes end-of-file record to account file).
- Optionally prints status report, especially for the queue file, by passing an internal PDISPLAY card to invoke P status task (if SYSLST assigned to a printer).
- Issues termination message.
- Issues EOJ macro which also PFREEs the permanent area and all other fixed pages.

The DOS/VS EOJ processor $$BEOJ4 then gives control to $$BPOWIN, which performs the last terminating action:

- Restores POWER/VS partition for normal use.

Abnormal Termination of POWER/VS

The termination routine IPW$$TR is entered from task selection C state processing in case of an error at completion of any I/O operation. It executes under the TCB of the failing task. The failing task is canceled.

The following specific failures necessitates POWER/VS termination:

- failure of the permanent command processor task
- logic error of POWER/VS function encountered
- failure during initiation of POWER/VS
- I/O error while IPW$$TR is trying to recover
- I/O error on the master record
- I/O error while obtaining a record from the free queue set in the queue file (IJQFILE).

POWER/VS can also abnormally be terminated by the PEND KILL command. As a result of this command a dump will be printed by the Command Processor (optional), after which an EOJ macro is issued.

POWER/VS MULTITASKING

In order to execute POWER/VS tasks concurrently, but asynchronously, POWER/VS incorporates multitasking support. Because this support does not depend on the multitasking (asynchronous processing) support provided in DOS/VS, it is called 'private multitasking'.

Each POWER/VS task is equipped with a task control block (TCB) created in fixed storage. The TCB is used to establish the identity of the task and to preserve its status when it is not in active control of the central processor.

The task control blocks present at any time in POWER/VS are linked together by means of 'next task' and 'previous task' pointers to form a logical list called the 'task selection list'. The task selection list is considered to begin and end with the Wait Control Block (WCB), a skeleton TCB whose function is to delimit the task selection list.

The logical position of each task control block within the task selection list (see Figure 2.4) determines its dispatching priority relative to the other tasks within the list. This priority takes effect only when task selection is entered; once a task is running it will continue to run until it yields control by means of one of the task selection service macro instructions (IPW$WFx) or sustains a page fault. Thus, a higher priority task will not interrupt a running task.

An initial task selection list is constructed by the POWER/VS initiator (IPW$$I2). This list contains the wait control block, the task control block of the initiator task, and the task control block of the permanent command processor task. All further additions to and deletions from the task selection list are performed by the task management service.

POWER/VS provides three components of task management service:

- Task initiation - attach new task
- Task selection - select next task for dispatch
- Task termination - detach current task

Each of these components will now be discussed in greater detail.

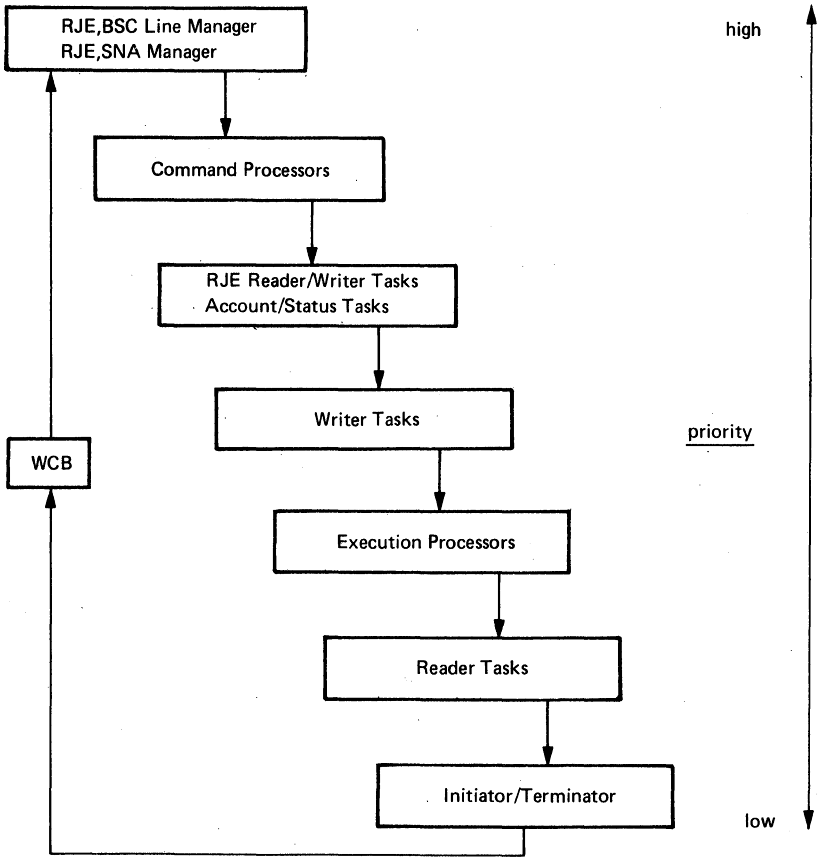


Figure 2.4. Task Selection List (TSL)

TASK INITIATION

Task initiation is entered from a POWER/VS task by means of the IPW$ATT (attach new task) macro instruction. The issuing task has already acquired storage for and formatted the task control block which will represent the new task; in particular it has created the task storage descriptor which establishes the task type and identity.

Task initiation determines the point within the current task selection list at which the new task control block must be inserted, and adjusts the 'previous task' and 'next task' pointers within the task control blocks concerned. The new task is then set into D(dispatchable) state, and return is made to the calling task.

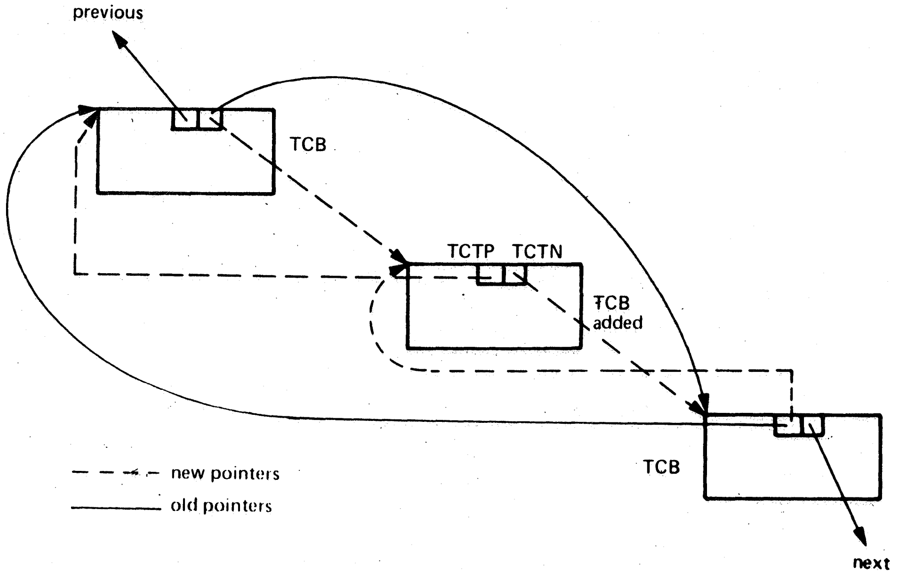


Figure 2.5. Attaching a Task

TASK SELECTION

Task selection is entered from a POWER/VS task when that task yields control to the central processor by means of one of the IPW$WFx (wait for 'x') macro instructions listed below. In each case 'x' represents the task state value to be associated with the task yielding control.

IPW$WFI - set I state and wait for initiation.
IPW$WFO - set O state and wait for operator response.
IPW$WFL - set L state and wait for locked resource.
IPW$WFM - set M state and wait for multiple control block posting.
IPW$WFQ - set Q state and wait for class table posting or multiple XECB posting.
IPW$WFC - set C state and wait for ECB or CCB posting.
IPW$WFS - set S state and wait for ECB or CCB posting.

IPW$WFD - set D state and wait for re-dispatch.

(The significance of these individual states will emerge in the discussion of the routines that issue the individual macro instructions.)

The status of the task yielding control is saved by storing the current contents of the general purpose registers (and the condition code) in the task register save area of the task control block. This done, the task selection process can begin.

The task selection list is used to address and examine each task control block in turn in order of dispatching priority to determine whether the associated task can be dispatched. This is done by means of the task state value set in the task control block. (In addition to the task states listed above, two further states must be mentioned: P state (page-bound), which is set by the page fault appendage (see "POWER/VS Appendages") when a task sustains a page fault, and F state (fetch-bound), which is set when a task requires to fetch a transient routine but finds that the necessary transient area is in use.)

Tasks in the following states are non-dispatchable:

I state - the task is waiting for reactivation.
P state - the task is waiting for a page-in operation.
O state - the task is waiting for operator response.

Tasks in the following states are conditionally dispatchable. A further test or tests must be performed to determine whether the condition has been satisfied and the task is in fact ready for dispatch.

L state - the task is waiting for a locked resource.
F state - the task is waiting for access to a transient area.
S state - the task is waiting for ECB or CCB posting.
C state - the task is waiting for ECB or CCB posting.
Q state - the task is waiting for class table posting or multiple XECB posting.
M state - the task is waiting for any of a set of ECB or CCB postings.

Tasks in the following state are unconditionally dispatchable:

D state - the task is ready for immediate dispatch.

As soon as a dispatchable task is found within the task selection list, the general purpose registers (and condition code) are restored from the task register save area of the task control block, the task is set into R state (running), and execution of the task is resumed from the point at which it previously ceased.

If the entire task selection list is scanned without any task being found to be dispatchable, the task selection service issues an SVC 7 to pass control to the DOS/VS supervisor. POWER/VS will wait till the occurrence of some related event (I/O completion, for example) causes DOS/VS to return control to the task selection service. The entire task selection process is then repeated.

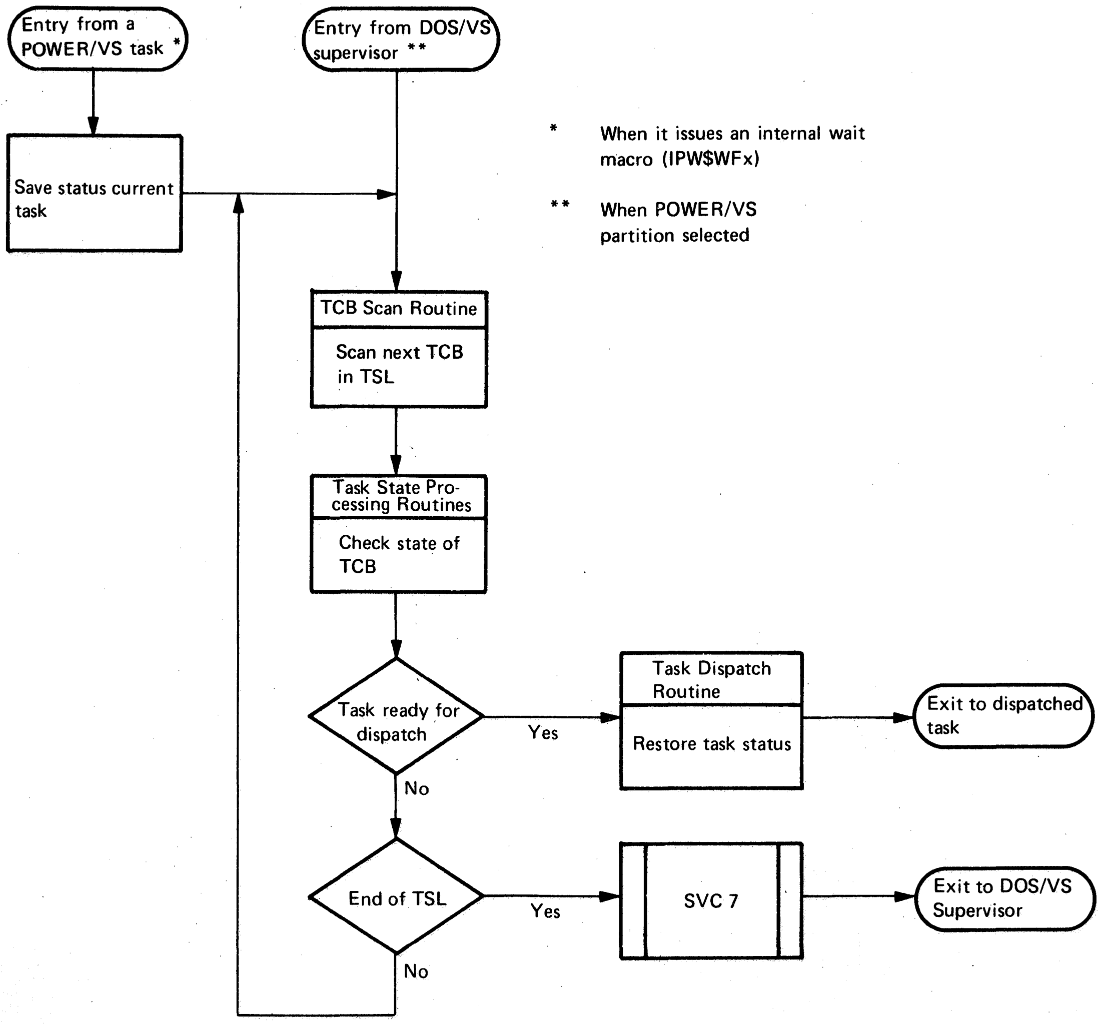

Entry from a POWER/VS task *
Entry from DOS/VS supervisor **
Save status current task
* When it issues an internal wait macro (IPW$WFx)
** When POWER/VS partition selected
TCB Scan Routine
Scan next TCB in TSL
Task State Processing Routines
Check state of TCB
Task ready for dispatch
Yes
No
Task Dispatch Routine
Restore task status
Exit to dispatched task
End of TSL
Yes
No
SVC 7
Exit to DOS/VS Supervisor

Figure 2.6. Overview of Task Selection

TASK TERMINATION

Task termination is entered from a POWER/VS task by means of the IPW$DET (detach current task) macro instruction.

Task termination removes the task control block of the current task from the task selection list by adjusting the 'previous task' and 'next task' pointers within the neighbouring task control blocks within the list. The storage occupied by the eliminated task control block is returned to the system, and control is then passed to task selection to determine the task next to be dispatched.

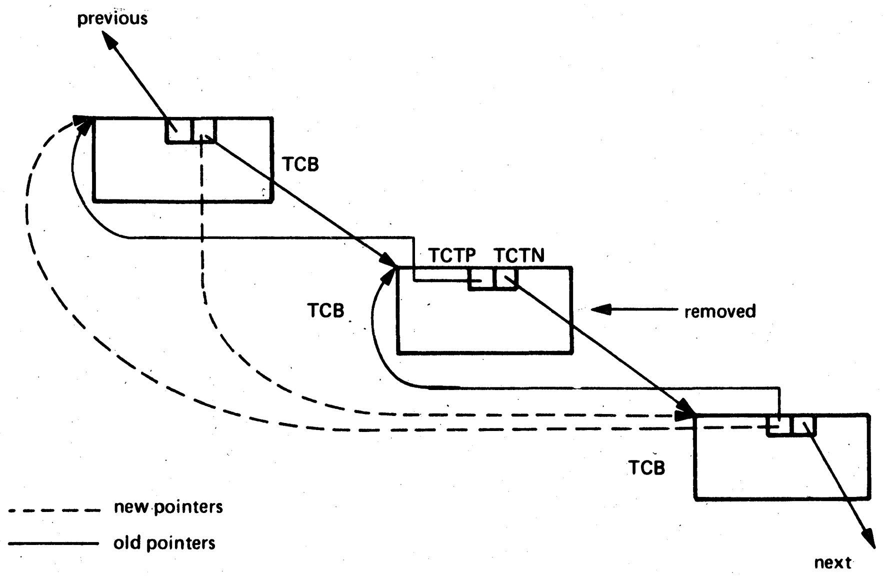


Figure 2.7. Detaching a Task

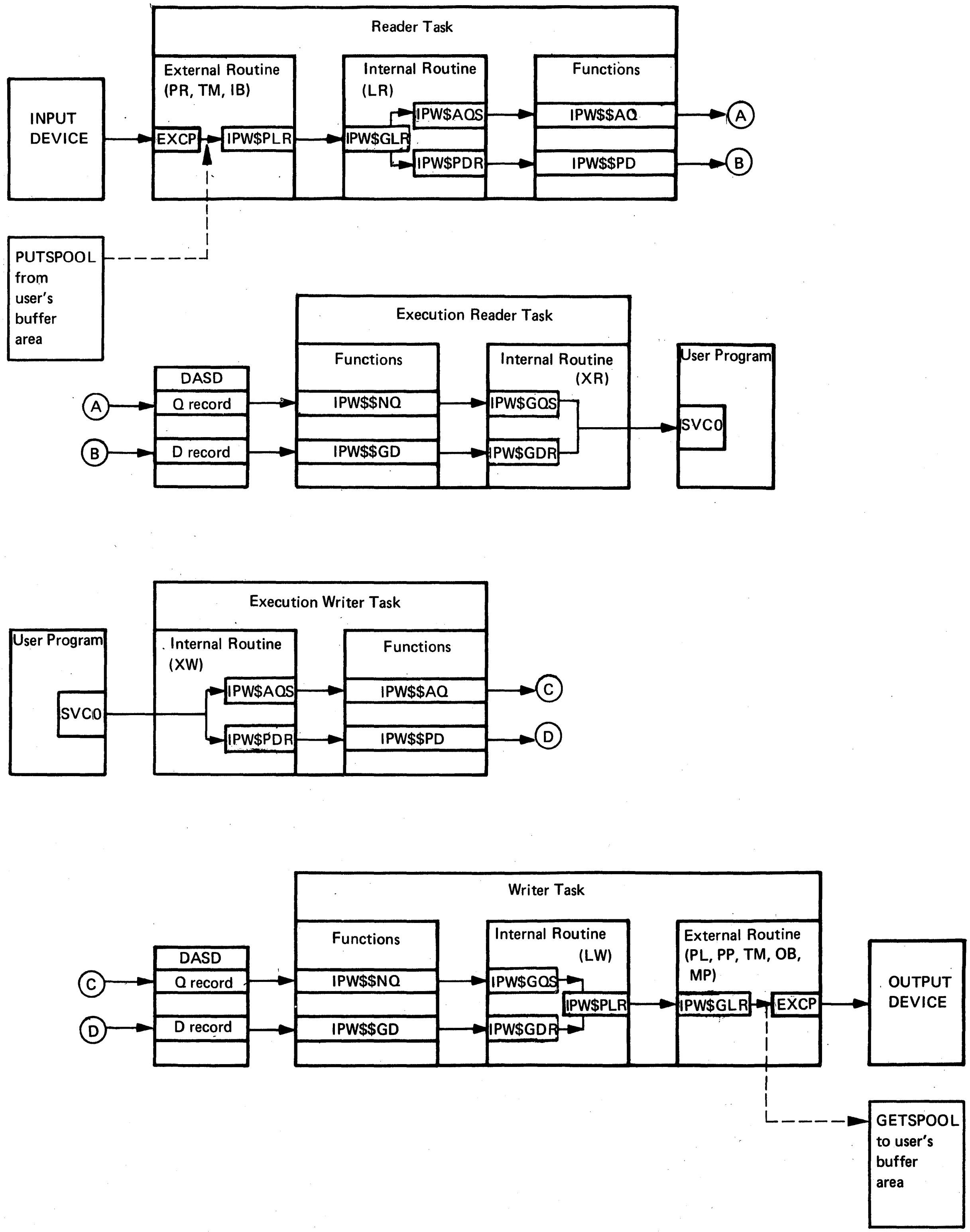


Figure 2.8. Data flow throughout the spooling process

READER TASKS

The reader task is executed by a physical reader routine (PR) and logical reader routine (LR). These routines pass control to each other through a logical record interface. At unit exception, the task places itself in a dormant state, releasing as much work space as possible.

Hot reader support enables a dormant task to continue without a PSTART command, if new input has become available (refer to "Device End Appendage").

Physical Reader (PR)

The PR is entered when a reader task is invoked by a PSTART command, or when an unsolicited device-end interrupt occurs while the task is in a dormant state (hot reader support). Special work areas will be allocated at entry time and initialized according to the supported physical device. The work areas can be released by the termination routine IPW$$TR.

The PR performs the physical input for one or more devices and establishes the linkage with the LR so that, on request, each logical record can be passed over the interface to the LR.

Each input operation will handle a number of records by means of command chained CCWs (refer to "Physical Data Record area" in Section 5). The input operation is performed with real addresses in the CCWs (EXCP real).

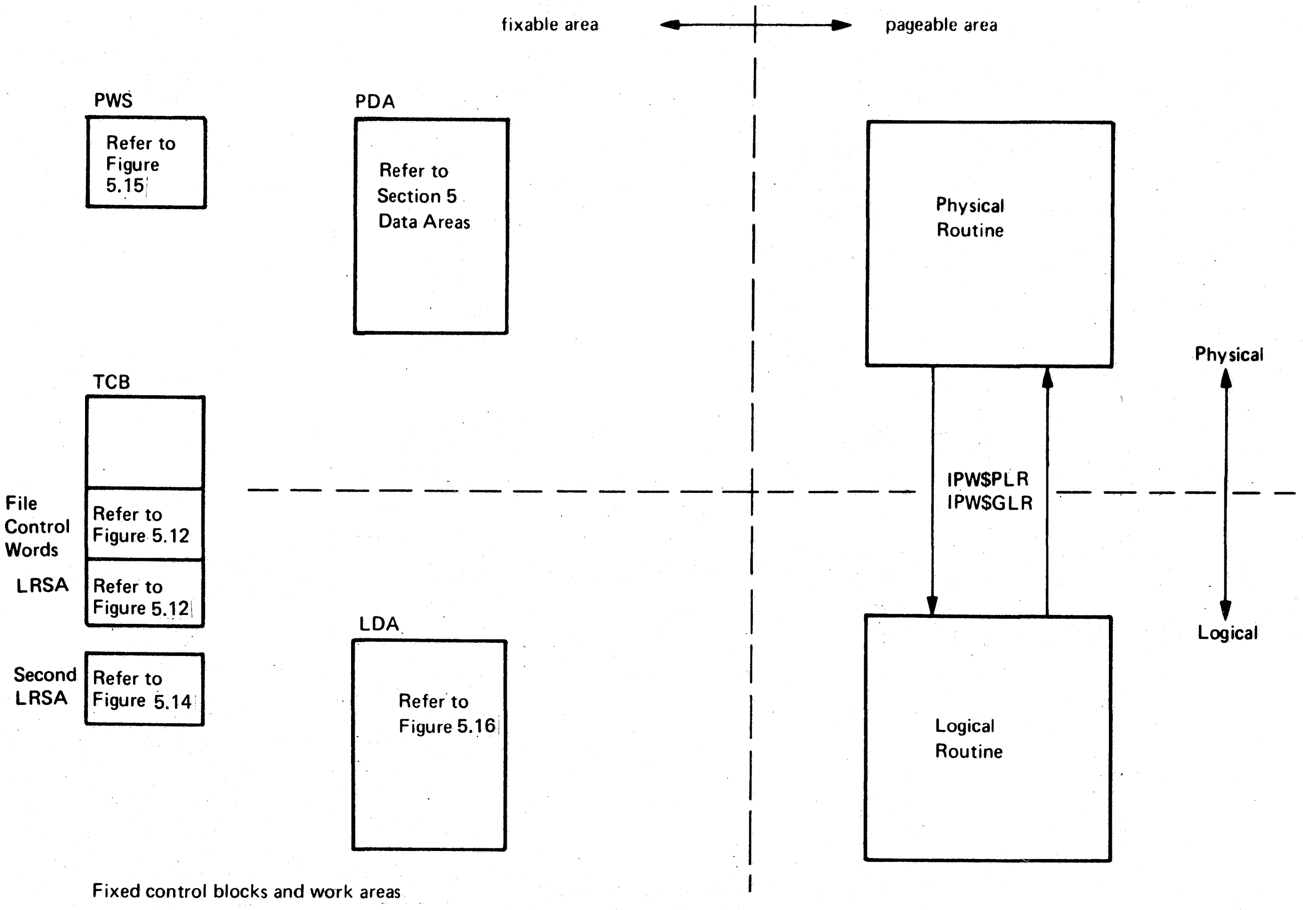


Figure 2.9. Physical and Logical Work Areas

Physical 3540 Diskette Reader (ER)

This routine is entered via the logical reader when a RDR statement is encountered in the input stream, or via task selection as a result of a PSTART command issued for the diskette reader only. It reads data from the physical diskette reader associated with the reader task.

Logical Reader (LR)

The first time the routine is entered, it reserves work space for the queue record area and acquires a queue record from the free queue set (via IPW$RQS macro instruction).

The values may be overwritten by specifications in the JECL statements (* $$ JOB and * $$ CTL). Records passed via the logical record interface will be passed in turn to the put data function routine (PD) for writing to the data file. The general purpose byte in the record request word (RRW) indicates what action is to be taken by the PD routine.

General purpose byte posted by LR:

Data break	for // EXEC, /* records (used for performance reasons)
End of data	for last record for this job entry
End of block	in case of unexpected end of input (expected delimiter not encountered, or last record of block).

The routine provides a user exit. It enables a user-written routine to examine each JCL and JECL statement and delete or insert records in the job stream.

If the last record for the currently processed job entry is passed, the IPW$AQS macro instruction is issued (to invoke the add queue function) to add a queue record on the queue file.

EXECUTION PROCESSOR TASKS

Each serviced DOS/VS partition is equipped with a partition control block (see Figure 5.21). A partition control block is created within the POWER/VS fixable area whenever a partition is brought under the control of POWER/VS by means of a PSTART command.

The partition control block contains header information pertinent to the partition itself and a set of one or more device entries. Each device entry relates to a single real or dummy physical device specified by the operator at the time at which the partition control block is created and thereafter defines that device to POWER/VS.

The first device entry within each partition control block describes the reader device for that partition. If the partition is a writer-only partition the device described by the reader entry is the system console device. Further device entries describe the list devices and punch devices for the partition.

Each device entry is used to pass information from the user partition to the POWER/VS execution processor task which is responsible for the emulation of that device.

The execution processor tasks are:

- Execution read task (XR and XJ)
- Execution writer tasks (XW).

An execution read task is started for each partition at the time at which the partition is brought under POWER/VS control. It continues to run until the partition is returned to DOS/VS control by means of a PSTOP command.

The execution read task is responsible for servicing all read requests addressed by the user program to the partition read device designated at partition PSTART time. It is additionally responsible for recognizing the first request addressed by each job executed within the partition to each of the partition list and punch devices designated at partition PSTART time, and initiating an execution writer task to service the further program requests addressed to that device.

Until end of job the execution tasks proceed concurrently but asynchronously. When the execution reader detects an end-of-job condition it posts a stop condition to each of the subordinate tasks that it started. It then waits until each of these tasks detaches itself in turn.

If no other queue entry can be processed the reader task will place itself in a wait state, after a message is issued. When a PSTOP command is issued the reader task and its subordinate tasks will eventually be detached after processing the current queue entry.

Execution Reader Routines (IPW$$XR and IPW$$XJ)

This routine will emulate the user channel program input requests for the reader device. To service these requests a data record is kept available throughout the process of this routine. Records are retrieved via IPW$GQS and IPW$GDR macro instructions. The routine does the following:

- Intercepts first request for output of the user channel program, and then attaches a subordinate writer task.
- Handles all input requests from the user channel program.
- In case of a writer-only partition, analyzes JECL statements from a console read operation and starts a writer task.
- Indicates termination of a writer task once a queue entry has been processed or a data break condition is recognized.
- When an SLI JECL statement is encountered, initiates for the insertion of source statement library records.
- When PUN, LST, or PRT JECL statements are recognized, terminates the appropriate writer task, builds a new queue record, and starts the writer task again.

Execution Writer Routine (IPW$$XW)

At entry of the execution writer routine, the execution read task reserved queue space and initialized (from * $$ LST or * $$ PUN) the queue record area.

Space is reserved for the data buffer for the output records. If a request from the user program is found in the task list entry of the partition control block, the user channel program is emulated. If no entry is found the task enters a wait state for further user program requests.

Each CCW is checked for validity and user data is transferred to the data file by invoking the put data record function.

At termination of the task, which is controlled by the execution reader task (stop code), a dummy end of data record is passed to the put data routine, and the current queue set is added to the appropriate class chain by invoking the add queue record function. The data buffer is released and the task detaches itself.

Output segmentation is driven by count (as specified in JECL) or by the user program (via an FCB buffer load or by issuing a SETPRT or SEGMENT macro) and is established through formation of a new queue set. The former queue set is added to the appropriate class chain.

Any request to alter the printer setup, either via a // SETPRT statement or a SETPRT macro instruction, is routed to SETPRT. When SETPRT determines that the device is being trapped by POWER/VS, it passes the SETPRT parameter list to POWER/VS after a basic validation. This is done by issuing an I/O to the device with an 'FD' channel command operation code, and with the data area address pointing to the SETPRT parameter list. The execution processor recognizes the 'FD' operation code as a valid command for the 3800.

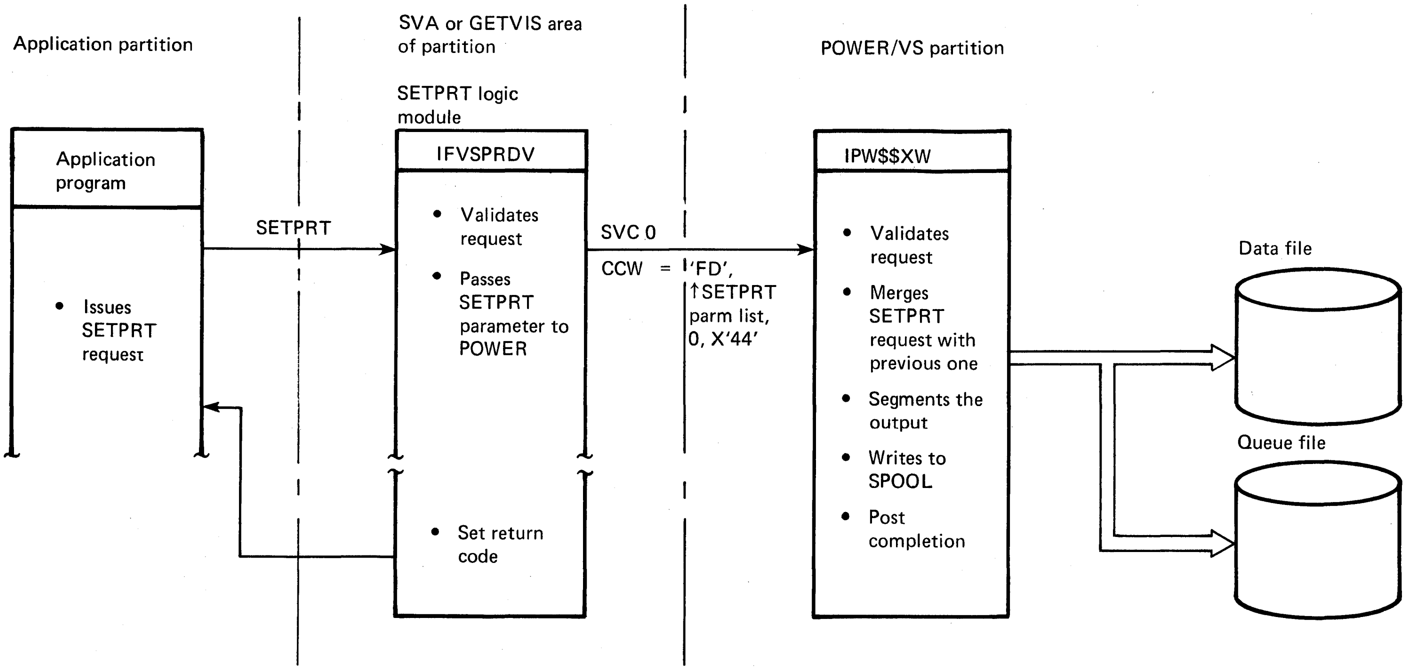
The execution processor maintains a control block, called the TCB extension area, which contains the current printer setup of the device being spooled. When a SETPRT parameter list is encountered by the execution processor, the TCB extension area is updated, which means the new setup request is merged with the previous one.

When the BURST, FORMS, FLASH, or copy group specifications have been changed, the output is segmented (that is, the output entry is closed and added to the class chain according to the priority; then a new output entry is created with the same jobname and job attributes but with a different job number, in order to facilitate queue manipulation by the operator). The SETPRT parameter list is then written as the first record in the new list queue entry.

Whenever the execution processor detects that a valid CINDX value (other than 0 or 1) was specified in the SETPRT parameter list, it assumes that the user will manage the copy group handling by himself.

The execution processor creates a new output LIST entry with the same job attributes and sets the transmission count to one.

When a SETPRT parameter list contains an FCB specification, the FCB image is loaded from the core image library and the internal representation of the page format is updated. The FCB phase name is saved in the TCB. The FCB image is validated for accuracy. If a 3800 FCB image is invalid, a message (IQ54I) is written to the operator and the hardware default FCB is used. The LTAB specification is assumed as the internal representation of the FCB.



The following 3800 CCW operation codes are not accepted by POWER/VS (execution writer) and cause the channel program check and the unrecoverable I/O error flags in the CCB to be posted:

- Load translate table (X'83')
- Load character module WCGM (X'53')
- Load forms overlay sequence control (X'43')
- Load copy number (X'23')
- Load graphic character modification (X'25')
- Load copy modification (X'35')

If the user is prepared to accept unrecoverable errors, control is returned to him in the normal way; otherwise, a cancel flag is set in the PIB of the user partition to force cancellation of this job.

The following 3800 CCW operation code is ignored:

- clear printer (X'87')

WRITER TASKS (LIST AND PUNCH)

The writer task is executed by a physical routine (PL or PP) and a logical routine (LW). These routines pass control to each other through a logical record interface. If no next job is available, the task places itself in a dormant state, releasing as much work space as possible.

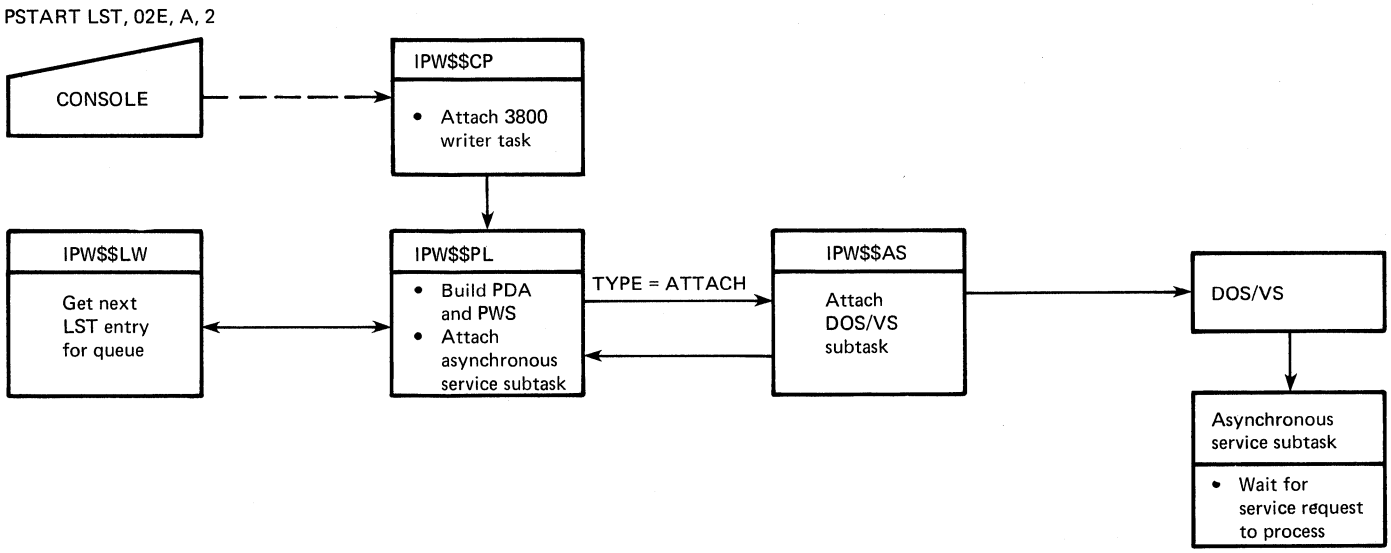
Physical List and Punch

These routines are entered when a list or punch task is invoked by a PSTART command. At entry, special work areas are allocated and initialized according to the supported physical device (see Figure 2.9). The work areas can be released, by the termination routine IPW$$TR. Both routines perform physical output. On request, the linkage allows logical records to be received in turn over the interface from the logical writer routine.

When a 3800 Printer list task has been started, a request is made to asynchronous service to attach a DOS/VS subtask. The subtask then is responsible for doing all SETPRT requests. The list writer task may be stopped during initialization when:

- No DOS/VS subtask available
- No GETVIS storage for SETPRT workspace

3800 Printer writer task initialization



PSTART LST, 02E, A, 2
CONSOLE
IPW$$CP
• Attach 3800 writer task
IPW$$LW
Get next LST entry for queue
IPW$$PL
• Build PDA and PWS
• Attach asynchronous service subtask
TYPE = ATTACH
IPW$$AS
Attach DOS/VS subtask
DOS/VS
Asynchronous service subtask
• Wait for service request to process

Each output operation will print or punch a number of records by means of command-chained CCWs (see Figure 5.15). The output operation is performed with real addresses in the CCWs (EXCP real).

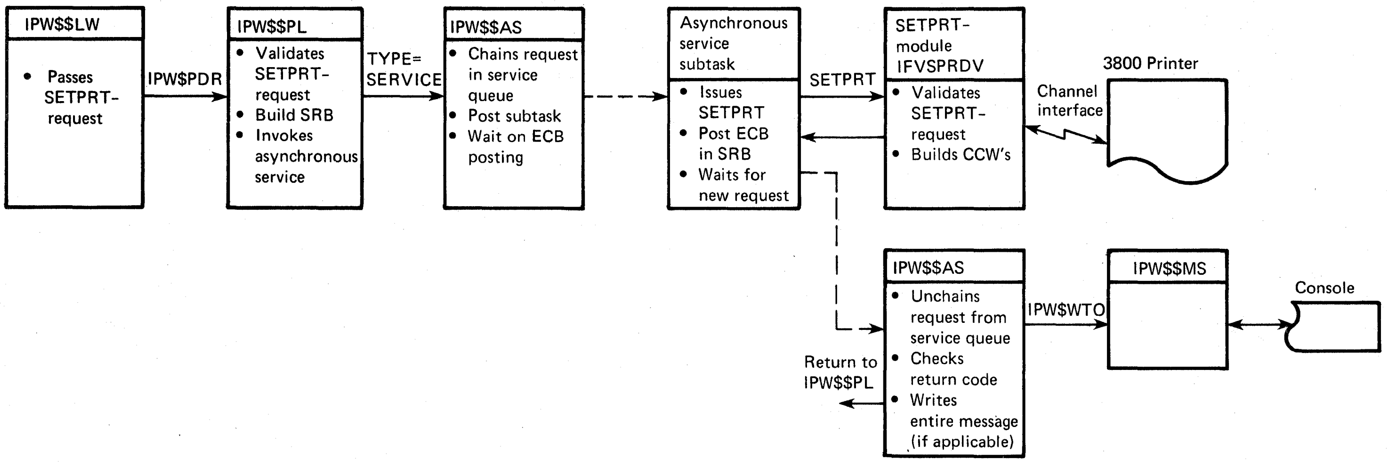
Logical Writer

A new queue set is addressed by invoking the get next queue set (NQ) function. A logical record is retrieved from the data file by invoking the get data record (GD) function.

The logical record is passed over the interface to the physical routine. The general purpose byte is tested for following action to be taken:

- Normal record: retrieve the following logical record.
- End of data record: delete the queue set by invoking the delete queue set function and address the next queue set to be processed.

SETPRT-request processing flow



Functions

This chapter describes only the queue and data management functions because they require an explanation of the organization of their respective files.

QUEUE FILE ORGANIZATION

Three types of records are physically present on the queue file:

a master record	queue identifier M (physical first in file)
queue records	queue identifier F, R, L, or P
a dummy record	queue identifier D (physical last in file).

Logically, by means of pointers, the queue records are either a member of the free queue set or a member of a queue set that is in turn a member of a class chain.

Records in the free queue set (queue identifier F) are chained by the next-in-set pointer. A next-in-set pointer with value zero delimits the chain. The start of this chain is kept in the master record.

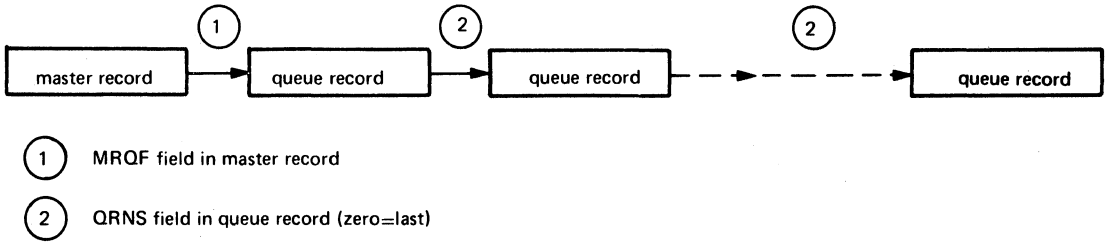

1
2
2
master record
queue record
queue record
queue record
1 MRQF field in master record
2 QRNS field in queue record (zero=last)

Figure 2.10. Free Queue Set

Each record in the free queue set has a pointer to a unique track group on the data file available for use.

For each queue entry of a class chain, there is a first-in-set queue record, which provides forward and backward pointers in the class chain. If more than one track group of the data file is required for a queue entry, additional queue records are chained to the first-in-set queue record by the next-in-set pointer. Each of these additional records points back to its first-in-set queue record.

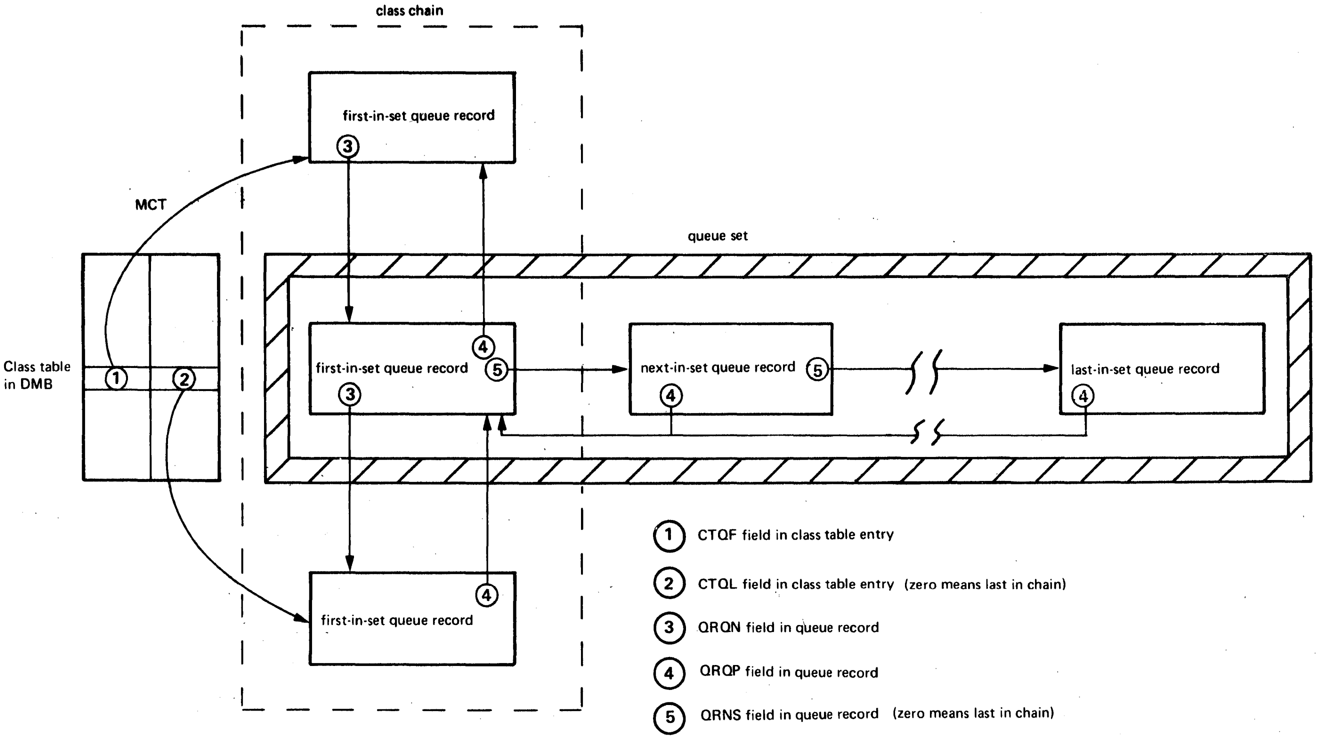

class chain
first-in-set queue record
3
MCT
queue set
Class table in DMB
1
2
first-in-set queue record
4
5
3
next-in-set queue record
5
4
last-in-set queue record
4
first-in-set queue record
4
1 CTQF field in class table entry
2 CTQL field in class table entry (zero means last in chain)
3 QRQN field in queue record
4 QRQP field in queue record
5 QRNS field in queue record (zero means last in chain)

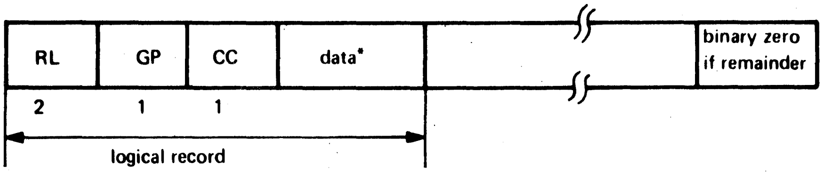
Figure 2.11. Class Chain and Queue Set

The master record and the queue record currently processed by a task are contained in storage. The master record in the Disk Managerent Block (DMB) and the queue record in the fixable area.

DATA FILE ORGANIZATION

The space available on the data file is arranged in track groups, each containing an integral number of tracks. Each track group has a fixed relationship with a queue record in the queue file. Hence, a queue record in the free queue set will point to an available track group. A queue record in a class chain will point to a track group that is in use.

Each physical record (fixed length) in the data file contains one or more logical records. Each logical record represents a unique record of the user program that is being spooled. The following figure shows the layout of a physical record:



RL - length of the logical record
GP - general purpose byte (see also section "Logical Reader")
CC - command code associated with the user channel program
* trailing blanks are suppressed

Figure 2.12. Data Record

QUEUE FILE AND DATA FILE PROCESSING

The queue and data file are maintained by the queue and data function routines:

- Reserve queue record (IPW$$RQ)
- Add queue set (IPW$$AQ)
- Delete queue set (IPW$$DQ)
- Free queue set (IPW$$FQ)
- Put data record (IPW$$PD)

Retrieval on the queue and data files is performed by the function routines:

- Get next queue set (IPW$$NQ)
- Get data record (IPW$$GD)

The reserve queue record function obtains the first record from the free queue set and updates the pointer in the master record to the next record in this set. If the queue file is exhausted, that is, no free queue record is available, the task is placed in a wait state until queue records in use are returned to the free queue set.

The add queue set function inserts, by means of its next and previous pointers a new queue entry in the appropriate class chain.

The delete queue set function removes a queue entry, which is no longer required, from its class chain. The free queue set function returns the queue records to the free queue set.

The put data record function moves a logical record into the output area for a physical record on the data file. If the output remainder is not large enough to contain this logical record, the output area is written to the data file as a physical record. If the track group is exhausted, a new queue record is obtained from the free queue set (explicitly coded without invoking a function).

The get next queue set function obtains the first queue record of a new queue set from a class chain referenced by the calling task. Class chains relating to the calling task are identified by a task class list in the TCB (see Figure 5.10). Each entry in the task class list is examined in turn. If the class chain it addresses is not empty, the queue sets which it contains are examined in turn until a set is found to be dispatchable. A positioning error occurs for the failure of a GETSPOOL request.

If all addressed class chains are empty or contain non-dispatchable queue sets, the calling task is placed in a wait state until a new queue set is added or an existing queue set becomes dispatchable. For a GETSPOOL request, the queue records are scanned for a matching job name and class with the one supplied in the GETSPOOL SPL.

The get data record function provides the calling routine with a logical record by means of the record request word. If the input area is exhausted a new physical record will be read from the data file. If the track group is exhausted the next queue record in the set will be obtained to address the next track group on the data file for the queue entry that is in process.

ASYNCHRONOUS SERVICE

The asynchronous service function of POWER/VS handles all SETPRT requests for the following reasons:

- SETPRT processing may use an extended time.
- Page faults in the SETPRT module would be otherwise handled by POWER/VS.
- Since the SETPRT module does not follow the POWER/VS register convention, page faults cannot be handled by POWER/VS.

Asynchronous service consists of 4 parts:

- ATTACH asynchronous service subtask
- DETACH asynchronous service subtask
- INVOKE asynchronous service subtask
- Asynchronous service subtask

INVOCATION OF ASYNCHRONOUS SERVICE

Before a POWER/VS task can use asynchronous service, an ATTACH request must be issued. This causes a DOS/VS subtask to be attached if one has not already been attached. Once an ATTACH request has been issued by the task, all SERVICE requests can be issued by the task. Finally, if asynchronous service is not needed any more, usually at task termination, a DETACH request must be issued in order to release the DOS/VS subtask.

Linkage to the asynchronous service function is established by the IPW$IAS macro instruction.

The macro invocation is as follows:

```
IPW$IAS TYPE=xxxx
```

Three different specifications of the type keyword are possible:

ATTACH - Attaches, if not already done, a DOS/VS subtask.

DETACH - Detaches, if necessary, the DOS/VS service subtask.

SERVICE - Invokes the asynchronous service subtask and passes the service request block to it.
Register 1 is assumed to point to the service request block (SRB).
It is the invoker's responsibility to build the SRB in the proper way.
Service is the default and may be omitted.

Following registers are used:

R0 - function type
R14 - return address of caller
R15 - entry point address of IPW$$AS

Note: When TYPE=SERVICE is specified, register 1 must point to the service request block.

The DOS/VS subtask is attached only once. A use count is used to keep track of how many POWER/VS tasks use the asynchronous service subtask. Whenever a POWER/VS task requests attachment of the DOS/VS asynchronous service subtask, the use count is incremented by one. When the use count is set to one, the actual attach of the subtask is performed. When no subtask is available, a message is issued to the operator, and the task requesting the attach is terminated. On the other hand, when a detach is requested by a POWER/VS task, the use count is decremented by one. If the use count becomes zero, the DOS/VS subtask is detached. However, the asynchronous service anchor block still exists.

When a service request is issued by a POWER/VS task, the service block (pointed to by register 1) is chained as the last entry in the service request queue, and the asynchronous service subtask is posted. The asynchronous service anchor block (ASAB) is unlocked and the task waits for the completion of the service request. After the completion of the service request, its ECB is posted by the subtask. The ASAB is locked and the service request block is unchained from the queue. If POWER/VS detects a chain error, an appropriate error message is written to the operator, and POWER/VS is abnormally terminated. Otherwise, the return code set by the subtask is analyzed and the appropriate action is taken.

When the attach function is entered the first time, storage for the asynchronous service anchor block (ASAB) is reserved. The ASAB exists as long as POWER/VS is active. The asynchronous service anchor block contains:

- Pointers to first and last entry in the service request queue. If there is no entry in the queue, the pointer to the first service request block is zero, but the pointer to the last service request block contains the address of the last service request being performed.
- DOS/VS subtask communication ECB
- Address of SETPRT routine in SVA
- Lock word

The asynchronous service function is serially reusable and is locked for the duration of the appropriate subfunction (ATTACH, DETACH, or SERVICE).

RESOURCE MANAGEMENT

Resource management is responsible for the protection of serially-reusable resources (control blocks) against concurrent access by more than one task. Entry to the services is made by means of the macro instructions IPW$RSR (reserve resource) and IPW$RLR (release resource).

Reserve Resource

This service is entered when a POWER/VS task issues a IPW$RSR macro instruction.

The resource lockword (bytes 28-31 of each resource control block) is examined. If the resource is not available (lock byte contains X'FF') the routine waits till it is available (by issuing IPW$WFL macro to task management). If the resource is available, ownership of the resource is established by storing the address of the TCB of the owning task in bytes 1 to 3 of the lockword.

Release Resource

This service is entered when a POWER/VS task issues a IPW$RLR macro instruction.

The resource lockword owner address is examined. If the task issuing the release request is not the resource owner the request is ignored. Otherwise, the lock byte in the resource lockword is set to zero so that the resource becomes available for use by any other task that may require it.

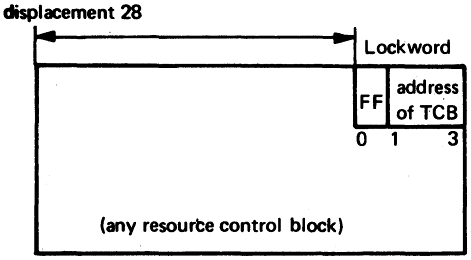


Figure 2.13. Resource lockword of a POWER/VS control block

STORAGE MANAGEMENT

Storage management controls the real storage allocated to the POWER/VS partition. Work space in the fixable area for a task is reserved and released as requested by the calling routine.

The storage control block (SCB), with storage assignment table (isomorphic map of all pages in fixed area), page control blocks and associated buffer control words (BCWs) are used to control the availability of real address storage in the POWER/VS partition (see Figure 2.14. The SCB is locked during handling resource/release request.

Reserve Work Space

This service is entered when a POWER/VS task issues a reserve work space (IPW$RSW) macro instruction. The buffers in the currently fixed pages are scanned to determine whether the required work space is available. If the space required is not available in the existing buffers, a new page is fixed (PFIX), in the POWER/VS real partition. Space is then allocated in the new page. The virtual and real addresses are passed to the calling routine.

Release Work Space

This service is entered when a POWER/VS task issues a release work space (IPW$RLW) macro instruction. The buffer is cleared (binary zero) and the appropriate buffer control words are updated. If the page is no longer in use (all buffers are cleared) the page is freed (PFREE).

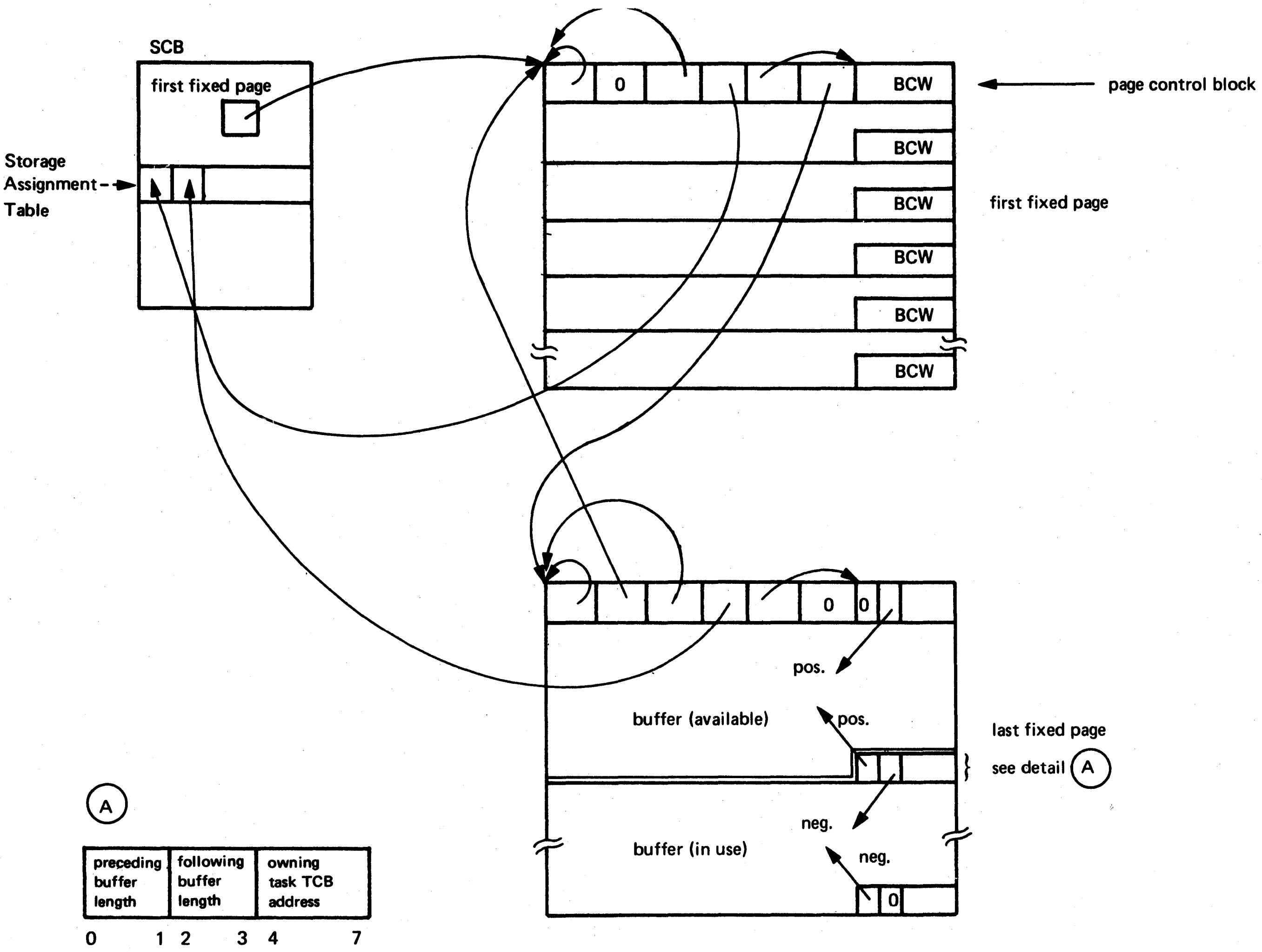


Figure 2.14. Storage management control blocks relationship

MESSAGE SERVICE

This service is invoked by a IPW$WTO or IPW$WTR macro instruction issued by the calling routine. It performs a console write operation or a write operation followed by a read operation, defined by information supplied by the calling routine in the message request word located in the TCB. (See Figure 2.16.) The message request word and reply request word contain the addresses of message and reply areas of the calling routine.

There is a message control block (MMB), which is locked for the duration of the operation. It contains the channel program, the CCB, the message output area and the reply input area.

The variable portions (lower case characters) of the message text are converted to indicate information pertinent to a specific task or queue entry.

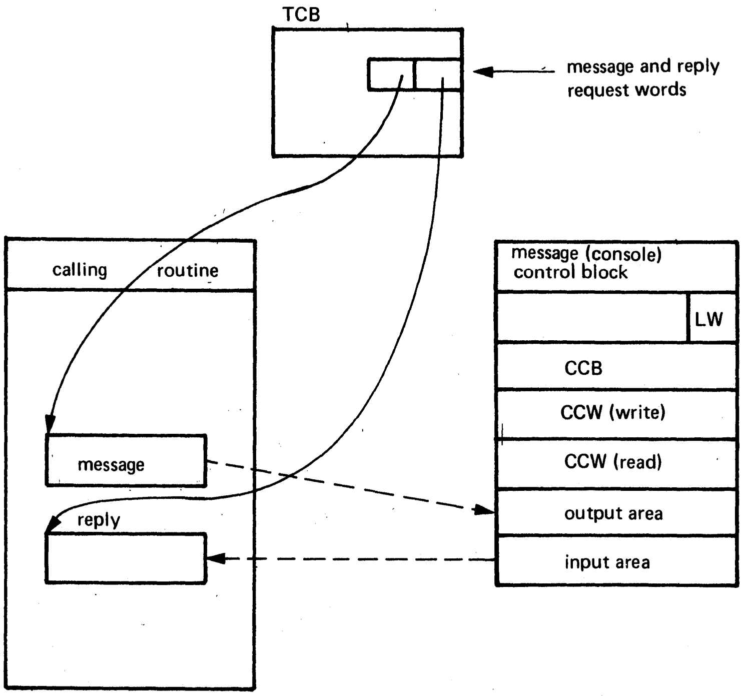


Figure 2.15. Message service control block relationship

DISK SERVICE

This service is invoked by IPWRDQ, IPWWTQ, IPW$RDD, or IPW$WTD macro instructions issued by the calling routine. It reads or writes records to the queue file or the data file defined by the information supplied by the calling routine in the I/O request words in the TCB.

The I/O request word contains the record seek address and the real and virtual addresses of the data or queue record area.

The first byte (M) in the record seek address is used to locate the associated Module Control Block (MCB) via the MCB address table in the CAT (control address table of IPW$$NU).

There is one MCB for every queue or data module, which is locked for the duration of the operation. It contains the CCB and skeleton channel program, which is appropriately initialized for each I/O operation.

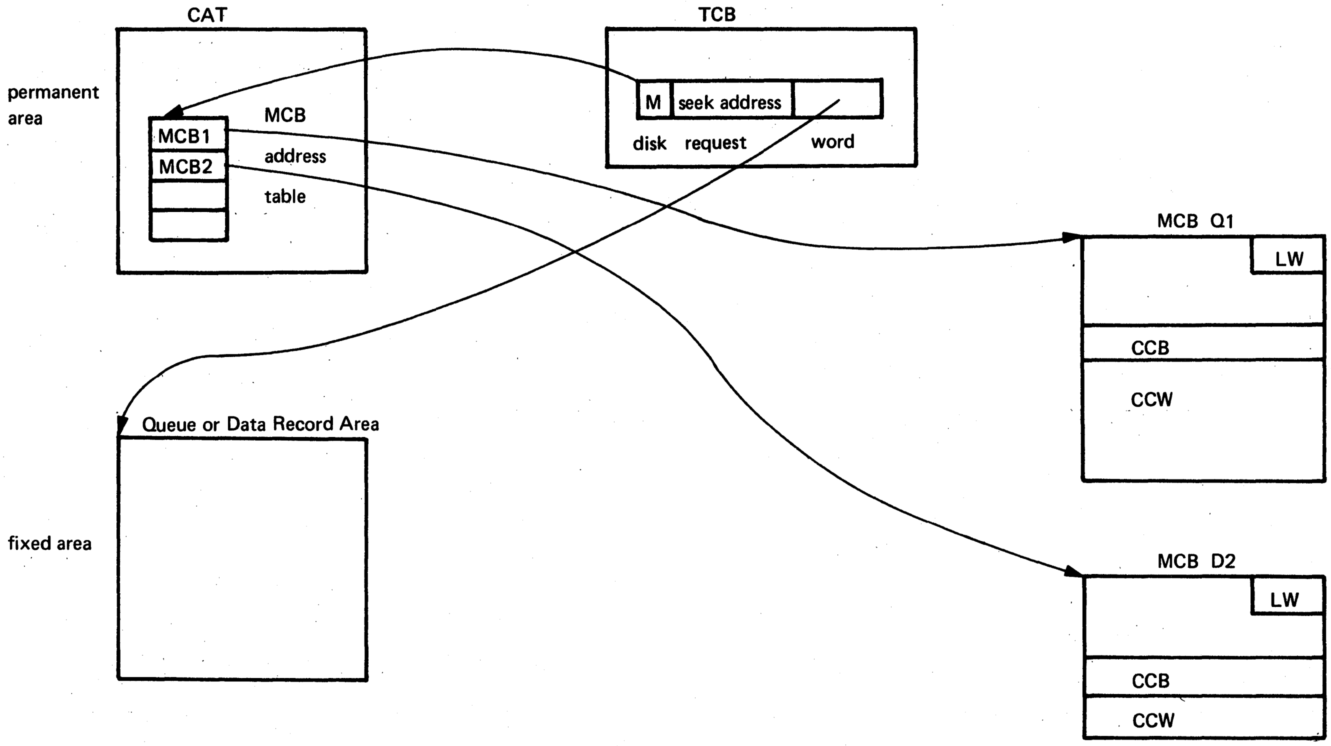


Figure 2.16. Disk management control blocks relationship

TAPE SERVICE

This service is invoked by IPWWTT, IPWRDT, or IPW$CTT macro instructions issued by the calling routine. It reads or writes records to tape file, or performs a tape control operation defined by information supplied by the calling routine in the I/O request words in the TCB.

Associated with the tape device is a Tape Control Block (TBB), counting the CCB and skeleton channel program which is appropriately initialized for each I/O operation.

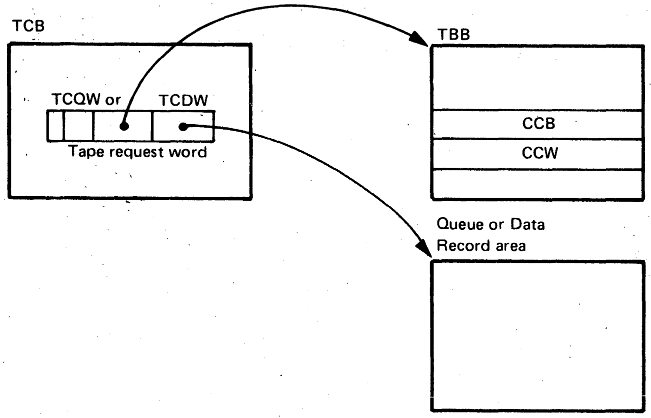


Figure 2.17. Tape service control blocks relationship

TIMER SERVICE

This service is invoked by the IPW$RDC macro instruction. It issues a GETIME standard macro instruction to obtain the time of day in packed decimal format. Also, the date field in the master record is updated with the value stored in the partition communication region.

VALIDATION SERVICE

This service is invoked by the IPW$VDA macro instruction. The data address and its associated length which are provided in the user-supplied channel command word and the address of the CCW itself are examined to ensure that they relate to a data area that the user is allowed to access.

The following table shows which areas the user is allowed to access:

	Read Operation	Write/ Control Operation
User's partition	Yes	Yes
LTA (logical transient area)	Yes	Yes
SVA (shared virtual area)	No	Yes

Command Processor

The command processor (IPW$$CP) will be under control either of a permanent TCB located in the nucleus code or a temporary TCB in the fixable area.

The permanent command processor task is invoked by the attention interface appendage when an operator command is received from the console.

The temporary command processor task is invoked by the IPW$ICP macro interface instruction.

On entry of the command processor the command to be analyzed and acted upon is contained in a command processor control block (CPB).

Initiation of the permanent command processor task

The attention routine $$BATTNA will pass control to the attention interface appendage in IPW$$NU for a potential POWER/VS command. In the appendage routine the command is verified and stored with its operands in fixed positions in the command processor control block (CPB). The command processor task is set dispatchable and normal return to $$BATTNA is taken.

In the case of an invalid command or if the command processor is not inactive an error return to $$BATTNA is taken, resulting in an invalid statement message.

Initiation of a temporary command processor task

The POWER/VS routine that wants to invoke the command processor for processing of a POWER/VS command issues a IPW$ICP macro instruction. Processing control is then given to phase IPW$$IC.

This phase builds and attaches a temporary command processor TCB, with the command and its operands in fixed positions in its CPB.

On exit, the temporary command processor task detaches itself and the permanent command processor task will place itself in inactive state. The permanent command processor has the highest priority of all common tasks in the task selection list. It enables the operator to maintain control over the POWER/VS partition in extreme circumstances.

COMMAND PROCESSING ROUTINES

In IPW$$CP, the command code contained in CPB is used to enter the appropriate command processing routine.

Note that the following commands if entered from a SNA terminal are processed by the IPW$$IB inbound processor:

- FLUSH
- GO
- RESTART
- SETUP
- SIGNOFF
- START
- STOP

The remaining commands are given to IPW$$CP for processing. After processing the command, control is returned to task management service. The following command processing routines exist:

PSTART - creates a TCB according to the operand specifications in the command. It attaches an execution reader task or a physical reader or writer task or RJE task. In case of a partition start it prompts the operator or the initiator task (if AUTOSTART) to supply the addresses of the devices to be spooled and builds the partition control block.

PSTOP - posts the termination type in the TCB for the task specified in the first operand of the command:

'S' stop immediately
'E' stop after processing of current queue entry
'R' stop immediately, resume operation on the next record if the task is started again.

PGO - posts the ECB in the TCB for the task specified in the command provided that the task was waiting for operator response (task status O).

PEND - changes the initiator TCB to terminator TCB.

The termination type E will be posted in all existing TCBs. In case of PEND KILL, an SVC 14 is issued without posting any termination type and POWER/VS is terminated immediately. If a printer address is specified, SYSLST is assigned to it.

PCANCEL - cancels status task.

PFLUSH - sets termination type in the TCB for the task specified by the command:

'F' current queue entry to be deleted
'H' current queue entry to be bypassed, but not to be deleted.

PRESTART - stores the number of records to be skipped, and in the case of a 3800 Printer also stores the new copy group index to be used, in the TCB of a local or remote writer task, according to the operand specifications. It also posts an index (type of skip) to this TCB:

X'04' - restart processing of the queue entry with specified record number.
X'08' - skip as many records forward as specified.
X'0C' - skip as many records back as specified.

PRELEASE - changes the disposition of the queue sets specified in the command:

1. Disposition H (hold) to D (dispatch)
2. Disposition L (leave) to K (keep).

PBRDCST - displays the message on SYSLOG (via IPW$WTO) or transfers it to the RJE message queue, (via IPW$RMS), depending on the operand specifications.

PALTER - removes the queue sets specified in the command from their class chains (via IPW$DQS); then priority, disposition, class, number of copies and remote-id can be changed. Finally, the queue sets are returned to the appropriate class chains (via IPW$AQS macro instruction).

PDISPLAY - prints a status report. Depending on the option(s) specified in the PDISPLAY command, one of the following "subprocessors" will be called:

DSPLYQ - displays queue status.

DSPLYA - displays active reader/writer tasks along with job name and job number.

DSPLYM - displays task waiting for operator (task status O) and repeats message issued before task was placed in the wait state.

DSPLYJOB - attaches the print status task (IPW$$PS) to read the queue file and extracts the status information required for a report to be printed on SYSLOG or on the specified printer.

DSPLYMSG - displays ALLUSER-type messages from remote message queue.

DSPYLT - display time, date, pages fixed and number of tasks.

PINQUIRE - displays status information for RJE lines to central operator:

- not supported (no line table entry exits)
- not initiated (no line control block or SNA control block exists)
- inactive (no sign-on)
- processing RJE-id (sign-on)

PACCOUNT -

- builds a TCB for a save account task which will save or delete the account file (IPW$$SA)
- conditionally assigns a free program LUB entry
- reserves work space for the TCB and attaches it.

PDELETE - deletes:

1. queue sets according to the operand specifications in the command. (Via IPW$DQS and IPW$FQS macro instructions), or
2. messages according to the operand specifications from the RJE message queue.

PSETUP - stores the number of pages to be printed in the TCB for a list task according to operand specifications. The task is posted dispatchable. The command is ignored when no list task exists or when the list task is not waiting for operator action.

ACCOUNT FILE PROCESSING

Operations on the account file are performed by two functional routines: the put account record function (IPW$$PA) invoked by a IPW$PAR macro instruction, and the get account function (IPW$$GA) invoked by a IPWOAF, IPWGAR, or IPW$CAF macro instruction.

The put account function routine will accept account records for the POWER/VS partition and the partitions running under control of POWER/VS (see Figure 2.18). The account records (VARUNB format) will be written to the POWER/VS account file under control of the account control block (ACB). The remaining file capacity is checked against a 20% limit. A warning message is issued if the limit is exceeded.

If the remaining capacity of the account file does not allow to store a presented record, the task concerned is placed in a wait state (wait for ECB posting in account control block), until the account file is emptied by the save account task.

The get account function routine is broken down into three operations:

- Open account file for get mode processing, invoked by IPW$OAF macro instruction.
- Get account record to retrieve the next sequential record from the account file, invoked by IPW$GAR macro instruction.
- Close account file to restore the mode for put account record processing, invoked by the IPW$CAF macro instruction.

OPEN Account File

The account control block is initialized for read operations (get mode) to start on the first record in the account file.

CLOSE Account File

The account file records are erased by writing EOF records on each track. The account control block is initialized for write operations (put mode) to start on the first record in the account file. The task(s) waiting for posting of the ECB in the account control block are now allowed to continue processing.

Save Account Task

The save account task is attached by the command processor after a PACCOUNT command is given. The save account routine (IPW$$SA) is entered when the task gets control. Its purpose is to empty the account file, erase it, or save it on another storage medium (disk, tape, or punch queue).

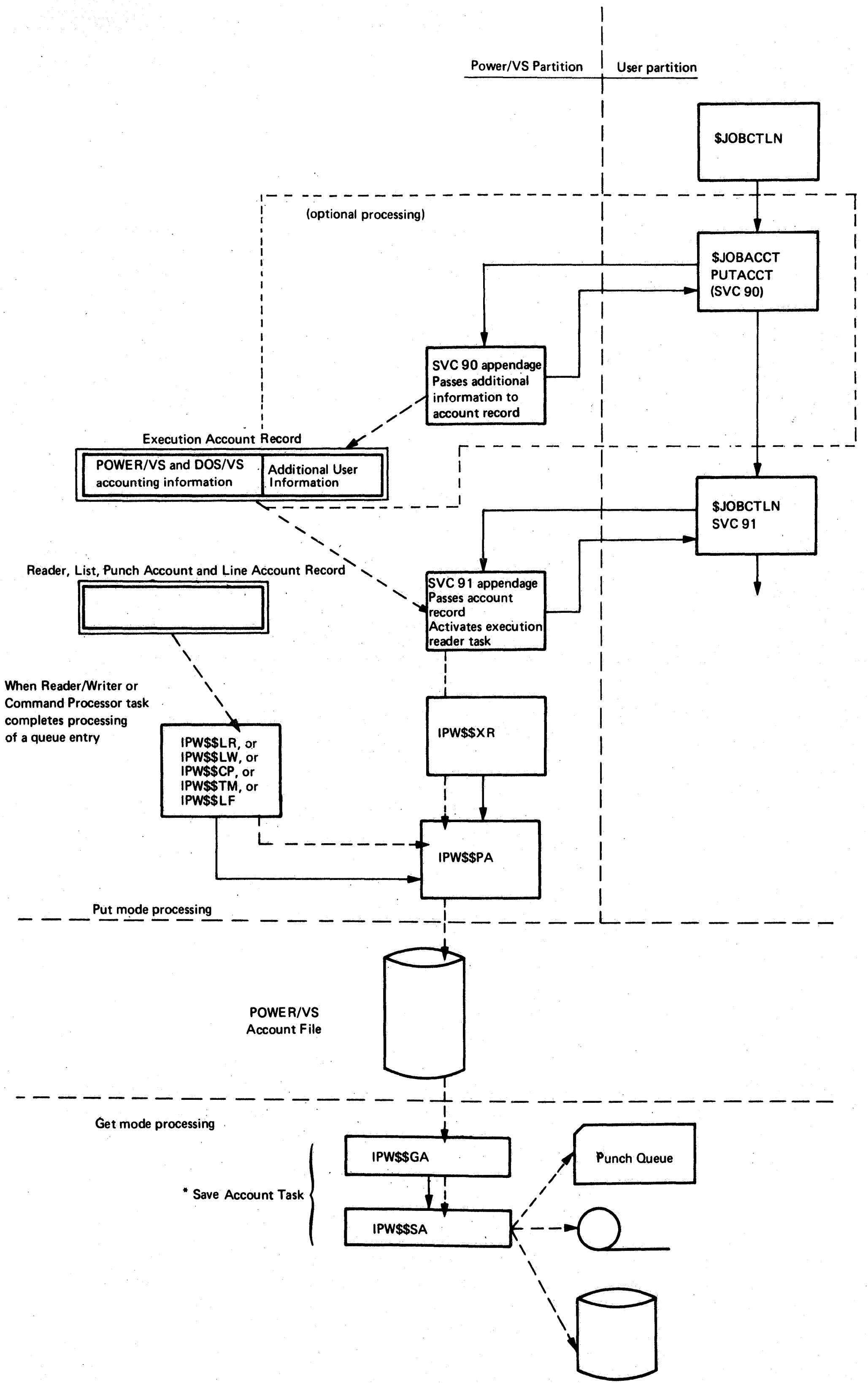


Figure 2.18. Relationship between POWER/VS and DOS/VS job accounting

RJE,BSC

The POWER/VS RJE,BSC operations are performed by phases IPW$$TM (mainline) and IPW$$MS (message handler). IPW$$TM is entered from IPW$$I2 at POWER/VS initialization time to save the RJE part of the generation table, after which it returns control to IPW$$I2. IPW$$TM consists of POWER/VS subtasks and subroutines: the line initialization, the line manager, and the RJE,BSC reader/writer tasks, the remote access method (RTAM) subroutines, the subroutines to handle the RJE,BSC remote commands (* ..SIGNON, *..SIGNOFF, * ..START, * ..STOP, * ..SETUP, and * ..GO), and the remote message writer subroutine. IPW$$MS contains the code of the POWER/VS message handler.

Figure 2.19 shows the relationship between the RJE,BSC tasks described in the following paragraphs.

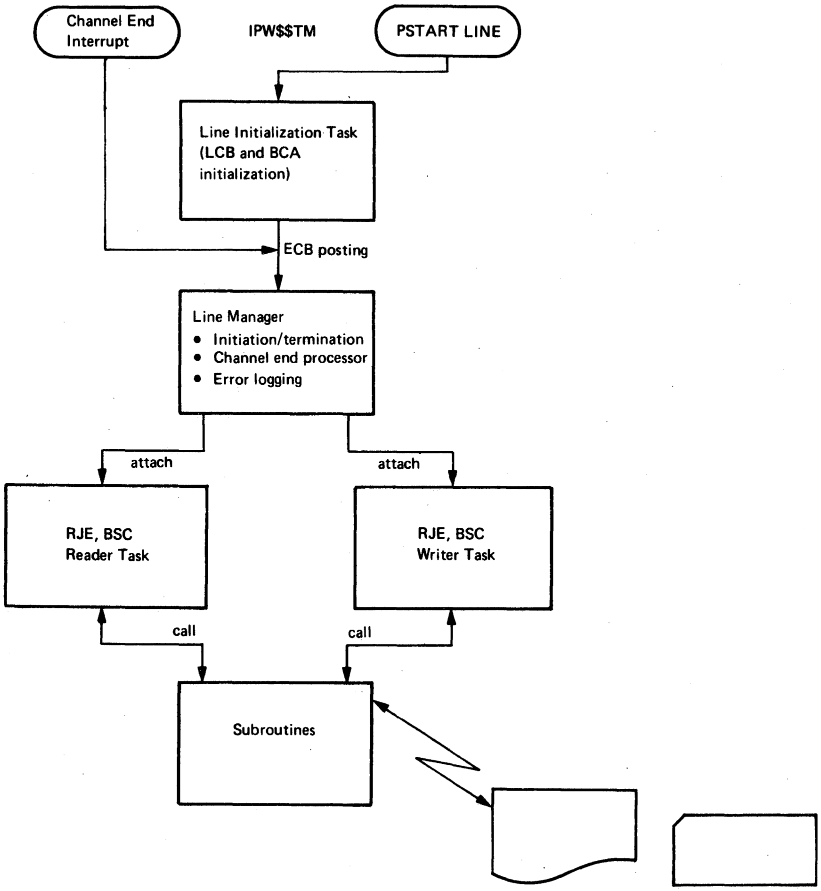


Figure 2.19. RJE,BSC Relationship

RJE,BSC LINE INITIALIZATION

The line initialization task, which uses the TCB of an RJE,BSC reader task, is attached by IPW$$CP when a line is started by the central operator. It reserves real storage for a line control block (LCB) and buffer control area (BCA), and performs the initialization of these blocks. (For a detailed description of these control blocks refer to Section 5 "Data Areas".) This task then activates the line manager and detaches itself.

RJE,BSC LINE MANAGER

The line manager task, which consists of a channel-end processor routine, an initiation/termination routine and an error logging routine, is attached during POWER/VS initialization. This task is activated whenever:

- A channel end is detected, by the channel end appendage routine (IPW$$NU).
- A line is started, by the line initialization routine (IPW$$TM).
- A line is stopped, by the PSTOP and PEND processors (IPW$$CP).

The line manager continuously scans the BCAs and LCBs in the system as shown in Figure 2.20, to detect which of the above events has caused this task to be activated.

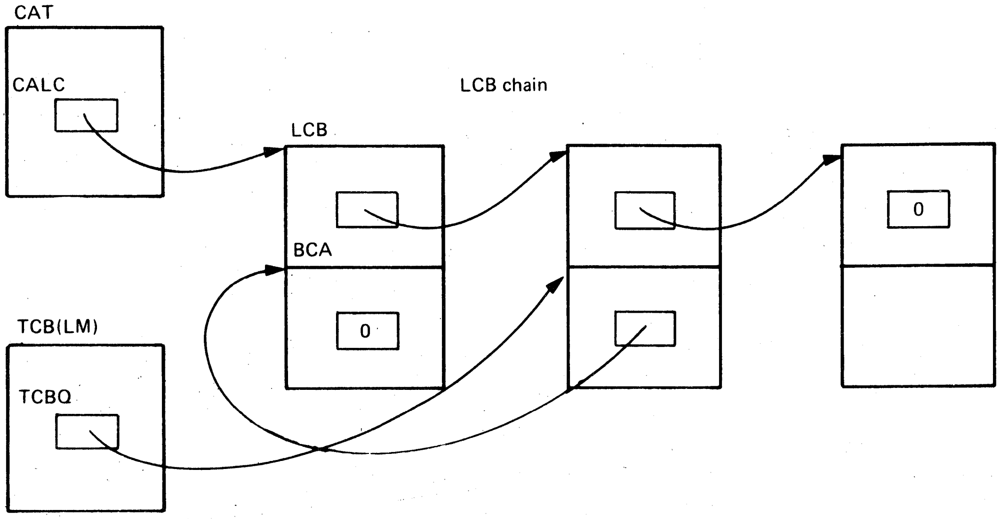


Figure 2.20. Scanning the LCBs and BCAs

When a channel end is detected, the channel-end processor is called. This routine consists of two sections. One section handles terminal responses during the CCW prepare sequence, which is used when the central system has messages or output to be sent to the terminal or when input is expected from the terminal. The other section handles terminal responses during the CCW read/write sequence, which is used to dynamically attach the reader/writer tasks when an ENQ/ACK is received from the terminal. (See Figure 2.21 for CCW sequences.)

When a start/stop condition is detected, the initiation/termination routine is called. This routine resets the line status and puts the line in a dormant state.

When a line error is detected, the error logging routine is called. This routine writes counter overflow records, unit check records, and end-of-day records to the DOS/VS recorder file on disk.

The line manager only detaches itself when a PEND command is given, after all LCBs and BCAs have been released.

Prepare Sequence	Read Sequence	Write Sequence
ENABLE NOP or WRITE ENQ/EOT or PREP READ (response)	ENABLE WRITE (response) READ (text)	WRITE (text) WRITE (end-of-text) READ (response)

Figure 2.21. CCW Sequences

RJE,BSC READER

An RJE,BSC reader task is dynamically attached by the line manager when an ENQ is received from the terminal. The RTAM open subroutine is first called to put the line into read mode by changing the CCW prepare sequence into a CCW read sequence. It then reads the first block of data, which is subsequently broken down into logical records by the RTAM get subroutine. Then the get subroutine automatically reads the next block of data into storage. When an RJE,BSC command is read at job boundary, it is processed by the RJE,BSC command processor routine. Logical data records are passed to the logical reader. After EOT, the RTAM close subroutine is called to put the line back into idling mode by restoring the CCW prepare sequence, after which the RJE,BSC reader task detaches itself. If EOT is not detected on a job boundary, however, the RJE,BSC reader task puts itself in the inactive state.

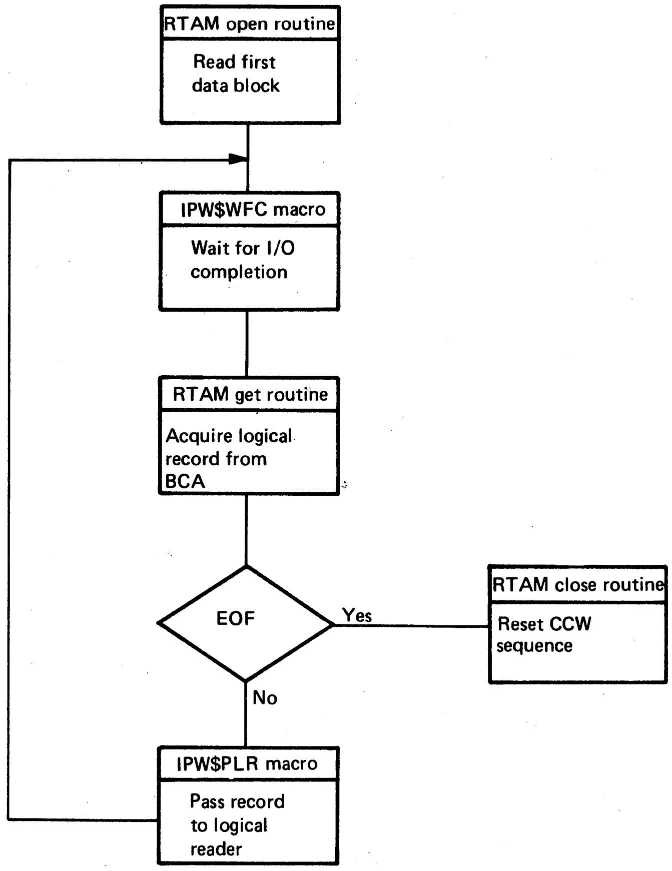


Figure 2.22. RJE,BSC Reader Flow

RJE,BSC WRITER

An RJE,BSC writer task is dynamically attached by the line manager when an ACK is received from the terminal. If any messages are to be transmitted, the remote message writer subroutine is called first. If no messages are to be transmitted, the RTAM open subroutine is called to put the line into write mode by changing the CCW prepare sequence into a CCW write sequence. It then obtains logical data records from the logical writer and passes them to the RTAM put subroutine to be grouped into physical data groups. After being grouped, the data is written out to the terminal. At end of job, the RTAM close subroutine is called to write the last data buffer and to put the line back into idling mode by restoring the CCW prepare sequence. The logical writer is again called to delete the queue record, after which the RJE,BSC writer task detaches itself. If the output device remains ready, a new writer task is attached by the line manager after a 6 to 9 seconds delay that allows the operator to start a reader task without interrupting an active writer task. If the output device becomes not ready during output processing, or when a forms change is required, the RJE,BSC writer task puts itself in the inactive state.

A hot writer facility is provided if the logical writer passes a no-more-output condition. The hot list/hot punch switch in the LCB is turned off. This prevents a new writer task from being attached until the switch is turned on again by queue management.

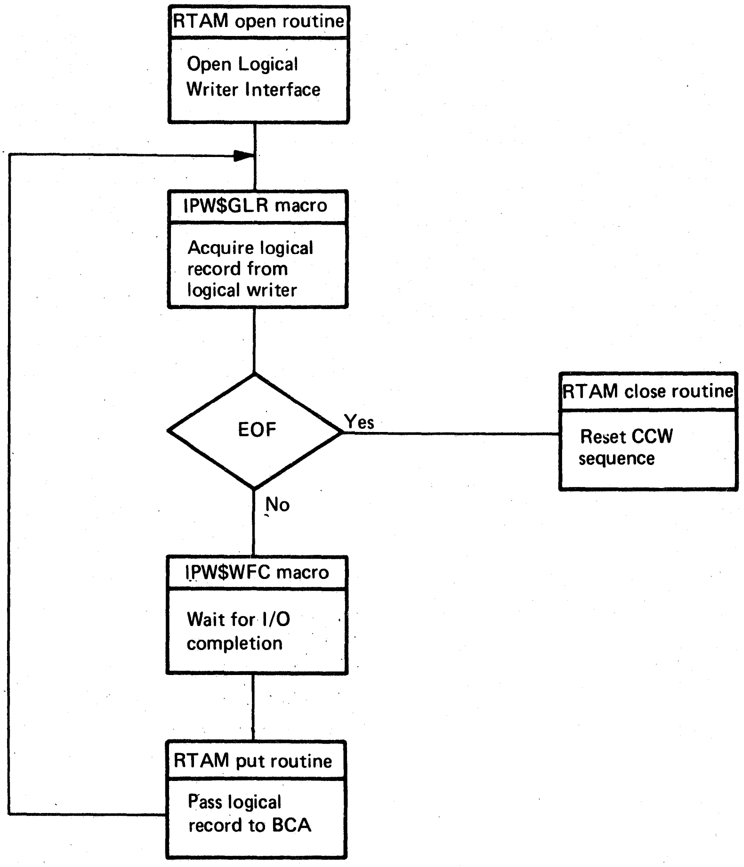


Figure 2.23. RJE,BSC Writer Flow

MESSAGE HANDLER (IPW$$MS)

This phase handles local as well as remote message requests. It is called by the message service routine in the POWER/VS nucleus whenever an IPW$WTO or IPW$WTR macro (local), or an IPW$RMS macro (remote) is issued.

For local message requests, information about the message to be issued is supplied by the calling routine in the message request word in the TCB. The message length is examined and, if necessary, truncated to the maximum of 72 characters. The message text is scanned to determine whether any message modification is necessary. If so, the message text is modified in the appropriate modification routine. Then a console write operation (for an IPW$WTO macro), or a console write operation followed by a read operation (for an IPW$WTR macro) is performed. For PUTSPOOL, GETSPOOL, and CTLSPOOL processing, the first 44 characters of the message text are placed in the user's buffer area at a displacement offset of 28 bytes.

For remote message requests, the function to be performed is indicated in the function indicator byte in the Remote Message Control Block (RMCB). The following functions are performed:

- Queue remote messages (BSC and SNA) to the virtual message queue
- Delete messages from the queue when it is completely full with pending messages
- Display ALLUSER type messages by passing them to the command processor
- Delete ALLUSER type messages
- Queue ALLUSER type messages to the ALLUSER type message queue
- Delete BSC messages from the LCB subchain
- Locate the first pending message for a specific BSC or SNA user
- Delete SNA messages from the SNA delete subchain
- Delete SNA messages temporarily by moving the entries from the SNA live subchain to the SNA delete subchain
- Add temporaily deleted SNA messages to the SNA live subchain

RJE,SNA

DESCRIPTION

POWER/VS RJE,SNA provides support for the SNA terminals that use Synchronous Data Link Control (SDLC). The communication with the SNA logical units is accomplished by using the VTAM access method at the API level. POWER/VS controls the SNA work stations through a logical connection. All physical connections within the logical path are controlled by VTAM and NCP. Since VTAM does some of its processing under the PIB of the DOS/VS application task, the DOS/VS supervisor handles VTAM page faults as if they were POWER/VS page faults. In order to minimize the effect of these page faults on non-RJE tasks, POWER/VS attaches a DOS/VS subtask under whose PIB VTAM processing can be executed.

The POWER/VS RJE,SNA operations are performed by the following phases:

- IPW$$SN (SNA manager)
- IPW$$LH (SNA logon processor 1)
- IPW$$LN (SNA logon processor 2)
- IPW$$IB (SNA inbound processor)
- IPW$$OB (SNA outbound processor)
- IPW$$OC (SNA outbound compaction processor)
- IPW$$MP (SNA message processor)
- IPW$$LF (SNA logoff processor)
- IPW$$VE (VTAM exit routines)

Figure 2.24 shows the POWER/VS RJE,SNA interrelationship.

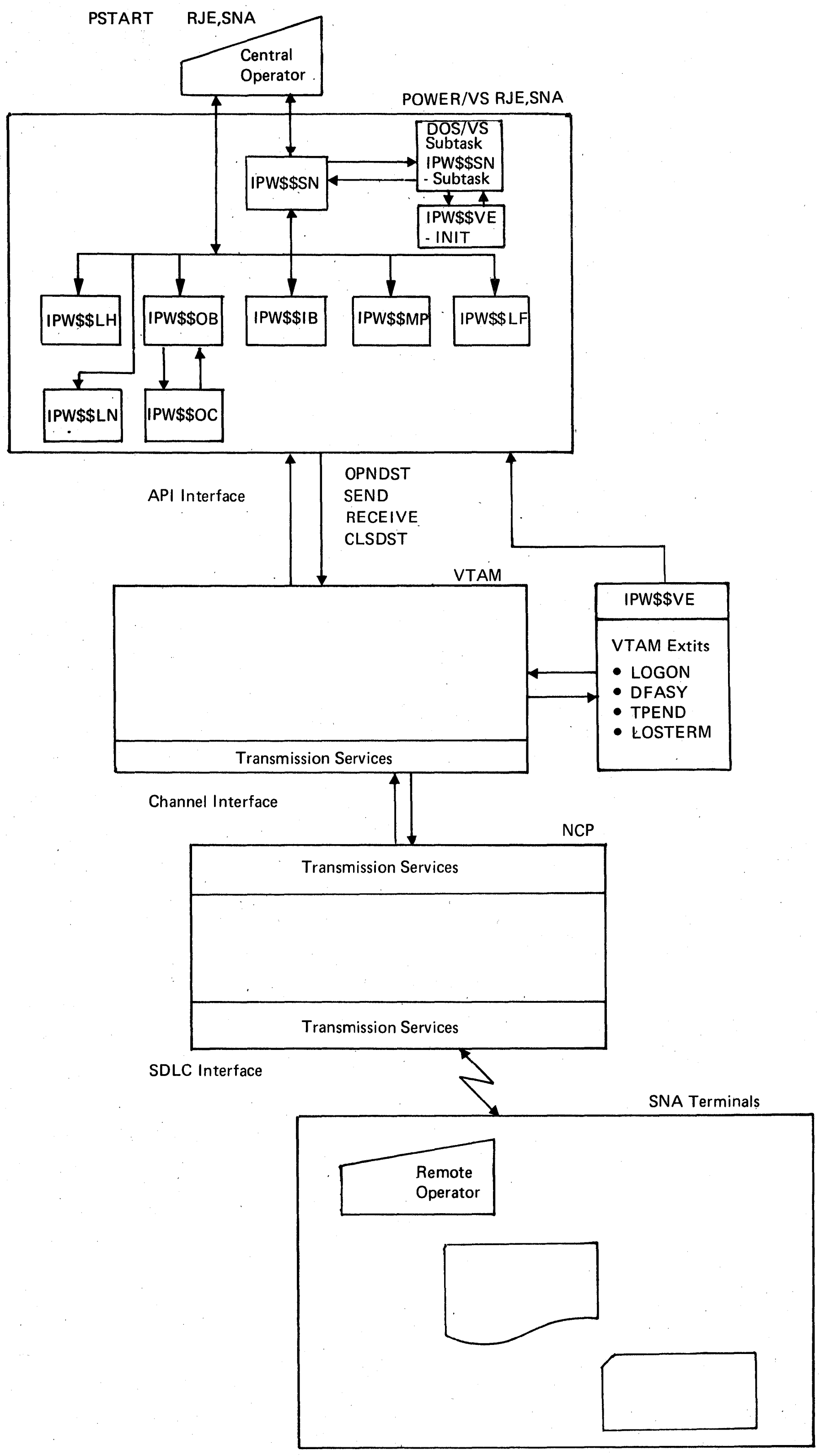


Figure 2.24. RJE,SNA Interrelationship

Initialization

If the central operator issues the PSTART command for SNA, the POWER/VS SNA manager is attached to the TCB chain and controls the activation of any inbound or outbound process related to a workstation and its associated sessions on a demand basis. The SNA manager attaches a DOS/VS subtask to the POWER/VS maintask in which the POWER/VS application opens the interface with the VTAM access method by issuing the OPEN ACB macro. The ACB points to an EXLST control block, which defines the asynchronous exit structure within the POWER/VS system to VTAM and consists of LOGON, LOSTERM, TPEND and DFASY exits. After the interface to VTAM is opened any logon request to POWER/VS will be queued by VTAM. After the OPEN ACB request has been completed successfully POWER/VS issues a SETLOGON START macro to enable VTAM to schedule the POWER/VS logon exit routine.

Logon Processing

Upon a logon command from a logical unit VTAM schedules the LOGON exit. The LOGON exit routine queues the request with the LOGMODE table entry address.

In a Multiple Logical Unit (MLU) environment, VTAM/NCP do not associate sessions within a work station concept. VTAM/NCP only see individual sessions between POWER/VS and the physical terminals. Hence, POWER/VS is responsible to associate sessions with work stations according to the DATA operand of the LOGON command.

The logon routine No. 1 (IPW$$LH) in POWER/VS processes all logon requests in the logon queue. For each logon request the routine performs the following functions:

a. Utilizing LOGMODE table entry address and a logon workarea, it requests the user DATA and BIND parameters from VTAM issuing the INQUIRE SESSPARM macro.

b. It performs syntax checking of the REMID and verifies its existence as specified in the PRMT macro.

c. It checks the corresponding password if specified.

d. It moves the 16 bytes of user information of the DATA into the session account record without validity checking.

e. It verifies that the logical unit is authorized to log on with this REMID, provided that an LU=(name,...) parameter has been specified in the PRMT macro.

f. It validates the BIND parameters.

In turn, the IPW$$LH routine checks whether or not any logical unit has been logged on with the same REMID. If no logical unit has as yet logged on with the same REMID, i.e., this work station is logging on its first session, the routine initializes all relevant work station and session related control blocks for this REMID.

If any logical unit has already logged on with the same REMID the routine verifies that this current logon request does not exceed the SESSLIM according to the specification in the PRMT macro. The routine initializes the relevant session related control block, if SESSLIM is not exceeded. Else it rejects the logon reuqest.

When the control blocks have been initialized the routine causes the logon No. 2 routine (IPW$$LN) to be attached, and it proceeds processing the next logon request from the queue.

The IPW$$LN routine completes processing the logon request for this session. It issues the OPNDST ACCEPT and SESSIONC SDT macros to VTAM and issues messages to the remote and central operators.

The Logon request may be rejected for several reasons:

a. Invalid user logon parameters, whereby VTAM sends a message to the central operator.

b. System error (non-zero return code from VTAM).

c. The BIND parameters are not accepted.

d. The number of logon commands for a given workstation exceeds the number specified by the SESSLIM parameter in the PRMT generation macro.

e. The name of the logical unit logging on with a given REMID is not associated with the REMID specified in the PRMT macro.
Note: This correlation of LU name and REMID is tested if and only if at least one LU name has been specified.

f. The INQUIRE or OPNDST macro to VTAM was not successful.

g. A GETVIS failed.

h. The logon request provides an invalid REMID or password.

POWER/VS then issues a CLSDST to VTAM resulting in a network procedure error being sent to the work station. Messages are sent to the central operator giving a logon reject reason code:

```
'1V06I UNABLE TO LOGON luname RC=' or
'1V26I INVALID REMID, PASSWORD, OR LUNAME RC='
```

The BIND Format

The BIND parameters exchanged between POWER/VS and SNA terminals are detailed in the DOS/VS POWER/VS Installation Guide and Reference, in the Appendix describing "RJE,SNA I/O Specifications." POWER/VS is flexible in its BIND requirements. Each BIND parameter affords one of the following characteristics:

- ignored parameter (I)
- enforced parameter (E)
- mandator parameter (M)
- variable parameter (V)

Ignored parameters are neither tested nor modified by POWER/VS. Enforced parameters are dictated by POWER/VS on the secondary logical unit. Mandatory parameters are tested, and if incorrect, the logon request is rejected. The variable parameters are copied.

Host-Work Station Communication

Logical records are grouped into RU's which are logically grouped into chains. Output related to one POWER/VS queue entry (job or segment) is sent as one chain unless interrupted by an inbound flow or an outbound message. An outbound job is always sent as a DS (data stream). Messages are sent as an only-chain.

Input related to one POWER/VS queue is not related to a chain by POWER/VS. POWER/VS only identifies job boundaries according to POWER/VS-JECL or DOS/VS-JCL statements with the exception that an end bracket forces End of JOB (EOJ), if no valid POWER/VS job delimiter was found. It is the option of the work station to associate jobs and chains if this association simplifies ERP (error recovery procedures) at the work station.

POWER/VS supports all three SNA function management headers for outbound, i.e., FMH1,FMH2 and FMH3, but only FMH1 for inbound. Concatenation of FMH's is not supported. If POWER/VS receives an FMH with the concatenation bit on, it returns an exception response.

Function Management Header Type 1 (FMH1). POWER/VS supports the standard 6-byte FMH1 for device selection and delimiting data set operations.

The Figure 2.24A details the FMH1 format layout.

Byte	Bits	Name	Content	Description
0	0-7		X'06'	Length of FMH1
1	0	FMHC	B'0'	Concatenation not supported
	1-7	TYPE	B'0000001'	Type 1 FMH
2	0	DEMAND		
		SELECT	B'0'	Not supported
	1-3	MEDIA	B'000'	CONSOLE
			B'001'	EXCHANGE MEDIA (not supported)
			B'010'	CARD
			B'011'	PRINT
				All other codes not supported.
	4-7	LOGICAL	X'0'	1st logical device
		ADDRESS	X'1'	2nd logical device for print
			X'2'	3rd logical device data only
3	0	STACKREF		Stack reference indicator
			B'0'	Refers to DS begun by sender
			B'1'	Refers to DS begun by receiver
	1-7		B'0000000'	Reserved
4		PROPERTY		
	0-2		(Note 1)	DS selection
	3	DST	B'0'	Basic exchange not supported
	4		B'0'	Reserved
	5	CMI	B'0'	No compression
			B'1'	Compression (outbound print only)
	6	CPI	B'0'	No compaction
			B'1'	Compaction (outbound print only)
	7		B'0'	Reserved
5	0-7	ERCL	X'00'	Basic exchange record length, not supported
6-n				DSNAME which is defined by architecture in bytes 6-n is not supported by POWER/VS.

Figure 2.24A. FMH1 Format

Note 1: The data stream selection bits are used in combination. The valid combinations are:

B'000' - Resume suspended data stream (RDS)
B'001' - End current data stream (EDS)
B'010' - Begin data stream (BDS)
B'011' - Begin and end data stream (BEDS)
B'100' - Suspend current data stream (SDS)
B'101' - Abort (abnormally end) current data stream (ADS)
B'110' - Reserved
B'111' - Reserved.

The following should be noted:

1. With the resumption of a suspended outbound data stream, POWER/VS will not change any of the FMH options selected in the original FMH.

2. An FMH may exist in an RU only at first-of-chain (FC) or only-chain (OC). The presence of an FMH is signaled by the format indicator bit in the request header. If data is received with no FMH where an FMH is expected, the default FMH applies as in Figure 2.24B.

3. When the data stream selection bits are set to B'011' the entire data stream is being sent within one chain, including the FMH. Print and card media output data are initiated by only-chain FMH's indicating BDS, followed by chain(s) of data, and terminated by an only-chain FMH indicating EDS.

4. An FMH1 (BDS) is sent prior to, and an FMH1 (EDS) after, each job output or segment. FMH1 (BDS) is sent after FMH3.

Byte	Bit	Name	Content	Description
0	0-7	length	X'06'	Length of FMH1
1	0	FMHC	B'0'	Concatenation not supported
	1	reserved	B'0'	
	2-7	TYPE	B'000001'	Type 1 FMH
2	0	DEMAND SELECT	B'0'	Not supported
	1-3	MEDIA	B'000'	CONSOLE
	4-7	LOGICAL ADDRESS	X'0'	1st console
3	0	STACKREF	B'0'	Stack reference indicator
	1-7		B'0000000'	
4	0-2	DS sel.	B'011'	Begin and end of data stream
	3	DST	B'0'	Basic exchange not supported
	4	reserved	B'0'	
	5	CMI	B'0'	No compression
	6	CPI	B'0'	Compaction not supported
	6	CPI	B'0'	Compaction not supported
	7	reserved	B'0'	
5	0-7	ERCL	X'00'	Basic exchange record length, not supported

Figure 2.24B. Default FMH1

Function Management Header Type 2 (FMH2). The FMH2 represents the peripheral data set information record (PDIR). It carries information relative to the destination selected by FMH1. POWER/VS only supports FMH2 outbound, but not inbound.

The format of the FMH2 is shown in Figure 2.24C.

Byte	Bits	Name	Content	Description
0	0-7	Length	X'44'	Length of FMH2
1	0	FMHC	B'0'	No concatenation
	1			Reserved
	2-7	TYPE	B'000010'	FMH type 2
2		CODE	X'01'	PDIR
3	0-7	Identif.	X'00'	Ordinary data set
4-11		DATE	MM/DD/YY	Date of queue set creation. EBCDIC characters in the form MM/DD/YY provided by POWER/VS internally.
12-19		TIME	HH.MM.SS	Time of queue set creation. EBCDIC characters in the form HH.MM.SS provided by POWER/VS internally.
20-27		FORMS	C'ccccbbbb'	Forms name. EBCDIC characters in the form C'ccccbbbb'. Default is all blanks. The forms name can be provided only by the * $$ LST or * $$ PRT JECL statements.
28-35		FCB	C'cccccccc'	FCB name (1-8 characters). EBCDIC characters in the form C'cccccccc' left justified. Default is all blanks. The FCB name can be provided only by the * $$ LST or * $$ PRT JECL statements.
36-43		TRAIN	C'bbbbbbbb'	(Not supported)
44-51		COPIES	C'cccccccc'	Copies (1-8 characters). Indicates the number of additional copies, i.e., zero means one copy. EBCDIC characters (digits), right justified, with leading zeros suppressed except low order digit. The maximum number of additional copies is 98. The number of additional copies plus one can be provided by the * $$ LST or * $$ PRT JECL statements or by means of the PALTER (central operator) or ALTER (remote operator) commands.
52-59		VOLIO	C'cccccccc'	Volume of I/O. EBCDIC characters (digits) in the form C'cccccccc' right justified with leading zeros suppressed. If printer selected the field indicates the number of expected print lines. Information provided by POWER/VS internally.
60-67		JOB NAME	C'cccccccc'	Job name. EBCDIC characters in the form C'cccccccc' left justified. The jobname can be provided by the * $$ LST or * $$ PRT JECL statement. Default is generated to AUTONAME by POWER/VS.

where: b=blanks
c=EBCDIC characters

Figure 2.24C. FMH2 Format

The FMH2 is sent as an only-chain in DS state after FMH1 has been sent, provided that the PDIR bit in the BIND parameters was turned on at logon time of the session. If the PDIR bit is off the SETUP/GO procedure will be performed.

The PDIR is always sent if the BIND indicates so, regardless of whether or not forms change is required. Without PDIR indicated in BIND the SETUP/GO procedure is performed only if forms change is required.

Function Management Header Type 3 (FMH3). The FMH3 carries information relative to the entire session. Type 3 information applies to all destinations reached through this session. The FMH3 is sent as only-chain and it is not chained with another FMH, nor does the RU contain any other data. POWER/VS supports only outbound FMH3. If POWER/VS receives an inbound FMH3 it returns an exception response.

The format of the FMH3 is shown in Figure 2.24D.

Byte	Bits	Name	Content	Description
0	0-7	LENGTH	(Note 1)	Length of FMH3
1	0	FMHC	B'0'	No concatenation
1	1	Reserved		
1	2-7	TYPE	B'000011'	Type 3 FMH
2	0-7	FUNCTION	X'02'	Compaction table
3	0-7	MASTER NO	3-16	No. of master characters
4-n		TABLE	(Notes 1 & 2)	Compaction table characters

Figure 2.24D. FMH3 Format

Note 1: Length is dependent on length of the compaction table. It can be calculated by

$$\text{length} = 4 + 256 - (m \times m) \text{ for } m < 16$$
$$\text{length} = 4 + 16 \text{ for } m = 16$$

where m is the number of master characters.

Note 2: The compaction table is transmitted in row major order, starting at the bottom row and omitting the cells in the upper left corner m by m matrix, where m is the number of master characters. The cells in the submatrix represent compacted characters.

The FMH3 including length indication is generated by the PCPTAB macro.

An FMH3 is sent to the secondary logical unit whenever a job is to be transmitted outbound in compacted form using a compaction table other than the one currently valid for the session. The FMH3 is always sent as only-chain, without data, and between DS state. The FMH3 RH may or may not indicate begin bracket depending on whether or not the session is already in bracket state.

Initiation of Data Processing

Data Inbound Processing. An inbound processor task is attached for a given session by the SNA manager in the following cases:

a. A VTAM RECEIVE ANY is satisfied:
The SNA manager determines the session on which an inbound flow is to be expected by means of a pointer in the user field of the RPL. It then attaches an inbound processor, and reissues RECEIVE ANY to allow input from other sessions to be accepted.

b. An inbound flow is interrupted for an inbound message:
The inbound processor being interrupted posts the SNA manager which attaches a second inbound processor for this session.

c. An outbound flow is interrupted for an inbound flow or message:
The outbound processor being interrupted posts the SNA manager which attaches an inbound processor for this session and reissues RECEIVE ANY.

In all three cases the inbound processor issues RECEIVE SPECIFIC. It verifies whether or not the device (RDR1 or console) selected by FMH1 (implicit or explicit) is already in use. If in use it rejects the inbound flow.

If RECEIVE ANY is satisfied on all sessions at approximately the same time, an inbound processor is attached on each session, whether free or involved in an outbound operation. Each inbound processor performs the verification described above upon reception of an

FMH1. This verification results in rejection of all simultaneous inbound flow requests, with the exception of one reader and one console at the maximum.

Data Outbound Processing. An outbound processor task is attached for a given session by the SNA manager in the following cases:

a. Outbound Data.
When a job is available in an output queue (list or punch) of a given class with a given REMID, the queue management (IPW$$NQ) routine of POWER/VS scans the control blocks for a match of the REMID. When the REMID is found the routine scans the classes of all outbound devices for this REMID. These classes are assigned to the devices by means of the START command. When a match is found, and if the device has been started, the routine flags the device and posts the SNA manager.

The SNA manager finds the flagged device and searches for a free session. If a free session is found the SNA manager attaches an outbound processor which starts processing the job output until the output queue is empty. When the queue is empty the outbound processor resets the device flag, posts the SNA manager and detaches itself.

The SNA manager does not take further action if no free session is found. It will repeat the attempt when it is posted again, e.g., when a processor is detached.

b. Outbound Message.
Outbound messages are queued by the message service routine (IPW$$MS). Whenever the routine queues a message for a given REMID it posts the SNA manager. The SNA manager searches a free session to the workstation identified by the REMID. If a free session is found the SNA manager attaches the message processor which sends the message to the workstation.

If no free session is found the SNA manager searches a session involved in an outbound flow. The search begins for a session which is waiting for a GO command or RESTART following intervention required. If not found, then the search continues for some session which is transmitting. If found, it flags the associated control block which causes the outbound processor to suspend. Upon suspension the SNA manager attaches the message processor.

No action is taken by the SNA manager if no session involved in an outbound flow is found. The SNA manager will repeat the attempt when it is posted again, e.g., when a processor is detached.

Once attached, the message processor transmits all messages queued for a given REMID and detaches.

Interruption of Data Processing

Interruption Of Data Outbound. The interruption of the outbound processor can be caused by the following conditions:

a. A SIGNAL from the work station has been received.

The outbound processor forces the termination of the current chain, sends an FMH1 with suspend DS and a change direction indication to the logical unit. It then posts the SNA manager which attaches the inbound processor.

b. A message is pending.

The outbound processor forces the termination of the current chain, sends an FMH1 with suspend DS to the logical unit, and posts the SNA manager which attaches the message processor.

Interruption Of Data Inbound. Interruption of an inbound processor receiving card data is accepted anytime when the logical unit has a message to send.

The interruption must be indicated to the inbound processor by an inbound FMH1 with suspend DS. The suspended inbound processor will then post the SNA manager which will attach a second inbound processor to receive messages. These messages are treated as commands. Upon reception of an FMH1 with resume DS, the interrupting inbound processor will detach, the suspended inbound processor will resume the DS, and normal inbound flow can continue.

Inbound interruptions for outbound data are not supported.

Protocols

Half-duplex, flip-flop mode protocols are employed. Only one data stream at a time is allowed per session and contention is resolved by the use of SNA brackets.

Termination

Session Termination. The termination of a session is requested by the remote operator either by issuing the logoff request through VTAM, or by submitting a SIGNOFF command (from card or via the console) in the inbound data flow. The logoff request may be an unconditional logoff in which VTAM breaks the session and notifies POWER/VS through the LOSTERM exit. If the remote operator issues a conditional logoff VTAM notifies POWER/VS also through the LOSTERM exit, but VTAM does not break the session. The SIGNOFF command is passed via the normal inbound data stream directly to POWER/VS where it is handled as a conditional logoff request for all sessions of a given workstation.

The work station may logoff any individual session within the MLU concept. The logoff may be conditional or unconditional. The SIGNOFF command causes logoff of all sessions of the work station conditionally.

POWER/VS handles the unconditional logoff as an emergency stop which means that the termination routines are entered without checking any internal job boundary state. In this case the current reader job entry will not be added to the queue. The conditional logoff will be interpreted as a request for an orderly deactivation of the current session. In this case the termination routines will be entered only at job boundary time, when processing of the current job entry is completed.

After the active processors have been terminated, either normally or abnormally, the SNA manager activates the logoff processor which sends a message to the work station and finally issues a CLSDST to terminate the session. In an MLU environment a SIGNOFF causes termination session-by-session at job boundary times.

Session termination can be causes by the central operator either by means of the PSTOP command or, in case of emergency, by issuing the VARY NET, INACT,I,ID=luname command. VTAM notifies POWER/VS in the LOSTERM exit. Because VTAM does not allow any IO request to be issued, POWER/VS handles this termination type similar to an unconditional logoff.

Application Termination. The central operator may cause RJE,SNA shut-down either through POWER/VS central operator commands (e.g., PSTOP or PEND) or through VTAM operator commands (e.g., HALT).

Refer to DOS/VS POWER/VS Installation Guide and Reference, GC33-6048.

RJE,SNA ROUTINES

Figure 2.24E briefly describes each of the routines used to support RJE,SNA. The Figure 2.24F further describes the control blocks and work areas used to aid execution. In addition, Figure 2.24H illustrates the scheme of chaining the control blocks.

An overview of the sequence of routine execution and events is provided by Figure 2.24G. This figure should be used along with Figure 2.24E to better understand the RJE,SNA architecture.

Routine	Called/ Attached by:	Returns to:	Function or Notes
IPW$$IB- Inbound Processor	IPW$$SN	IPW$$NU	Issues RECEIVE Specific request to VTAM to receive data and then deblocks the data for spooling by IPW$$LR. Processes remote operator commands: • START • STOP • FLUSH • RESTART • SETUP • GO • SIGNOFF and transfers all other commands to IPW$$CP for processing. Posts the outbound processor command following GO, or RESTART when intervention is required. Posts the SNA manager. Detaches itself.
IPW$$LF- Logoff Processor	IPW$$SN	IPW$$NU	Logs off a logical unit using the VTAM macros SESSIONC and CLSDST. Sends message "1V12I LOGOFF COMPLETED" to the remote terminal and then sends the central operator the message "1V11I REMOTE rrr LOGGED OFF FROM POWER ON luname." Posts the SNA manager. Detaches itself.
IPW$$LH Logon Processor No.1	IPW$$SN		Completes SNA control block construction (SUCB, LUCB and WACB). Checks LOGON request and LU BIND parameters for validity. Sets an indicator for IPW$$SN to attach logon processor No. 2 (IPW$$LN). Posts the SNA manager. Detaches itself.

Figure 2.24E. Description of RJE,SNA Routines (Part 1 of 3)

Routine	Called/ Attached by:	Returns to:	Function or Notes
IPW$$LN-Logon Processor No.2	IPW$$SN	IPW$$NU	Establishes SNA session and starts data traffic with VTAM macros CPNDST and SESSIONC. Prints message "1V09I REMOTE rrr LOGGED ON TO POWER ON luname" at central operator console and then queues the same message for the remote terminal to be sent by the message processor (IPW$$MP). Posts the SNA manager. Detaches itself.
IPW$$MP-Message Processor	IPW$$SN	IPW$$NU	Transmits messages in message queue using VTAM macro SEND. Detaches itself.
IPW$$OB-Outbound Processor	IPW$$SN as LSTn or PUN task	IPW$$NU	Obtains job output data from spool file and transmits data to the LU using the VTAM macro SEND. The following intermediate steps occur: • Obtain spool file through IPW$$LW. • Create Function Management Headers (FMH). • Compress and compact if required. • Pack data into request units (RU). Wait on GO posting from IPW$$IB if SETUP remote command issued. Post the SNA manager and detaches itself.
IPW$$OC-Outbound Compaction Manager	IPW$$OB	IPW$$OB	Creates and updates COCB(s) and loads compaction table phases into GETVIS area.
IPW$$SN-SNA Manager	IPW$$CP	IPW$$NU	Sets up following ECBS in the TCB of IPW$$SN: • VTAM RECEIVE any ECB • Work ECB for RJE,SNA posting of IPW$$SN. Attaches IPW$$SN-Segment SUBTASK which calls IPW$$VE-Segment INIT (see below). Issues VTAM RECEIVE macro. Prints central operation message "'V04I RJE,SNA STARTED" Waits on ECB posting.
IPW$$SN-Segment SUBTASK	IPW$$SN-INIT	DOS/VS	When called the first time at RJE,SNA start-up time, it calls the IPW$$VE-Segment INIT and enables communication through VTAM with SETLOGON macro. Then posts IPW$$SN ECB and POWER/VS master ECB, and waits on posting by IPW$$SN. At termination time, the VTAM macro SETLOGON QUIESCE is called to halt further LOGON requests.

Figure 2.24E. Description of RJE,SNA Routines (Part 2 of 3)

Routine	Called/ Attached by:	Returns to:	Function or Notes
IPW$$SN-Segment MAIN	Posted by: • VTAM due to: RECEIVE Any input VTAM exits • POWER/VS routines: IPW$$AQ IPW$$CP IPW$$IB IPW$$LH IPW$$LN IPW$$MS IPW$$OB	IPW$$NU	After VTAM posting due to SNA line activity, a RDR task is attached which causes IPW$$IB to execute. After posting from other POWER/VS routines, a scan of the work station control blocks (SUCBs) and logical unit control blocks (LUCBs) is made. If any found to be active, the appropriate processor tasks are attached: • LST or PUN tasks (IPW$$OB) • Messages (IPW$$MP) Then a loop back if made to wait cn further posting.
IPW$$SN-Segment TERM	IPW$$SN-MAIN	IPW$$NU	Frees certain work areas and control blocks. Prints message "1V05I RJE,SNA TERMINATING." Detaches IPW$$SN task.
IPW$$VE-Segment LOGON	VTAM	VTAM	Creates and chains control blocks, used at logon time. Posts IPW$$SN work ECB and POWER/VS master ECB.
IPW$$VE-Segment INIT	IPW$$SN-SUBTASK	IPW$$SN-SUBTASK	Inserts addresses of VTAM exits in the ACB exit list.
IPW$$VE-Segment DFASY	VTAM	VTAM	If request to interrupt data flow, then the signal received indicator is set in the LUCB of the LU. If request to shut down, then stop session indicator is set. Posts IPW$$SN work ECB and POWER/VS master ECB.
IPW$$VE-Segment TPEND	VTAM	VTAM	Sets SNA stop code in SNCB. Post IPW$$SN work ECB and POWER/VS master ECB.
IPW$$VE-Segment LOSTERM	VTAM	VTAM	Sets on the stop session indicator in the LUCB of the LU. Posts IPW$$SN work ECB and POWER/VS master ECB.

Figure 2.24E. Description of RJE,SNA Routines (Part 3 of 3)

RJE,SNA Control Blocks/ Work Areas	When		Created by:	Stored at:	Purpose/General Contents
	Created:	Freed:			
CAT (Control Address Table)	At POWER/VS Initialization	At POWER/VS Termination	IPW$$IR	Real storage area	Pointers to modules and major control blocks.
SNCB (SNA Control Block)	At POWER/VS Initialization	At POWER/VS Termination	IPW$$IR	Fixed real storage area	RJE,SNA control block.
COCB (Compaction Control Block)	At time of first compaction table usage	At RJE,SNA Termination (IPW$$SN)	IPW$$OC	GETVIS area	Contains compaction table names, pointers and status. One COCB is generated for each 64 default tables. First is loaded with a default table name specified in PRMT macro.
LRCB (Logon Request Control Block)	At first LU LOGON	When (last) LOGON processed	IPW$$VE	GETVIS area	Used for LOGON processing. Consists of a header plus LRUB's.
LUCB (LU Control Block)	Space reserved at logon of first LU of work station	At work station logoff time (IPW$$LF)	IPW$$LH	GETVIS area	Contains information required for LU session, e.g. variable BIND parameters. Created by copying the logon LUCB ontothe One is created for each LU logged on.
Logon LUCB	At LOGON of first LU of work station	At RJE,SNA Termination (IPW$$SN)	IPW$$VE-Segment INIT	GETVIS area	Contains information required for LU session. Used as work area during logon processing.
SUCB (SNA unit control block)	Space reserved At LOGON of first LU of work station	At LOGOFF of last LU of work station	IPW$$LH	GETVIS area	Contains information pertaining to the work station of two types: a) General information, for example: REMID, SESSLIM. b) Device information for LSTn, PUN, RDR and Console, for example: class. Created by copying the LOGON SUCB onto the reserved SUCB area. One is created for each work station logged on.

Figure 2.24F. Description of RJE,SNA Control Blocks and Work Areas (Part 1 of 2)

RJE,SNA Control Blocks/ Work Areas	When Created:	Created Freed:	Stored by:	at:	Purpose/General Contents
Logon SUCB	At logon of first LU of work station	At RJE,SNA Termination	IPW$$VE-Segment INIT	GETVIS area	Used as a work area during LOGON processing.
WACB	• WACB for inbound interruption • WACB for LU inbound data • WACB for LU outbound data	• At LOGOFF of last session of work station • At LOGOFF of LU • At LSTn/ PUN task termination	IPW$$LH IPW$$LH IPW$$OB IPW$$MP	GETVIS area GETVIS area GETVIS area	Contains VTAM RPLs, RU buffers and some BIND information. One exists for each SUCB for inbound console data; another exists for each LUCB logged on for inbound; and another exists for each LUCB logged on during outbound data or message handling.
LOGON WACB	At LOGON of first LU of work station	At RJE,SNA Termination	IPW$$VE-Segment INIT	GETVIS area	Used as work area for LOGON processing
RMCB (Remote Control Block)	POWER/VS Initialization	POWER/VS Termination	IPW$$IR	GETVIS area	Contains information from the PRMT macro.
LRUB (LOGON Request Unit Control Block)	At LOGON time	After LOGON processing by IPW$$LH	IPW$$VE-LOGON Exit	GETVIS area	Contains LOGON request information.

Figure 2.24F. Description of RJE,SNA Control Blocks and Work Areas (Part 2 of 2)

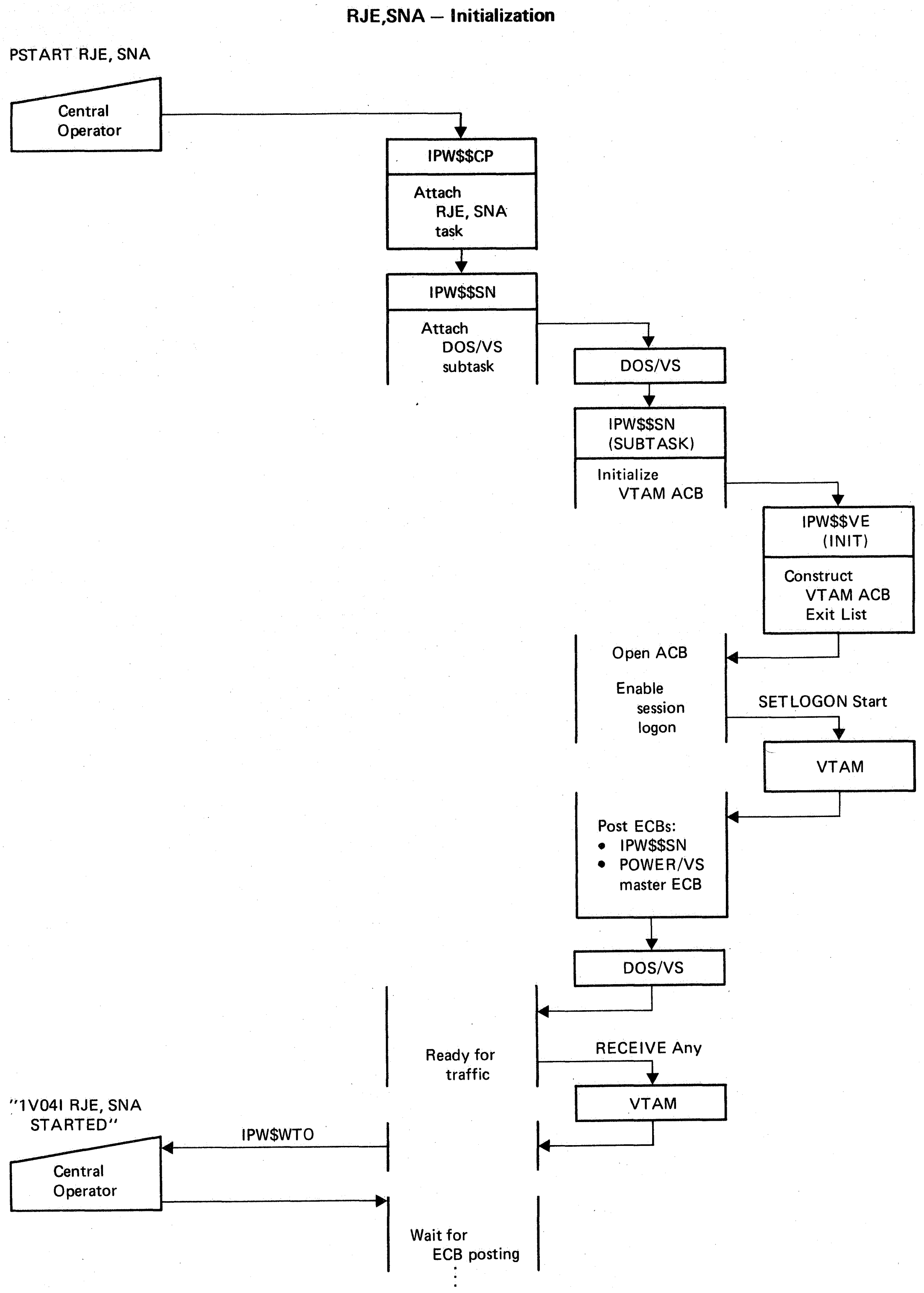


Figure 2.24G. RJE,SNA Execution Flow (Part 1 of 9)

RSE,SNA — Logon

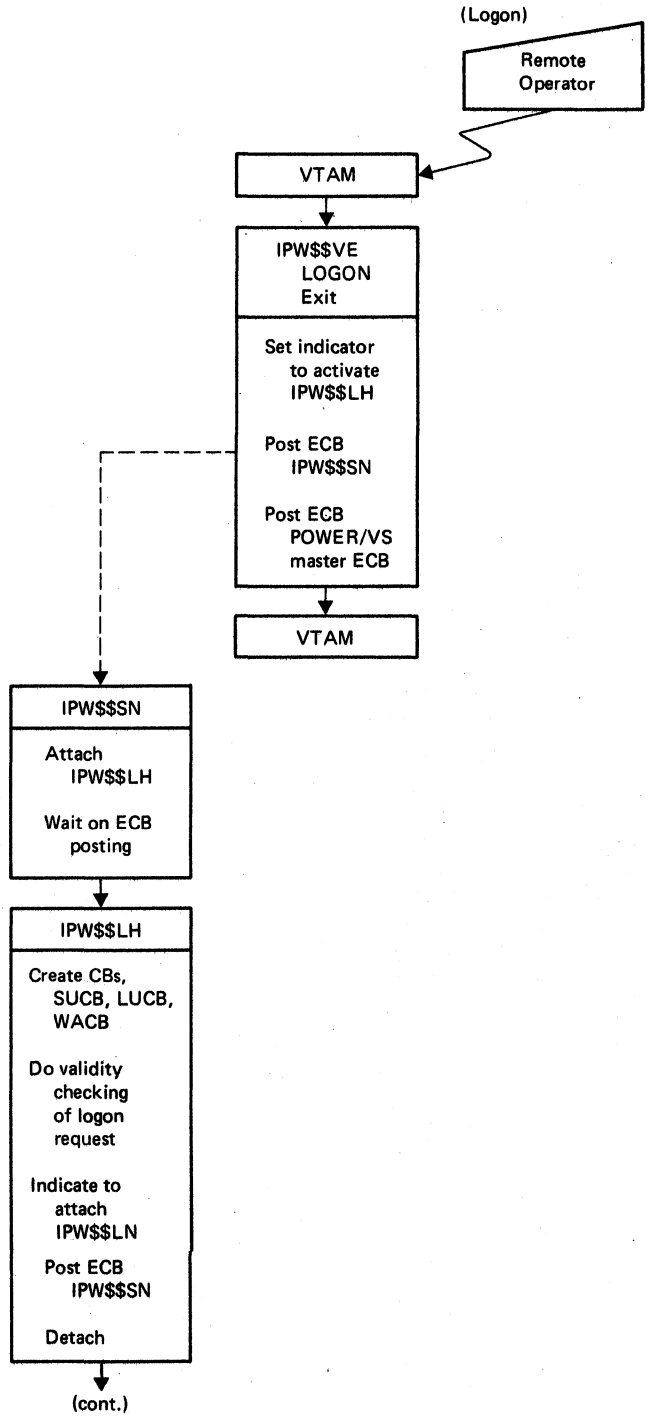

(Logon)
Remote Operator
VTAM
IPW$$VE LOGON Exit
Set indicator to activate IPW$$LH
Post ECB IPW$$SN
Post ECB POWER/VS master ECB
VTAM
IPW$$SN
Attach IPW$$LH
Wait on ECB posting
IPW$$LH
Create CBs, SUCB, LUCB, WACB
Do validity checking of logon request
Indicate to attach IPW$$LN
Post ECB IPW$$SN
Detach
(cont.)

Figure 2.24G. RJE,SNA Execution Flow (Part 2 of 9)

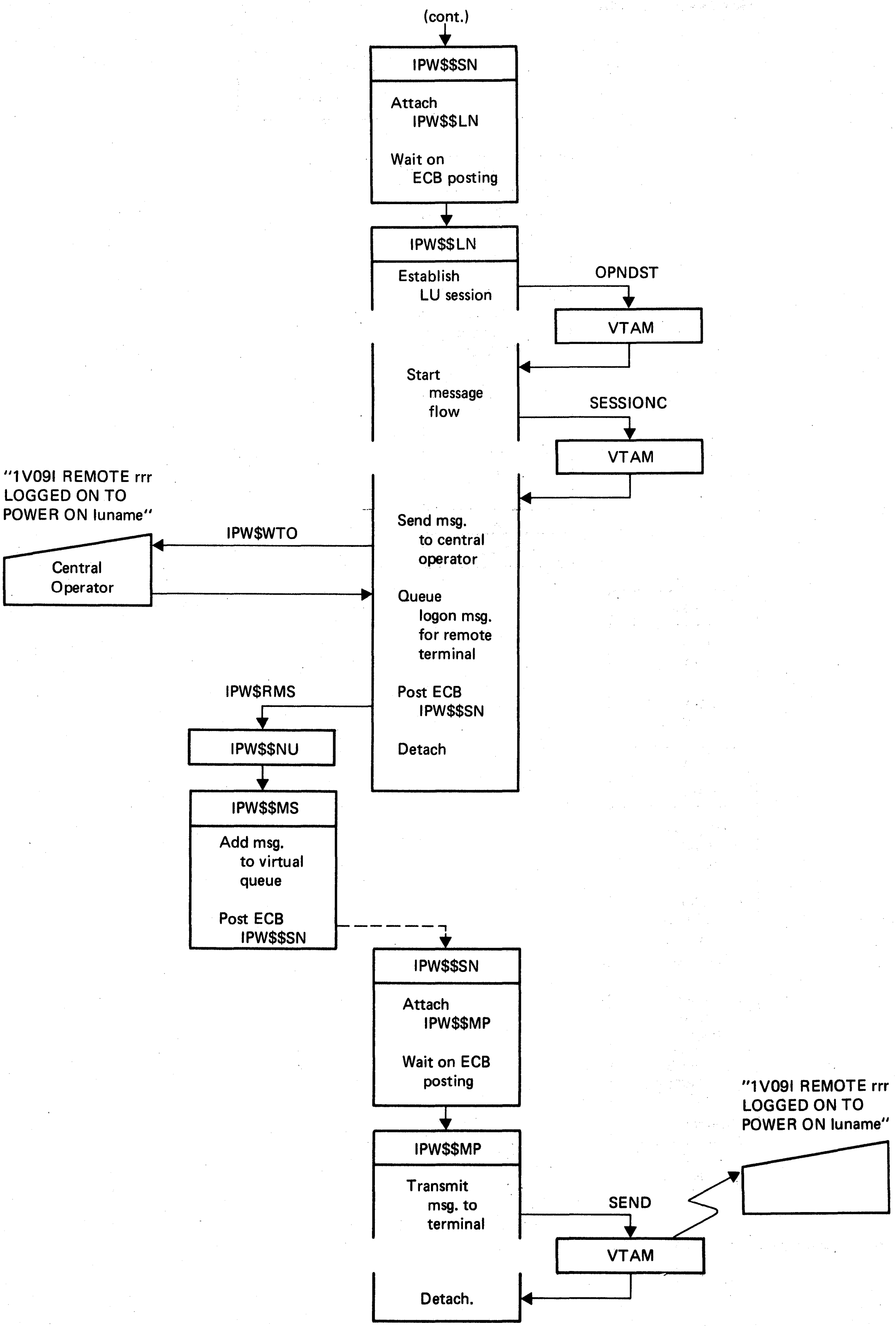


Figure 2.24G. RJE,SNA Execution Flow (Part 3 of 9)

RJE,SNA — Receive Console/Card Reader Data

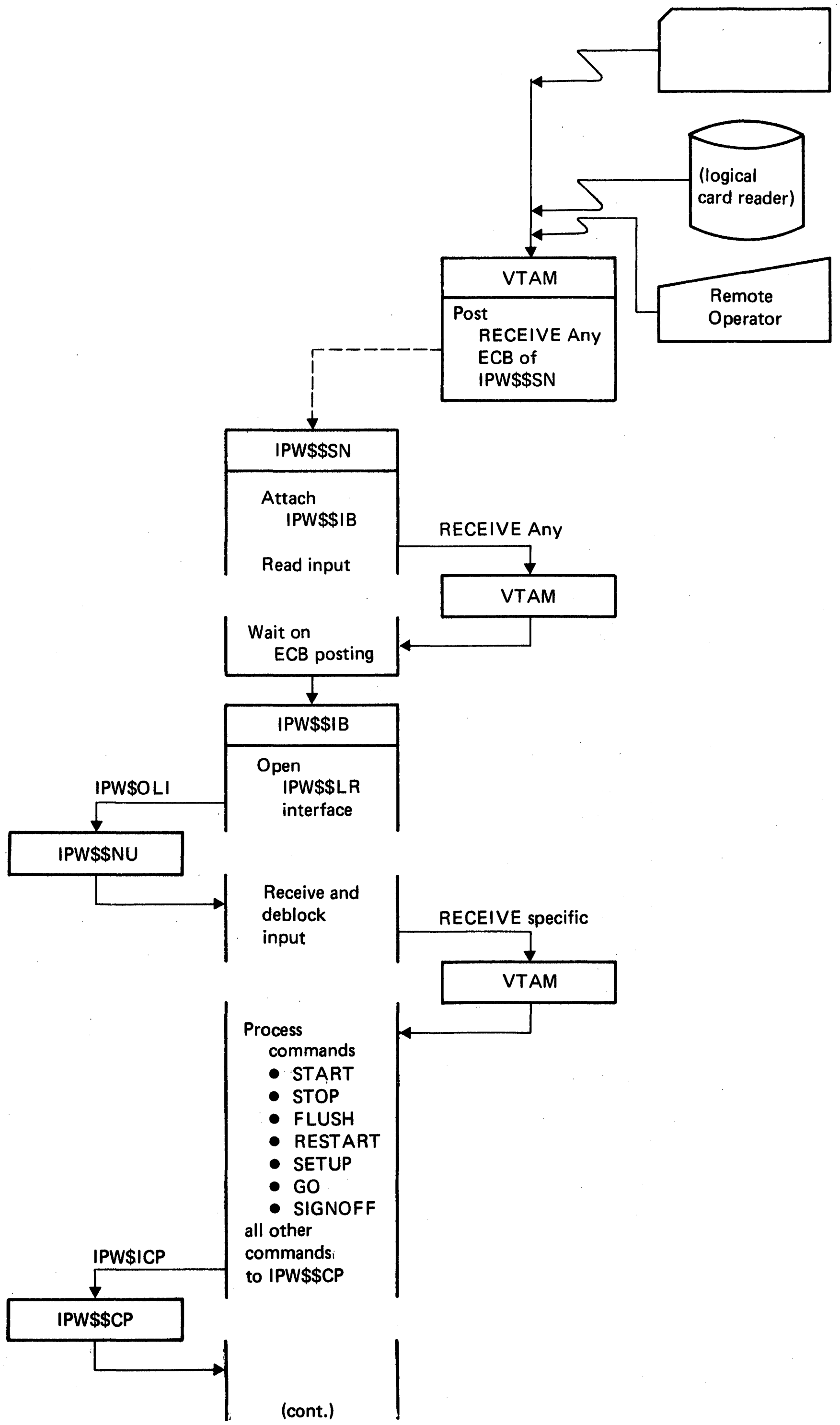


Figure 2.24G. RJE,SNA Execution Flow (Part 4 of 9)

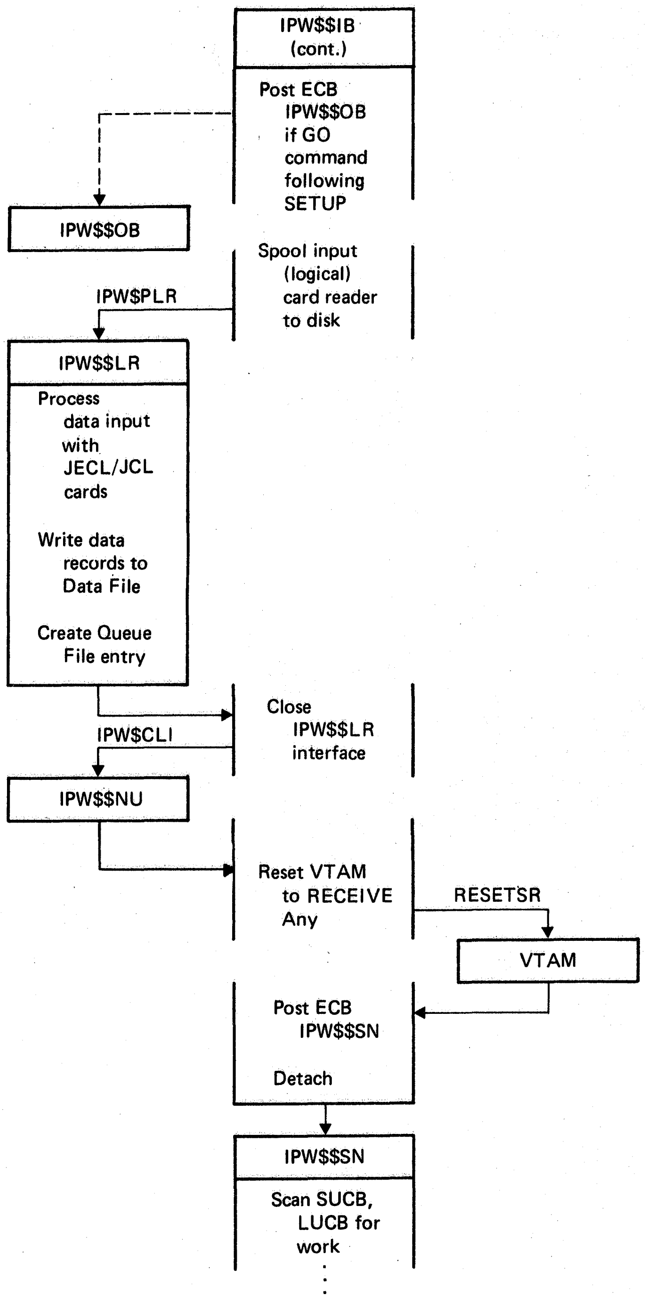

IPW$$IB (cont.)
Post ECB IPW$$OB if GO command following SETUP
IPW$$OB
Spool input (logical) card reader to disk
IPW$PLR
IPW$$LR
Process data input with JECL/JCL cards
Write data records to Data File
Create Queue File entry
Close IPW$$LR interface
IPW$CLI
IPW$$NU
Reset VTAM to RECEIVE Any
RESETSR
VTAM
Post ECB IPW$$SN
Detach
IPW$$SN
Scan SUCB, LUCB for work

Figure 2.24G. RJE,SNA Execution Flow (Part 5 of 9)

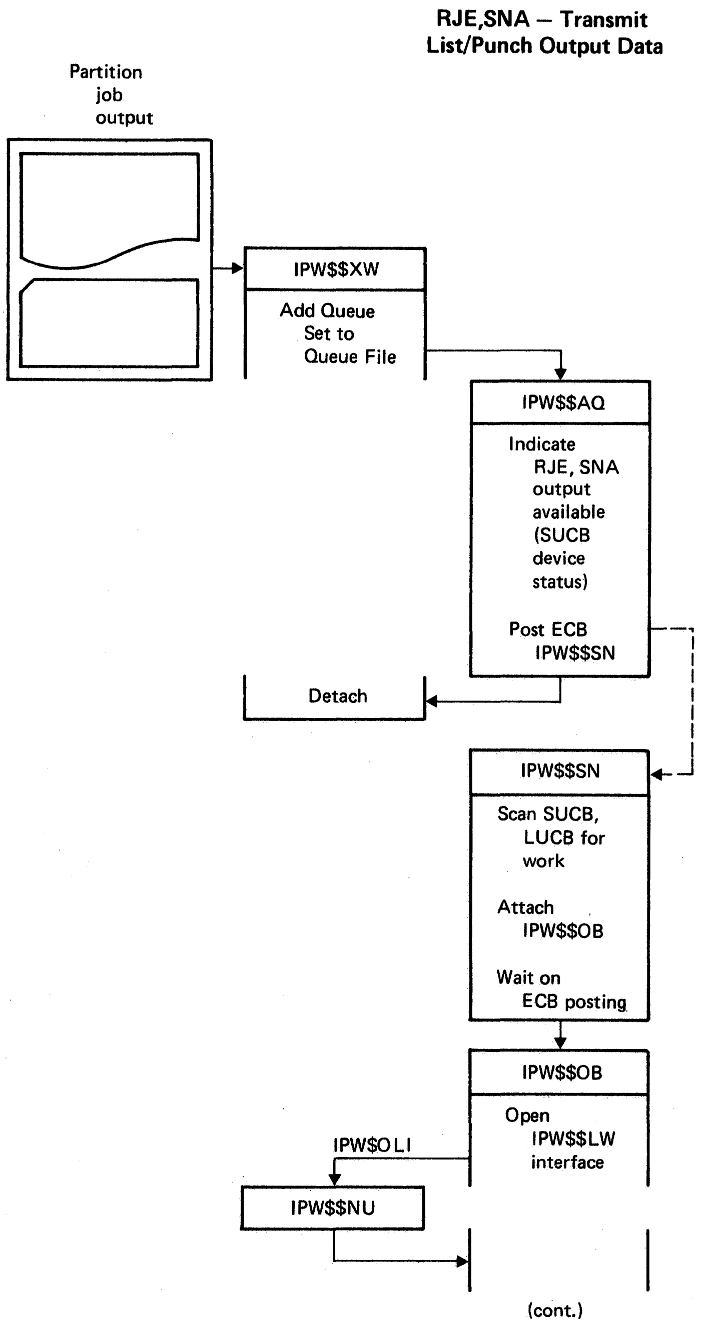


Figure 2.24G. RJE,SNA Execution Flow (Part 6 of 9)

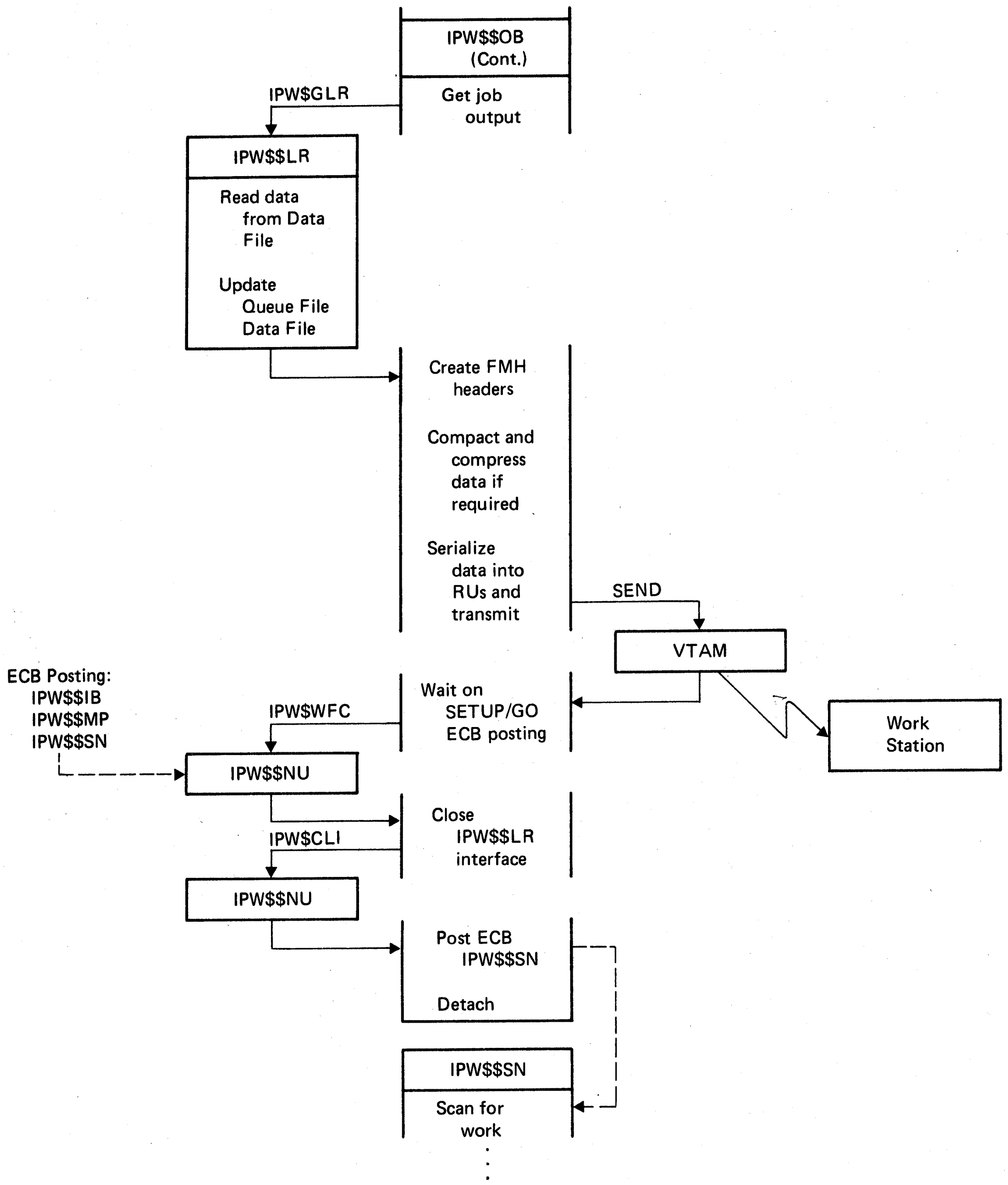


Figure 2.24G. RJE,SNA Execution Flow (Part 7 of 9)

RJE,SNA – Logoff

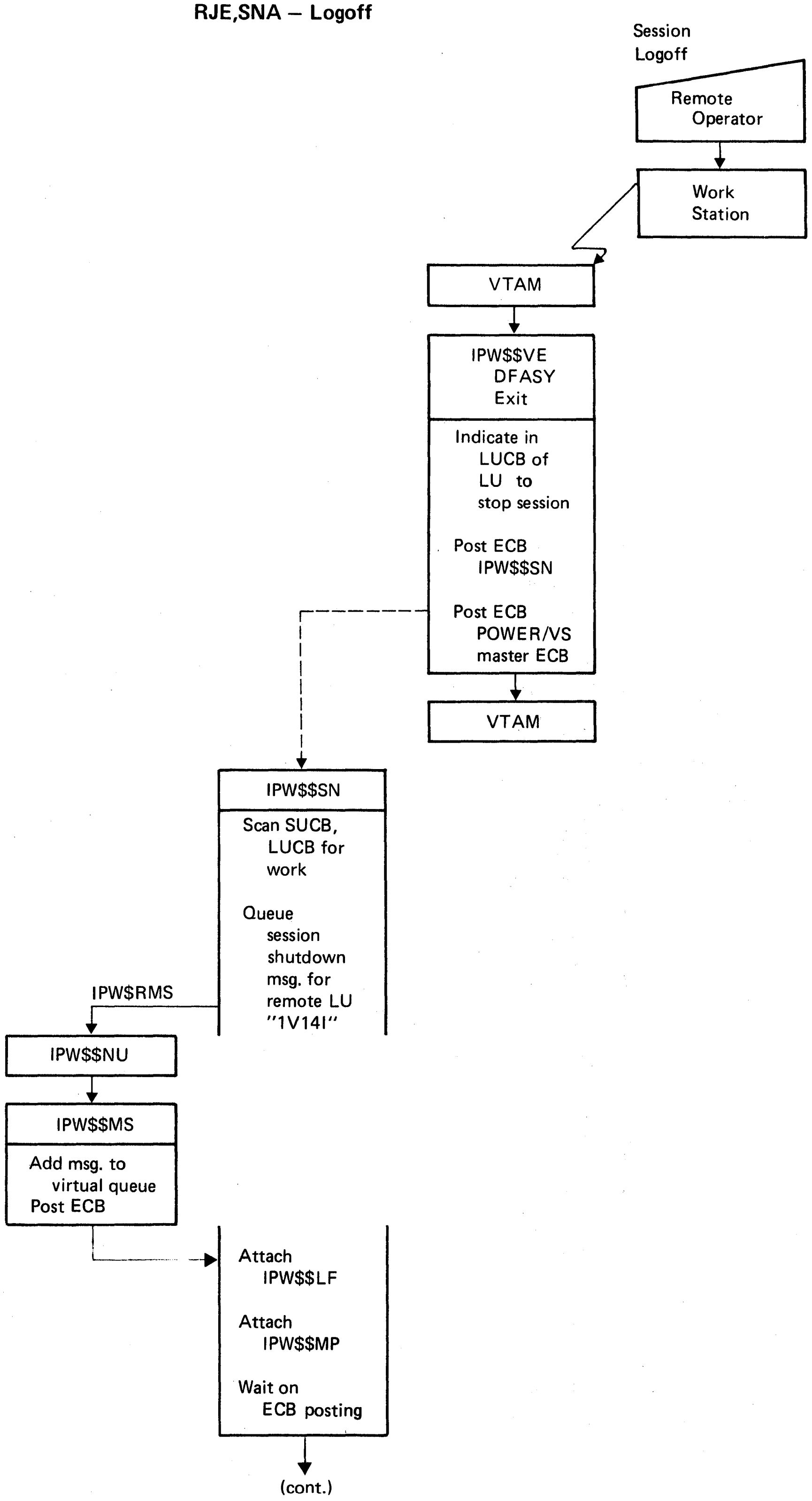


Figure 2.24G. RJE,SNA Execution Flow (Part 8 of 9)

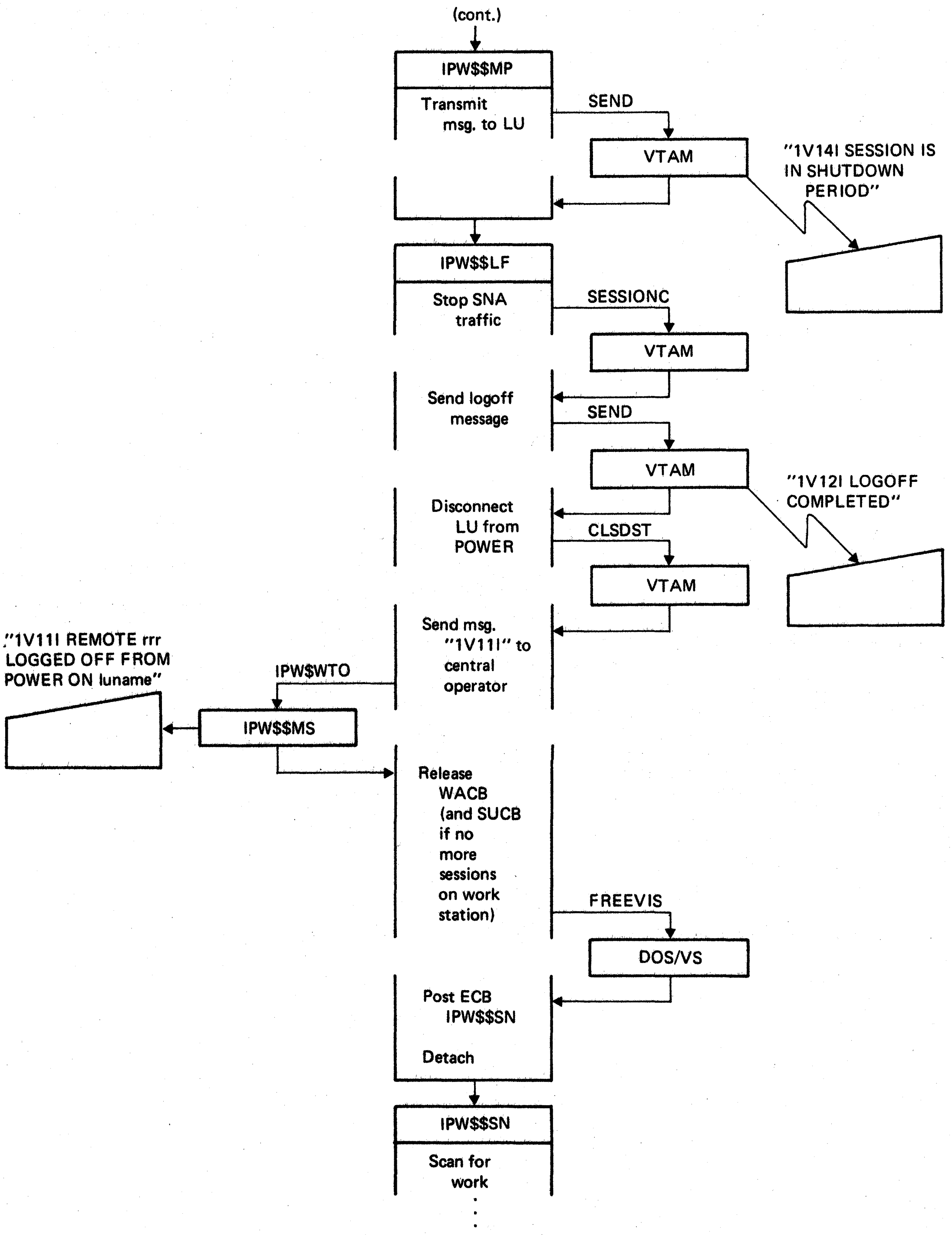


Figure 2.24G. RJE,SNA Execution Flow (Part 9 of 9)

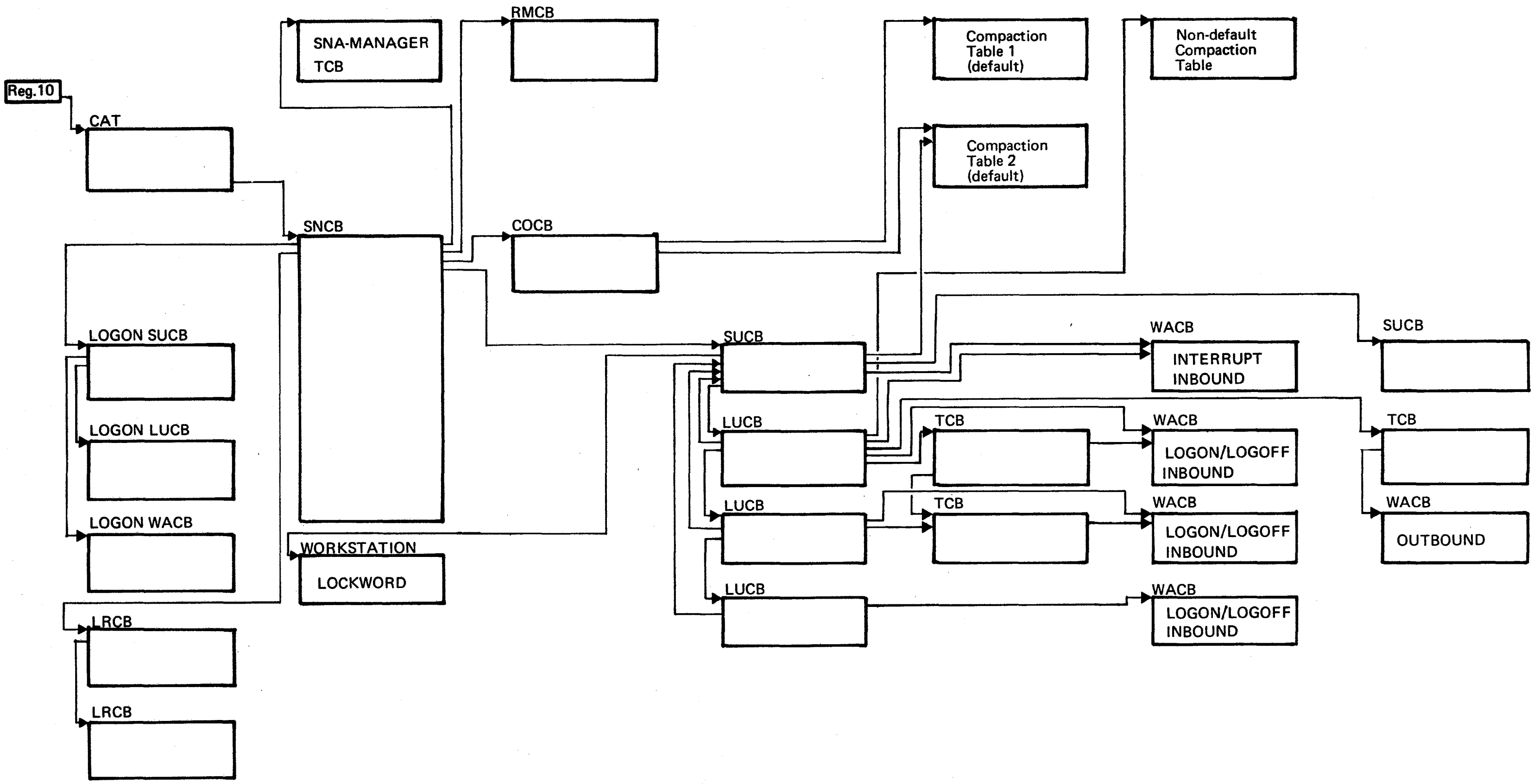


Figure 2.24H. RJE,SNA Control Block and Work Area Chaining

Appendages

PAGE FAULT APPENDAGE

If a page fault occurs, normally the partition is placed in a wait state, until the processing of the page fault is completed.

When a page fault overlap appendage linkage is established, the partition remains dispatchable in order to enable selection of another private task (within the partition) under control of its private multitasking routine.

The appendage routine for the POWER/VS partition is entered from the page manager routine in the supervisor on two conditions:

- the partition sustains a page fault (pre-processor)
- handling of a page fault is completed (post-processor).

The page fault pre-processor will take the following actions:

1. Save the task status, address of next instruction in PSW, and general registers (taken from the partition save area), in the TCB, because of page handling overlap by the supervisor later on.
2. Simulate a IPW$WFP macro instruction (put TCB in P state). This action is transparent to the task management routine.
3. Change the address of the next sequential instruction in the PSW to the entry of POWER/VS task management, because of page handling overlap by supervisor later on.
4. Queue the page fault request within an internal queue (in TCB chain), unless no page fault is being currently handled for the POWER/VS partition.
5. If no page fault is currently handled the request is returned directly to the page manager routine in the supervisor. If a page fault is currently handled, a request of zero is returned to the page manager.

The page fault post-processor will take the following actions:

1. Post the task, for which the page fault handling is completed, dispatchable (in TCB reset P state).
2. Post the partition dispatchable (in PIB), since the partition may be SVC 7 bound if all tasks were waiting.
3. Clear page fault request.
4. Examine the internal page fault queue (in TCB chain).

   If another page fault is found, it is passed to the page manager routine in the supervisor. If no other request is found, a page request of zero is returned to the page manager.

Note: The page fault currently handled for this partition and the address of the related TCB are saved in the appendage routine itself.

ATTENTION INTERFACE APPENDAGE

Refer to Initiation of the permanent command processor in "Command Processor".

RJE,BSC CHANNEL END APPENDAGE

During POWER/VS initialization a modification is made to the PIB of the POWER/VS partition in order to allow for a channel end appendage used for all RJE,BSC I/O operations. All RJE,BSC CCBs contain the address of the same channel end appendage routine, which is located in the POWER/VS nucleus in real storage.

The appendage routine gets control from the DOS/VS I/O interrupt handler whenever an interrupt is received from an RJE,BSC device. It then performs the following functions:

1. It moves bytes 1 through 3 of the CSW to the BCA (Buffer Control Area). These bytes contain the address of the last executed CCW plus 8, which is normally moved to the CCS (bytes 13-15) by the DOS/VS I/O interrupt handler. This move is not performed, however, when these bytes contain the address of a channel appendage routine, as in this case.

2. It queues the BCA to a BCA chain that will be processed by the line manager.

3. It activates the line manager.

Control is then returned to the next sequential instruction in the DOS/VS supervisor.

HOT READER APPENDAGE

The supervisor passes control to this appendage whenever an unsollicited device end interrupt for a unit record device is recognized.

The reader TCBs are scanned on cuu number to locate the task concerned with the interrupt. If the matching task is inactive, it is posted dispatchable. The POWER/VS partition is set dispatchable in PIB. In all other cases, no action is taken.

SVC 0 APPENDAGE

When the SVC 0 trap for a spooling request has located the appropriate task list entry in the partition control block, this appendage is entered.

The following actions are taken:

- The address of the CCB is stored in the task list entry of the partition control block.

- The related execution processor task ECB is posted to let the task simulate the request.

- The POWER/VS master ECB is posted (in CAT) after selection later on.

- The PIB flag for the POWER/VS partition is set dispatchable.

- The SIO table in the Job Accounting Interface Partition table (if existing) is updated.

Then control is returned to the supervisor.

SVC 90/91 APPENDAGE

The supervisor passes control to this appendage whenever an SVC 90 or SVC 91 interrupt is recognized. The address of the account information is stored in the reader entry of the partition control block. The execution reader task ECB is posted. The POWER/VS partition is set dispatchable for task selection by the supervisor.

Appendage Summary

Event	Appendage	Task Selection Action	Control Blocks
Page fault occurred	Page Fault (pre-processor)	Place current task in wait state, reenter task selection.	TCB
Page fault completed	Page Fault (post-processor)	Make task dispatchable, activate partition.	TCB
Attention interrupt	Attention Interface	Make CP task dispatchable, activate partition.	TCB, CPB
Unsollicited device end	Hot Reader	Set RDR task dispatchable, activate partition.	TCB
RJE channel end	Channel End	Set LM task dispatchable, activate partition.	TCB, BCA
SVC 0 intercepted	SVC 0	Set XR/XW task dispatchable, activate partition.	TCB, PDB
SVC 90 interrupt SVC 91 interrupt	SVC 90/SVC 91	Set XR task dispatchable, activate partition.	TCB, PDB

Section 3: Program Organization

The program organization of DOS/VS POWER/VS is described in DOS/VS POWER/VS Logic Part 2, SY33-8577, which manual contains detailed descriptions of the POWER/VS phases.

Section 4: Directory

The purpose of this section is to establish relationships between program identifier names (phase names, module names, control section names, and segment names) and between these names and Section 5 of this manual and the charts in DOS/VS POWER/VS Logic Part 2. The directory enables you to:

- Determine the type of name of any program identifier (phase, module, control section, macro, or segment).
- Determine the phase with which that name is associated.
- Determine the module with which that name is associated.
- If the name is a phase, locate the appropriate chart in DOS/VS POWER/VS Logic Part 2.
- If the name is a linkage macro, determine the invoked phase and its chart in DOS/VS POWER/VS Logic Part 2.
- If the name is a definition macro (control block, or data block), locate the matching figure in Section 5.

A reference list of messages is also included in this section. It relates a message with the issuing phase.

NAME LIST

Name	Type	Phase	Module	Reference Chart (Logic Manual - Part 2)
ACCB	Storage descriptor of control block. See Figure 5.24			
AIS	SEGMENT	IPW$$NU	IPW$$NU	BB
AQCS	SEGMENT	IPW$$AQ	IPW$$QM	
AQCS	CSECT	IPW$$AQ	IPW$$QM	
ASCS	SEGMENT	IPW$$AS	IPW$$DM	
ASCS	CSECT	IPW$$AS	IPW$$DM	
ASWS	Storage descriptor of control block. See Figure 5.48			
CASA	SEGMENT	IPW$$SA	IPW$$AM	
CAT	SEGMENT	IPW$$NU	IPW$$NU	
CAT	Storage descriptor of control block. See Figure 5.4			
CESG	SEGMENT	IPW$$NU	IPW$$NU	BC
COCB	Storage descriptor of control block. See Figure 5.34A			
CPB	Storage descriptor of control block. See Figure 5.13			
CPCS	SEGMENT	IPW$$CP	IPW$$CM	
CPCS	CSECT	IPW$$CP	IPW$$CM	
DMB	SEGMENT	IPW$$NU	IPW$$NU	
DMB	Storage descriptor of control block. See Figure 5.8			
DMS	SEGMENT	IPW$$NU	IPW$$NU	AE
DQCS	SEGMENT	IPW$$DQ	IPW$$QM	
DQCS	CSECT	IPW$$DQ	IPW$$QM	
FQCS	CSECT	IPW$$FQ	IPW$$QM	
FQCS	SEGMENT	IPW$$FQ	IPW$$QM	
GACS	SEGMENT	IPW$$GA	IPW$$AM	
GACS	CSECT	IPW$$GA	IPW$$AM	
GDCS	SEGMENT	IPW$$GD	IPW$$DM	
GDCS	CSECT	IPW$$GD	IPW$$DM	
GNB	Storage descriptor of control block. See Figure 5.41			
HRS	SEGMENT	IPW$$NU	IPW$$NU	BD
IBCS	CSECT	IPW$$IB	IPW$$IB	
ICCS	SEGMENT	IPW$$IC	IPW$$DM	
ICCS	CSECT	IPW$$IC	IPW$$DM	
IPW$xxx see macro list below				
IPW$$AM	MODULE			
IPW$$AQ	PHASE	IPW$$AQ	IPW$$QM	DB
IPW$$AS	PHASE	IPW$$AS	IPW$$DM	
IPW$$CM	MODULE			
IPW$$CP	PHASE	IPW$$CP	IPW$$CM	CP
IPW$$DD	MODULE	IPW$$DD	IPW$$DD	
IPW$$DD	PHASE	IPW$$DD	IPW$$DD	NA
IPW$$DM	MODULE			
IPW$$DQ	PHASE	IPW$$DQ	IPW$$QM	DD
IPW$$ER	MODULE			HC
IPW$$FQ	PHASE	IPW$$FQ	IPW$$QM	DE
IPW$$GA	PHASE	IPW$$GA	IPW$$AM	FB
IPW$$GD	PHASE	IPW$$GD	IPW$$DM	EB
IPW$$IB	MODULE			MG
IPW$$IC	PHASE	IPW$$IC	IPW$$DM	GD
IPW$$IR	MODULE			
IPW$$I1	PHASE	IPW$$I1	IPW$$IR	LA
IPW$$I2	PHASE	IPW$$I2	IPW$$IR	LB
IPW$$LF	MODULE			ME
IPW$$LH	MODULE			MK
IPW$$LN	MODULE			MJ
IPW$$LR	PHASE	IPW$$LR	IPW$$RR	HB
IPW$$LU	PHASE	IPW$$LU	IPW$$DM	GC
IPW$$LW	PHASE	IPW$$LW	IPW$$WR	JC
IPW$$MD	MODULE			
IPW$$MP	MODULE			MF
IPW$$MS	PHASE	IPW$$MS	IPW$$MS	MB
IPW$$MS	MODULE			
IPW$$NQ	PHASE	IPW$$NQ	IPW$$QM	DC
IPW$$NU	MODULE	IPW$$NU	IPW$$NU	
IPW$$NU	PHASE	IPW$$NU	IPW$$NU	AA-BF

Name	Type	Phase	Module	Reference Chart (Logical Manual - Part 2)
IPW$$OB	MODULE			MH
IPW$$OC	Module			ML
IPW$$OE	MODULE			GF
IPW$$OT	MODULE	IPW$$OT	IPW$$DM	GG
IPW$$PA	PHASE	IPW$$PA	IPW$$AM	FA
IPW$$PD	PHASE	IPW$$PD	IPW$$DM	EA
IPW$$PL	PHASE	IPW$$PL	IPW$$WR	JA
IPW$$PP	PHASE	IPW$$PP	IPW$$WR	JB
IPW$$PR	PHASE	IPW$$PR	IPW$$RR	HA
IPW$$PS	PHASE	IPW$$PS	IPW$$PS	CQ
IPW$$PS	MODULE			
IPW$$QM	MODULE			
IPW$$RQ	PHASE	IPW$$RQ	IPW$$QM	DA
IPW$$RR	MODULE			
IPW$$SA	PHASE	IPW$$SA	IPW$$AM	CR
IPW$$SC	PHASE	IPW$$SC	IPW$$RR	GA
IPW$$SL	PHASE	IPW$$SL	IPW$$DM	GE
IPW$$SM	PHASE	IPW$$SM	IPW$$SM	PA
IPW$$SM	MODULE			
IPW$$SN	MODULE			MC
IPW$$TM	PHASE	IPW$$TM	IPW$$TM	MA
IPW$$TM	MODULE			
IPW$$TR	PHASE	IPW$$TR	IPW$$TR	LC
IPW$$TR	MODULE			
IPW$$VE	MODULE			MV
IPW$$WR	MODULE			
IPW$$XJ	PHASE	IPW$$XJ	IPW$$XP	GB
IPW$$XP	MODULE			
IPW$$XR	PHASE	IPW$$XR	IPW$$XP	KA
IPW$$XW	PHASE	IPW$$XW	IPW$$XP	KB
I1CS	SEGMENT	IPW$$I1	IPW$$IR	
I1CS	CSECT	IPW$$I1	IPW$$IR	
I2CS	SEGMENT	IPW$$I2	IPW$$IR	
I2CS	CSECT	IPW$$I2	IPW$$IR	
LFCS	CSECT	IPW$$LF	IPW$$LF	
LHCS	CSECT	IPW$$LH	IPW$$LH	MK
LNCS	CSECT	IPW$$SN	IPW$$LN	
LMCS	SEGMENT	IPW$$TM	IPW$$TM	
LRCB	Storage descriptor of control block. See Figure 5.34B			
LRCS	SEGMENT	IPW$$TM	IPW$$TM	RR
LRCS	CSECT	IPW$$LR	IPW$$RR	
LUCB	Storage descriptor of control block. See Figure 5.34C			
LUCS	SEGMENT	IPW$$LU	IPW$$DM	
LUCS	CSECT	IPW$$LU	IPW$$DM	
LWCS	CSECT	IPW$$LW	IPW$$WR	
LWCS	SEGMENT	IPW$$LW	IPW$$WR	
MCB	Storage descriptor of control block. See Figure 5.17			
MMB	SEGMENT	IPW$$NU	IPW$$NU	AD
MMB	Storage descriptor of control block. See Figure 5.7			
MPCS	CSECT	IPW$$MP	IPW$$MP	
MSCB	Storage descriptor of control block. See Figure 5.39.			
MSCS	SEGMENT	IPW$$MS	IPW$$MS	
MSCS	CSECT	IPW$$MS	IPW$$MS	
NQCS	CSECT	IPW$$NQ	IPW$$QM	
NQCS	SEGMENT	IPW$$NQ	IPW$$QM	
OBCS	CSECT	IPW$$OB	IPW$$OB	
OCCS	CSECT	IPW$$OC	IPW$$OC	ML
OTCS	CSECT	IPW$$OT	IPW$$DM	GG
PACCOUNT	SEGMENT	IPW$$CP	IPW$$CM	
PACCOUNT	CSECT	IPW$$CP	IPW$$CM	
PACS	SEGMENT	IPW$$PA	IPW$$AM	
PACS	CSECT	IPW$$PA	IPW$$AM	
PALTER	SEGMENT	IPW$$CP	IPW$$CM	
PALTER	CSECT	IPW$$CP	IPW$$CM	
PBRDCST	SEGMENT	IPW$$CP	IPW$$CM	

Name	Type	Phase	Module	Reference Chart (Logic Manual - Part 2)
PBRDCST	CSECT	IPW$$CP	IPW$$CM	
PCANCEL	SEGMENT	IPW$$CP	IPW$$CM	
PCANCEL	CSECT	IPW$$CP	IPW$$CM	
PDB	Storage descriptor of control block. See Figure 5.21			
PDCS	SEGMENT	IPW$$PD	IPW$$DM	
PDCS	PHASE	IPW$$PD	IPW$$DM	
PDELETE	SEGMENT	IPW$$CP	IPW$$CM	
PDELETE	CSECT	IPW$$CP	IPW$$CM	
PDISPLAY	SEGMENT	IPW$$CP	IPW$$CM	
PDISPLAY	CSECT	IPW$$CP	IPW$$CM	
PEND	SEGMENT	IPW$$CP	IPW$$CM	
PEND	CSECT	IPW$$CP	IPW$$CM	
PFLUSH	SEGMENT	IPW$$CP	IPW$$CM	
PFLUSH	CSECT	IPW$$CP	IPW$$CM	
PFS	SEGMENT	IPW$$NU	IPW$$NU	BA
PGO	SEGMENT	IPW$$CP	IPW$$CM	
PGO	CSECT	IPW$$CP	IPW$$CM	
PINQUIRE	SEGMENT	IPW$$CP	IPW$$CM	
PINQUIRE	CSECT	IPW$$CP	IPW$$CM	
PLCS	SEGMENT	IPW$$PL	IPW$$WR	
PLCS	CSECT	IPW$$PL	IPW$$WR	
POWERMS	CSECT	IPW$$MS	IPW$$MS	
POWERTM	CSECT	IPW$$TM	IPW$$TM	
PPCS	CSECT	IPW$$PP	IPW$$WR	
PPCS	SEGMENT	IPW$$PP	IPW$$WR	
PPCS	CSECT	IPW$$NU	IPW$$NU	
PRCS	CSECT	IPW$$PR	IPW$$RR	
PRCS	SEGMENT	IPW$$PR	IPW$$RR	
PRELEASE	SEGMENT	IPW$$CP	IPW$$CM	
PRELEASE	CSECT	IPW$$CP	IPW$$CM	
PRESTART	SEGMENT	IPW$$CP	IPW$$CM	
PRESTART	CSECT	IPW$$CP	IPW$$CM	
PSCS	SEGMENT	IPW$$PS	IPW$$PS	
PSCS	CSECT	IPW$$PS	IPW$$PS	
PSETUP	SEGMENT	IPW$$CP	IPW$$CM	
PSETUP	CSECT	IPW$$CP	IPW$$CM	
PSTART	SEGMENT	IPW$$CP	IPW$$CM	
PSTART	CSECT	IPW$$CP	IPW$$CM	
PSTOP	SEGMENT	IPW$$CP	IPW$$CM	
PSTOP	CSECT	IPW$$CP	IPW$$CM	
RMCB	Storage descriptor of control block. See Figure 5.36			
RMS	SEGMENT	IPW$$NU	IPW$$NU	AB
RQCS	SEGMENT	IPW$$RQ	IPW$$QM	
RQCS	CSECT	IPW$$RQ	IPW$$QM	
SACS	CSECT	IPW$$SA	IPW$$AM	
SCB	SEGMENT	IPW$$NU	IPW$$NU	AC
SCB	Storage descriptor of control block. See Figure 5.6			
SCCS	SEGMENT	IPW$$SC	IPW$$RR	
SCCS	CSECT	IPW$$SC	IPW$$RR	
SLCS	CSECT	IPW$$SL	IPW$$DM	
SLSD	SEGMENT	IPW$$SL	IPW$$DM	
SMCS	CSECT	IPW$$SM	IPW$$SM	
SNCB	Storage descriptor of control block. See Figure 5.35			
SNCS	CSECT	IPW$$SN	IPW$$SN	
SUCB	Storage descriptor of control block. See Figure 5.37			
SVS	SEGMENT	IPW$$NU	IPW$$NU	BE, BF
TACS	SEGMENT	IPW$$TM	IPW$$TM	
TACS	CSECT	IPW$$TM	IPW$$TM	
TBB	Storage descriptor of control block. See Figure 5.18			
TCB	Storage descriptor of control block. See Figures 5.9 - 5.12			
TPS	SEGMENT	IPW$$NU	IPW$$NU	AF

Name	Type	Phase	Module	Reference Chart (Logic Manual - Part 2)
TRCS	CSECT	IPW$$TR	IPW$$TR	
TRCS	SEGMENT	IPW$$TR	IPW$$TR	
TRS	SEGMENT	IPW$$NU	IPW$$NU	AG
VECS	CSECT	IPW$$VE	IPW$$VE	MV
WACB	Storage descriptor of control block. See Figure 5.38			
WCB	SEGMENT	IPW$$NU	IPW$$NU	AA
WCB	Storage descriptor of control block. See Figure 5.5			
XJCS	SEGMENT	IPW$$XJ	IPW$$XP	
XJCS	CSECT	IPW$$XJ	IPW$$XP	
XRCS	SEGMENT	IPW$$XR	IPW$$XP	
XRCS	CSECT	IPW$$XR	IPW$$XP	
XWCS	SEGMENT	IPW$$XW	IPW$$XP	
XWCS	CSECT	IPW$$XW	IPW$$XP	

MACRO LIST

Macro	Type	Phase	Chart (Part 2)	Figure
IPW$AQS	LINKAGE	IPW$$AQ	DB	
IPW$ATT	LINKAGE	IPW$$NU	AA	
IPW$CAF	LINKAGE	IPW$$GA	FB	
IPW$CLI	LINKAGE	Note 1		
IPW$CTT	LINKAGE	IPW$$NU	AF	
IPW$DAB	DEFINITION			
IPW$DAC	DEFINITION			5.24
IPW$DBC	DEFINITION			5.33
IPW$DCB	DEFINITION			5.42
IPW$DCO	DEFINITION			5.34A
IPW$DCT	DEFINITION			5.8
IPW$DCW	DEFINITION			5.43
IPW$DDE	DEFINITION			
IPW$DDR	DEFINITION			5.16
IPW$DDV	DEFINITION			
IPW$DET	LINKAGE	IPW$$NU	AA	
IPW$DGN	DEFINITION			5.41
IPW$DLC	DEFINITION			5.32
IPW$DLR	DEFINITION			5.34B
IPW$DLU	DEFINITION			5.34C
IPW$DMC	DEFINITION			5.17
IPW$DMM	DEFINITION			5.7
IPW$DMS	DEFINITION			5.39
IPW$DPA	DEFINITION			5.1
IPW$DPC	DEFINITION			5.19
IPW$DPD	DEFINITION			5.21
IPW$DPW	DEFINITION			5.15
IPW$DQC	DEFINITION			5.8
IPW$DQR	DEFINITION			5.22
IPW$DQS	LINKAGE	IPW$$DQ	DD	
IPW$DRM	DEFINITION			5.36
IPW$DSA	DEFINITION			5.13
IPW$DSC	DEFINITION			5.6
IPW$DSL	DEFINITION			5.23
IPW$DSN	DEFINITION			5.35
IPW$DSR	DEFINITION			
IPW$DSV	DEFINITION			5.13
IPW$DSU	DEFINITION			5.37
IPW$DTB	DEFINITION			5.18
IPW$DTC	DEFINITION			5.9
IPW$DTE	DEFINITION			
IPW$DWA	DEFINITION			5.38
IPW$EQU	DEFINITION			
IPW$FCH	LINKAGE	IPW$$NU	AA	
IPW$FQS	LINKAGE	IPW$$FQ	DE	
IPW$GAR	LINKAGE	IPW$$GA	FB	
IPW$GDR	LINKAGE	IPW$$GD	EB	
IPW$GLR	LINKAGE	Note 1		
IPW$GQS	LINKAGE	IPW$$NQ	DC	
IPW$GSL	LINKAGE	IPW$$SL	GE	
IPW$IAS	LINKAGE	IPW$$AS		
IPW$ICP	LINKAGE	IPW$$IC	GD	
IPW$IOC	LINKAGE	IPW$$OB	MH	
IPW$OAF	LINKAGE	IPW$$GA	FB	
IPW$OEF	LINKAGE	IPW$$OE	GF	
IPW$OLI	LINKAGE	Note 1		
IPW$OTP	LINKAGE	IPW$$OT	GG	
IPW$PAR	LINKAGE	IPW$$PA	FA	
IPW$PDR	LINKAGE	IPW$$PD	EA	
IPW$PLR	LINKAGE	Note 1		
IPW$RDC	LINKAGE	IPW$$NU	AG	
IPW$RDD	LINKAGE	IPW$$NU	AE	

Macro	Type	Phase	Chart (Part 2)	Figure
IPW$RDQ	LINKAGE	IPW$$NU	AE	
IPW$RDT	LINKAGE	IPW$$NU	AF	
IPW$RET	LINKAGE	Note 2		
IPW$RLR	LINKAGE	IPW$$NU	AB	
IPW$RLW	LINKAGE	IPW$$NU	AC	

Macro	Type	Phase	Chart (Part 2)	Figure
IPW$RMS	LINKAGE	IPW$$NU	AD	
IPW$RQS	LINKAGE	IPW$$RQ	DA	
IPW$RSR	LINKAGE	IPW$$NU	AB	
IPW$RSW	LINKAGE	IPW$$NU	AC	
IPW$SAV	LINKAGE	Note 2		
IPW$SRJ	LINKAGE	IPW$$SC	GA	
IPW$SXJ	LINKAGE	IPW$$XJ	GB	
IPW$ULP	LINKAGE	IPW$$LU	GC	
IPW$VDA	LINKAGE	IPW$$NU	AH	
IPW$WFC	LINKAGE	IPW$$NU	AA	
IPW$WFD	LINKAGE	IPW$$NU	AA	
IPW$WFI	LINKAGE	IPW$$NU	AA	
IPW$WFL	LINKAGE	IPW$$NU	AA	
IPW$WFM	LINKAGE	IPW$$NU	AA	
IPW$WFO	LINKAGE	IPW$$NU	AA	
IPW$WFQ	LINKAGE	IPW$$NU	AA	
IPW$WFS	LINKAGE	IPW$$NU	AA	
IPW$WTD	LINKAGE	IPW$$NU	AE	
IPW$WTO	LINKAGE	IPW$$NU	AD	
IPW$WTQ	LINKAGE	IPW$$NU	AE	
IPW$WTR	LINKAGE	IPW$$NU	AD	
IPW$WTT	LINKAGE	IPW$$NU	AF	

Note 1: Refer to 'Interface Linkage' under 'Linkage Conventions' in DOS/VS POWER/VS Logic Part 2.

Note 2: Refer to 'Function Linkage' under 'Linkage Conventions' in DOS/VS POWER/VS Logic Part 2.

MESSAGE LIST

Note: Whenever the field "task" appears in a message, the same text in the code listing appears as "ttttt" which is replaced by the task-ID before printing.

Message		Phase	Chart (Part 2)
1Q01I	POWER/VS CANNOT RUN IN REAL MODE	IPW$$I1	LA02
1Q02I	POWER/VS CANNOT RUN AS A SUBTASK	IPW$$I1	LA01
1Q03I	INSUFFICIENT REAL STORAGE AVAILABLE; 10K REQUIRED	IPW$$I1	LA04
	INSUFFICIENT REAL STORAGE AVAILABLE; nnK REQUIRED	IPW$$I2	LB20
1Q04I	QUEUE FILE MISMATCH	IPW$$I2	LB05
1Q05I	PAGEABLE AREA nnnK TOO SMALL	IPW$$I1	LA05
1Q06I	SYSLOG NOT ASSIGNED TO CONSOLE	IPW$$I1	LA01
1Q07I	INVALID LOGICAL UNIT	IPW$$I2	LB22
1Q08I	SUPERVISOR WITHOUT POWER/VS SUPPORT	IPW$$I1	LA02
1Q09I	TRACKGROUP CHANGED TO nn	IPW$$I2	LB07
1Q10I	SUPERVISOR WITHOUT ACCOUNTING SUPPORT	IPW$$I1	LA02
1Q11D	FORMAT QUEUES=	IPW$$I2	LB04
1Q12I	POWER/VS INITIATION COMPLETED	IPW$$I2	LB14
1Q13I	ERRONEOUS AUTOSTART CARD(S) READ	IPW$$I2	LB04
1Q14I	NO MATCHING PUB FOR cuu	IPW$$I1	LA02, LA04
1Q15I	PHASE phasename NOT FOUND	IPW$$I1	LA05
1Q16I	INVALID PUN/LST ROUTING FOR remid	IPW$$I1	LA03, LA04
1Q17I	QUEUE FILE TOO SMALL	IPW$$I2	LB07
1Q18I	TOO MANY EXTENTS IJDFILE	IPW$$I2	LB18
1Q19I	INVALID EXTENT IJDFILE	IPW$$I2	LB07,LB18
1Q20I	AUTOSTART IN PROGRESS	IPW$$I2	LB13
1Q21I	POWER/VS HAS BEEN TERMINATED	IPW$$I2	LB14
1Q22I	POWER/VS ALREADY ACTIVE	IPW$$I1	LA01
1Q23I	LTA CANCEL	IPW$$I2	LB04
1Q24I	QUEUE FILE CHAIN ERROR	IPW$$I2	LB06
1Q25I	partition IN STOP STATE	IPW$$IR	LB15
1Q26I	GETVIS AREA TOO SMALL	IPW$$I1	LA03
1Q27I	UNABLE TO INITIALIZE SPOOL MANAGEMENT	IPW$$I2	LB13
1Q30D	ABNORMAL POWER/VS TERMINATION. PRINTER=	$$BPOWIN	LD01
1Q31I	MORE THAN 80% FULL ACCOUNT FILE (IJAFILE) ******	IPW$$PA	FA02
1Q32I	NO MORE ACCOUNT FILE (IJAFILE) SPACE FOR task, cuu	IPW$$PA	FA03
1Q33I	STOPPED task,cuu	IPW$$TR	LC14
		IPW$$XR	KA04
1Q34I	task WAITING FOR WORK ON cuu	IPW$$LW	JC01
		IPW$$PR	HA05
		IPW$$XR	KA07
1Q35A	EOF ON cuu	IPW$$PR	HA05
1Q36I	task CANCELED DUE TO I/O ERROR ON cuu	IPW$$PR	
1Q37I	JECL CARD INCORRECT NEAR COLUMN xxx	IPW$$LR	HB16, HB17, HB18-HB19
		IPW$$RQ	DA01
1Q38I	NO DASD SPACE AVAILABLE FOR ttt, cuu	IPW$$PD	EA02
		IPW$$RQ	DA01
1Q39I	JOB jjjjjjjj FLUSHED BY THE OPERATOR OR POWER/VS	IPW$$LW	JC07, JC08
1Q40A	ON task FORMS ffff NEEDED FOR jjjjjjjj nnnnnn	IPW$$LW	JC03
1Q41I	WRONG PRINTER/PUNCH FOR jjjjjjjj cuu	IPW$$LW	JC02
1Q42I	PAGE/CARD COUNT EXCEEDS END QUEUE ENTRY FOR cuu	IPW$$LW	JC10
1Q43I	END OF SPOOL TAPE task,cuu	IPW$$NQ	DC01
1Q44I	BOOK s.bbbbbbbb NOT FOUND	IPW$$SL	GE08
1Q45I	SLI STATEMENT NOT SUPPORTED	IPW$$XJ	GB15
1Q46I	DISP FORCED TO D FOR jjjjjjjj nnnnnn	IPW$$XJ	GB07
1Q47I	task jjjjjjjj nnnnnn FROM rrr uuuuuuuuuuuuuuuu	IPW$$XJ	GB05
1Q48D	NO MATCHING SPOOL DEVICE task	IPW$$XJ	GB06
1Q49D	INVALID DELIMITER task	IPW$$XJ	GB04,GB08, GB14
1Q50D	UNKNOWN KEYWORD task	IPW$$XJ	GB10

Message		Phase	Chart (Part 2)
1Q51D	INVALID DISP PARAMETER task	IPW$$XJ	GB14
	INVALID CLASS PARAMETER task	IPW$$XJ	GB08,GB10
	INVALID FNO PARAMETER task	IPW$$XJ	GB08,GB10
	INVALID COPY PARAMETER task	IPW$$XJ	GB09,GB11
	INVALID TADDR PARAMETER task	IPW$$XJ	GB09,GB11
	INVALID LTAB PARAMETER task	IPW$$XJ	GB13
	INVALID REMOTE PARAMETER task	IPW$$XJ	GB10
	INVALID JSEP PARAMETER task	IPW$$XJ	GB10
	INVALID RBM PARAMETER task	IPW$$XJ	GB09,GB11
	INVALID RBS PARAMETER task	IPW$$XJ	GB11
	INVALID PRI PARAMETER task	IPW$$XJ	GB12
	INVALID UCS PARAMETER task	IPW$$XJ	GB12
	INVALID FCB PARAMETER task	IPW$$XJ	GB12
	INVALID LST PARAMETER task	IPW$$XJ	GB12
	INVALID PUN PARAMETER task	IPW$$XJ	GB12
	INVALID JNM PARAMETER task	IPW$$XJ	GB04
	INVALID USER PARAMETER task	IPW$$XJ	GB04
	INVALID COPYG PARAMETER task		
	INVALID FLASH PARAMETER task		
	INVALID BURST PARAMETER task		
	INVALID DFLT PARAMETER task		
	INVALID CHARS PARAMETER task		
	INVALID MODIFY PARAMETER task		
1Q52I	OUTPUT LIMIT EXCEEDED FOR jjjjjjjj nnnnnn task	IPW$$XW	KB14
1Q53I	OUTPUT SEGMENTED FOR jjjjjjjj nnnnnn task	IPW$$XW	KB06
1Q54I	FCB/UCS ERROR FOR jjjjjjjj nnnnnn task	IPW$$XW	KB02
1Q55D	SPECIFY TAPE ADDRESS FOR jjjjjjjj nnnnnn task	IPW$$OT	GG01
1Q56I	INVALID TAPE ADDRESS/MODE SET	IPW$$OT	GG01
1Q57A	WRITE RING REQUIRED ON cuu task	IPW$$OT	GG02
1Q58A	MOUNT TAPE ON cuu FOR jjjjjjjj nnnnnn task	IPW$$OT	GG02
1Q59I	task,cuu WAITING FOR REAL STORAGE	IPW$$NU	AC01
1Q60I	OPEN FAILURE ON PACCOUNT OUTPUT DEVICE	IPW$$SA	CR09
1Q61I	IRRECOVERABLE I/O ERROR ON xFILE n cuu	IPW$$TR	LC03,LC04, LC08
	IRRECOVERABLE I/O ERROR ON SYSTEM/PVT SSL	IPW$$TR	LC04
	IRRECOVERABLE I/O ERROR ON cuu	IPW$$TR	LC11,LC13, LC14,LC16
	IRRECOVERABLE I/O ERROR ON PACCOUNT OUTPUT DEVICE	IPW$$TR	LC15
1Q61D	IRRECOVERABLE I/O ERROR ON task,cuu - CANCEL, RESTART, OR IGNORE?	IPW$$WR	JA02.1
1Q62I	CANCEL FOR POWER/VS IGNORED	DOS/VS	Transient
1Q63I	IRRECOVERABLE I/O ERROR IN QUEUE MASTER REC - cuu	IPW$$TR	LC04
1Q64I	JOB jjjjjjjj task SET DELETED	IPW$$TR	LC05,LC07, LC08
1Q65I	UNKNOWN task SET DELETED	IPW$$TR	LC06
1Q66I	ACCOUNT FILE KEPT	IPW$$TR	LC12
1Q67I	FREE SET NOT ACCESSIBLE	IPW$$TR	LC05
1Q68I	SEGMENTATION FORCED FOR jjjjjjjj nnnnnn task cuu	IPW$$TR	LC11
1Q69I	DEFAULT OPTIONS TAKEN FOR jjjjjjjj nnnnnn task cuu	IPW$$TR	LC11
1Q70I	TASK FAILURE, STOPPED task	IPW$$TR	LC13
1Q71I	task,cuu TERMINATED	IPW$$TR	LC14
1Q72I	PACCOUNT TERMINATED	IPW$$TR	LC12
1Q73I	STATUS DISPLAY TERMINATED	IPW$$TR	LC13
1Q74I	ACCOUNT SUPPORT FUNCTIONS TERMINATED	IPW$$TR	LC04
1Q75I	MULTIPLE TERMINATION OF TASK, POWER/VS TERMINATED	IPW$$TR	LC01
1Q76I	POWER/VS CANNOT CONTINUE	IPW$$TR	LC04
1Q77I	INVALID SPOOL TAPE task,cuu	IPW$$NQ	DC01
1Q78I	NO REAL STORAGE AVAILABLE FOR task,cuu	IPW$$SA	CR01,CR12
1Q79I	ACCOUNT FILE SAVED	IPW$$SA	CR04,CR08, CR11
1Q80I	ACCOUNT FILE ERASED	IPW$$SA	CR05
1Q81I	'filename' EXTENT TOO SMALL, COMMAND NOT EXECUTED	IPW$$SA	CR11
1Q82I	I/O ERROR DURING task, POWER/VS TERMINATED	IPW$$TR	LC01
1Q83I	ACCOUNT FILE NOTHING TO SAVE	IPW$$SA	CR01
1Q87I	cuu EOJ ADDED jobname,jobnumber	IPW$$ER	HC08

Message		Phase	Chart (Part 2)
1Q88I	INVALID 3540 UNIT FOR task	IPW$$XR	KA08
1Q89I	PROGRAM OUT OF SEQUENCE FOR task	IPW$$XR	KA10
1Q90I	* $$ RDR STATEMENT NOT ALLOWED --JOB FLUSHED	IPW$$LR	HB19
1Q91D	cuu NON-COMPATIBLE DISKETTE FOR rdr,cuu		
	VOL1 LABEL ERROR OR NOT FOUND R=	IPW$$OE	GF03
	NON-BASIC EXCHANGE DISKETTE TYPE R=	IPW$$OE	GF03
	NON-BASIC EXCHANGE ffffffff FILE R=	IPW$$OE	GF04
	ffffffff BYPASS REQUIRED R=	IPW$$OE	GF04
	LABEL STANDARD VERSION VIOLATION R=	IPW$$OE	GF03
	MULTIVOLUME IND NOT C, L, OR BLANK R=	IPW$$OE	GF04
	ffffffff END XTNT BELOW BEGIN XTNT R=	IPW$$OE	GF06
	VOL SEQ NO. ERR HDR1 LABEL (nnnnn) R=	IPW$$OE	GF05
	BLOCKLENGTH ERR HDR1 LABEL (nnnnn) R=	IPW$$OE	GF05
	BEGINEXTENT ERR HDR1 LABEL (nnnnn) R=	IPW$$OE	GF05
	END EXTENT ERR HDR1 LABEL (nnnnn) R=	IPW$$OE	GF06
	END-OF-DATA ERR HDR1 LABEL (nnnnn) R=	IPW$$OE	GF06
	EOD ADDR BELOW BEGIN XTNT R=	IPW$$OE	GF06
1Q92D	cuu NO HDR1 FOR ffffffff, rdr,cuu R=	IPW$$OE	GF09
1Q93D	cuu SECURED VOLUME/FILE FOR rdr,cuu R=	IPW$$OE	GF09
1Q94D	cuu EXPECT VOL nn, NOT mm, rdr,cuu R=	IPW$$OE	GF09
1Q95D	cuu NON-VERIFIED ffffffff, rdr,cuu R=	IPW$$OE	GF10
1Q96I	cuu ffffffff IS EMPTY FILE FOR rdr,cuu	IPW$$OE	GF10
1Q97D	cuu PREMATURE LAST VOL FOR rdr,cuu R=	IPW$$OE	GF10
1Q98D	cuu ffffffff TOO MANY VOLS rdr,cuu R=	IPW$$OE	GF11
1Q9nD	NO PRECEEDING VOL, INCONSIST RESP R=	IPW$$OE	GF13
1Q9nD	INVALID RESPONSE R=	IPW$$OE	GF11,GF13
1QA0I	NO SUBTASK AVAILABLE FOR task, cuu		
1QA1I	SETPRT ROUTINE NOT FOUND IN SVA task, cuu		
1QA2I	CATASTROPHIC LOGIC ERROR OCCURRED rc		
1QA3I	SETPRT ERROR FOR jjjjjjjj nnnnn task, cuu		
1QA4I	OUTPUT PROCESSING STOPPED jjjjjjjj nnnnn task, cuu		
1QA5A	cuu SETUP REQUIRED jjjjjjjj -FORMS= -FLASH= -THREAD=		
1QA6I	NO STORAGE AVAILABLE FOR task, cuu		
1R02I	LINE cuu STOPPED	IPW$$TM	MA48
1R03I	TRANSM number, TIMEOUTS number, ERRORS number	IPW$$TM	MA25
1R10I	INVALID commandcode COMMAND	IPW$$TM	MA33
1R11I	INVALID STOP/SETUP COMMAND	IPW$$TM	MA31
1R12I	INVALID CLASS SPECIFICATION	IPW$$TM	MA30
1R13I	INVALID TASK SPECIFICATION	IPW$$TM	MA32
1R14I	EOF ON THE READER	IPW$$TM	MA02
1R15I	REMOTE remid SIGNED ON	IPW$$TM	MA29
1R16I	REMOTE remid SIGNED OFF	IPW$$TM	MA25
1R17I	LINE HAS BEEN STOPPED	IPW$$TM	MA25
1R18I	REMOTE remid FORCED TO SIGN OFF	IPW$$TM	MA25
1R19I	FIRST CARD MUST BE SIGNON COMMAND	IPW$$TM	MA02
1R20I	nnn MESSAGES DELETED	IPW$$MS	MB04
1R21I	SIGNON IGNORED. INVALID REMOTE-ID	IPW$$TM	MA27
1R22I	SIGNON IGNORED. INVALID PASSWORD	IPW$$TM	MA28
1R24I	xxxx COMMAND OUT OF SEQUENCE	IPW$$TM	MA32
1R30I	INVALID CCW-CCB ADDR = X'ccb addr' jobname jobnumber,cuu	IPW$$XP	KA07.1
1R33D	CORRECT FULL STATEMENT	IPW$$XJ	GB07
	NO VALID CORRECTION	IPW$$XP	GB07
	ERROR IN CONTINUATION CARD-CORRECT FULL STATEMENT	IPW$$XP	GB07
1R34I	cccc....ccc (52 characters)	IPW$$OT	GG03
1R35D	VOL1 LABEL ON task, cuu - ENTER DISK, TAPE, OR IGNORE	IPW$$OT	GG03
1R41I	INVALID SPECIFICATION FOR DISKETTE	IPW$$CP	CP21
1R42I	OPERAND 7 INCORRECT	IPW$$CP	CP20
1R43I	OPERAND 6 INCORRECT	IPW$$CP	CP20
1R44I	OPERAND 5 INCORRECT	IPW$$CP	CP20
1R45I	OPERAND 4 TOO LONG	IPW$$CP	CP20

Message		Phase	Chart (Part 2)
1R46I	status report	IPW$$PS	CP02
	class QUEUE NOTHING TO DISPLAY	IPW$$CP	CP39
	TIME IS xx:xx:xx, DATE IS xx/xx/xx	IPW$$CP	CP95
	NOTHING TO DISPLAY	IPW$$CP	CP39
1R47I	NO MESSAGES PENDING	IPW$$CP	CP38
1R48I	NO READER OR WRITER TASK CURRENTLY ACTIVE	IPW$$CP	CP37
1R49I	FREE RECORDS QUEUE FILE nnnnn	IPW$$CP	CP36
	ACCOUNT FILE nn% FULL	IPW$$CP	CP36
	NO ACCOUNTING SUPPORT	IPW$$CP	CP36
1R50D	task READER=	IPW$$CP	CP91
	task PRINTERS=	IPW$$CP	CP91
	task PUNCHES=	IPW$$CP	CP91

Message		Phase	Chart (Part 2)
1R51I	OPERAND 1 DESIGNATES NON-EXISTING TASK	IPW$$CP	CP55,CP62, CP63,CP66, CP67
	NON-EXISTING TASK DESIGNATED	IPW$$CP	CP67
IR52I	NO STATUS REPORT IN PROGRESS	IPW$$CP	CP58
	NO SECOND OPERAND ALLOWED FOR PSTOP PARTITION	IPW$$CP	CP57
	OPERAND 2 NEITHER DECIMAL NOR OMITTED	IPW$$CP	CP61
	OPERAND 2 NEITHER 'HOLD' NOR OMITTED	IPW$$CP	CP66
	NO VALID KEYWORD SPECIFIED	IPW$$CP	CP26
	OPERAND n IS NOT SPECIFIED AS KEYWORD	IPW$$CP	CP26
	OPERAND 1 INVALID DESTINATION	IPW$$CP	CP64
	OPERAND n IS INVALID KEYWORD	IPW$$CP	CP26
	OPERAND 1 MISSING OR INVALID	IPW$$CP	CP55,CP66, CP67
	OPERAND 1 OR 2 INVALID	IPW$$CP	CP64
	OPERAND 2 NO DEVICE ADDRESS	IPW$$CP	CP67
	OPERAND 1 NO VALID QUEUE	IPW$$CP	CP25,CP52, CP59
	OPERAND 1 NOT 'STATUS'	IPW$$CP	CP58
	INVALID NUMBER OF COPIES	IPW$$CP	CP30
	SECOND OPERAND INVALID	IPW$$CP	CP57
	OPERAND 3 NOT DECIMAL	IPW$$CP	CP52,CP59
	INVALID DISPOSITION	IPW$$CP	CP28
	INVALID REMOTE ID	IPW$$CP	CP31
	LAST OPERAND INVALID	IPW$$CP	CP72
	OPERAND 1 INVALID	IPW$$CP	CP35,CP44
	OPERAND 2 INVALID	IPW$$CP	CP25,CP52, CP53,CP56, CP59
	INVALID PRIORITY	IPW$$CP	CP27
	INVALID CLASS	IPW$$CP	CP29
	INVALID OUTPUT-CLASS	IPW$$CP	CP11
	BUFFER SPECS NOT 1 OR 2	IPW$$CP	CP22
	INVALID OPERAND	IPW$$CP	CP68
	DELIMITER NOT BLANK OR COMMA	IPW$$CP	CP70
	INVALID COMPACT NAME	IPW$$CP	CP25
	OPERAND 3 INVALID	IPW$$CP	CP61.1
	INVALID BUFFER SPECIFICATION	IPW$$CP	CP22.1
	OPERAND 4 IGNORED	IPW$$CP	CP22.1
1R53I	INVALID DENSITY	IPW$$CP	CP46
1R54I	CLASS x INVALID	IPW$$CP	CP88
1R55I	INVALID FILENAME	IPW$$CP	CP44,CP46
1R56I	cuu PROCESSING remid	IPW$$CP	CP69
	cuu NOT INITIATED	IPW$$CP	CP69
	cuu INACTIVE	IPW$$CP	CP69
	NO LOGICAL UNIT LOGGED ON	IPW$$CP	CP68
	luname PROCESSING remid	IPW$$CP	CP68
	luname NOT LOGGED ON	IPW$$CP	CP68
	luname LOGGED ON	IPW$$CP	CP68
	luname LOGGING ON	IPW$$CP	CP68
1R57I	COMMAND IGNORED, TASK IS AT JOB BOUNDARY	IPW$$CP	CP61,CP66
	RDREXIT FLUSH IGNORED, TASK IS AT JOB BOUNDARY	IPW$$LR	HB14
1R58I	DEVICE cuu NOT KNOWN	IPW$$CP	CP45,CP78
	DEVICE cuu IN USE	IPW$$CP	CP78
	DEVICE cuu IS DOWN	IPW$$CP	CP78
1R59I	OPERAND 2 IGNORED	IPW$$CP	CP50
1R60I	partition NOT SUPPORTED	IPW$$CP	CP07
1R61I	INVALID FOR WRITER-ONLY PARTITION	IPW$$CP	CP66
1R62I	INVALID RJE PASSWORD	IPW$$CP	CP23
1R63I	partition PRIORITY TOO HIGH	IPW$$CP	CP07,CP08
1R64I	SYSLST LUB NOT AVAILABLE	IPW$$CP	CP77
	SYSLST LUB NOT AVAILABLE task, cuu	IPW$$PL	
	NO FREE LUB AVAILABLE	IPW$$CP	CP77
	NO LUB AVAILABLE, DISP FORCED TO D jobname, jobnumber, partition		

Message		Phase	Chart (Part 2)
1R65I	RJE-BSC NOT SUPPORTED	IPW$$CP	CP90,CP64 CP68
	RJE,SNA ALREADY STARTED	IPW$$CP	CP24
	RJE/SNA NOT STARTED	IPW$$CP	CP56
	RJE,SNA NOT SUPPORTED	IPW$$CP	CP24,CP56
	RJE,SNA NOT ACTIVE	IPW$$CP	CP56
1R66I	cuu LIST WRITER TASK DOES NOT EXIST	IPW$$CP	CP65
	NO WRITER TASK SPECIFIED	IPW$$CP	CP62,CP63, CP66
1R67I	OPERAND 2 REDUCED TO 9999	IPW$$CP	CP61
	NUMBER OF PAGES REDUCED TO 99	IPW$$CP	CP65
1R68I	partition PARTITION NOT AVAILABLE	IPW$$CP	CP12
	xx IS THE POWER PARTITION	IPW$$CP	CP7
1R69I	NO ACCOUNTING SUPPORT	IPW$$CP	CP44
1R70I	NO DEVICE ADDRESS SPECIFIED	IPW$$CP	CP62
1R71I	OPERAND n IS NO VALID READER/PRINTER/PUNCH	IPW$$CP	CP92
1R72I	VIRTUAL partition SMALLER THAN 64K	IPW$$CP	CP09
1R73I	INVALID DEVICE TYPE FOR xxx	IPW$$CP	CP79
1R74I	OPERAND n INVALID PRINTER SPECIFICATION	IPW$$CP	CP48,CP50
	INVALID TAPE SPECIFICATION	IPW$$CP	CP45
	INVALID DEVICE SPECIFICATION	IPW$$CP	CP17,CP21
	NO PRINTER ADDRESS SPECIFIED	IPW$$CP	CP65
	INVALID LINE ADDRESS	IPW$$CP	CP23
1R75I	partition AUTOSTARTED	IPW$$CP	CP91
1R76I	NUMBER OF PAGES NOT DECIMAL	IPW$$CP	CP65
1R77I	TASK NOT WAITING FOR OPERATOR	IPW$$CP	CP65,CP67
1R78I	SHORT 'PEND' NO LONGER SUPPORTED	IPW$$CP	CP48
1R79I	ERRONEOUS AUTOSTART CARD(S) READ	IPW$$CP	CP92,CP94
1R80I	WARNING: CLASS SPECIFICATION IGNORED	IPW$$CP	CP16
1R81I	MESSAGE DOES NOT START WITH QUOTE	IPW$$CP	CP71
	MESSAGE TOO LONG OR NO CLOSING QUOTE	IPW$$CP	CP71
1R82I	'PSETUP' OR 'PRESTART' IN PROGRESS	IPW$$CP	CP61
1R83I	OPERAND NEITHER 'ALL' NOR LINE ADDRESS	IPW$$CP	CP68
1R84I	DELETION NOT ALLOWED OR IMPOSSIBLE	IPW$$CP	CP53
1R85I	COMMAND INVALID FOR REMOTE OPERATOR	IPW$$CP	CP44,CP48
1R86I	PLEASE SPECIFY DEVICES TO BE SPOOLED	IPW$$CP	CP91
1R87I	TOO MANY CLASSES, FIRST n PROCESSED	IPW$$CP	CP88
1R88I	NOTHING TO RELEASE	IPW$$CP	CP59
	NOTHING TO DELETE	IPW$$CP	CP52
	NOTHING TO ALTER	IPW$$CP	CP25
		IPW$$CP	CP25,CP52,
	OK	IPW$$CP	CP25,CP52, CP59
	JOB jjjjjjj nnnnnn CANNOT BE ALTERED	IPW$$CP	CP32
	JOB jobname jobnumber CANNOT BE ALTERED (PDIR)	IPW$$CP	CP32
1R89I	POWER/VS INITIATION NOT COMPLETE	IPW$$CP	CP48
1R90I	INVALID TASK SPECIFICATION, operand	IPW$$CP	CP04
1R91I	TOO MANY OPERANDS, FIRST n PROCESSED	IPW$$CP	CP24,CP70
1R92I	ALLUSER MESSAGE QUEUE IS FULL	IPW$$CP	CP64
1R93I	REMOTE remid CURRENTLY NOT SIGNED ON	IPW$$CP	CP64
	NO SESSION ESTABLISHED FOR xxxxxxxx	IPW$$CP	CP56
1R94I	INVALID DEVICE DUPLICATION	IPW$$CP	CP94
1R95I	LINE cuu NOT SUPPORTED	IPW$$CP	CP68
1R97I	COMMAND INVALID DURING SHUTDOWN PERIOD	IPW$$CP	CP02
1R98I	INVALID POWER/VS COMMAND CODE	IPW$$CP	CP74
1R99I	POWER/VS IS IN SHUTDOWN PERIOD	IPW$$CP	CP51
	POWER/VS HAS BEEN TERMINATED	IPW$$CP	CP48
1V01I	NO SUBTASK AVAILABLE FOR RJE,SNA	IPW$$SN	MC3
1V02I	VTAM OPEN FAILURE RTNCD=return code	IPW$$SN	MC3
1V03I	ERROR ON rpl request type RTNCD,FDB2=rtncd,fdb2 SENSE=sense	IPW$$SN	MC3,MC4, MC9
1V04I	RJE,SNA STARTED	IPW$$SN	MC4
1V05I	RJE,SNA TERMINATED	IPW$$SN	MC8

Message	Phase	Chart (Part 2)
1V06I UNABLE TO LOGON luname RC=yyy	IPW$$LN	MK4,MK5 MK11
	IPW$$LH	MK6,MK10
1V07I ERROR ON rpl request type RTNCD,FDB2=rtncd,fdb2 SENSE=sense ON luname	IPW$$SN	MC3,MC4 MC9
	IPW$$LN	MK4,MK11
	IPW$$MP	MF5
	IPW$$IB	MG4,MG5, MG10
	IPW$$OB	MH8,MH12, MH13,MH16,
	IPW$$LH	MK6,MK16
1V08I luname BIND PARAMETERS INVALID	IPW$$LH	MK8
1V09I REMOTE remid LOGGED ON TO POWER ON luname	IPW$$LN	MK11
1V10I RJE,SNA IS IN SHUTDOWN PERIOD	IPW$$SN	MC6,MC12
1V11I REMOTE remid LOGGED OFF FROM POWER ON luname	IPW$$LF	ME2
1V12I LOGOFF COMPLETED	IPW$$LF	ME2
1V13I LOGOFF FORCED	IPW$$LF	ME2
1V14I SESSION IS IN SHUTDOWN PERIOD	IPW$$SN	MC12
1V15I NO STORAGE AVAILABLE FOR task	IPW$$OB	MH3
1V16I NO STORAGE AVAILABLE FOR task FOR luname rrr	IPW$$OB	MH3
1V17A task SUSPENDED FOR FORMS MOUNT	IPW$$OB	MH6
1V18A REPLY WITH RESTART ON INTERVENTION REQUIRED task	IPW$$OB	MH16
1V22I INVALID command code COMMAND	IPW$$IB	MG11
1V23I command code OUT OF SEQUENCE	IPW$$IB	MG11
1V24I task TERMINATED REASON=reason code	IPW$$IB	MG5,MG10
	IPW$$OB	MH8
1V25I EOJ ADDED FOR jobname jobnumber	IPW$$IB	MG4
1V26I INVALID REMID, PASSWORD, OR LUNAME RC=yyy	IPW$$LH	MK7,MK20
1V27I REMID remid EXCEEDS SESSLIM	IPW$$LH	MK21
1V28I JOB jobname GETVIS FOR COCB FAILED	IPW$$OC	ML3
1V29I JOB jobname GETVIS FOR COMPACT TABLE FAILED	IPW$$OC	ML4
1V30I JOB jobname COMPACTION TABLE NOT FOUND	IPW$$OC	ML6
1V31I JOB jobname NO SPACE AVAIL. IN COMPACT POOL	IPW$$OC	ML2
1V32I JOB jobname INVALID COMPACTION TABLE	IPW$$OC	ML6
1V33I REMOTE remid OUTPUT FOR NON WRITER WORK STATION	IPW$$SN	MC33
1V34I (bind parameters are printed on the console)	IPW$$LH	MK8

Section 5. Layout of the Storage and Data Areas

This section describes the control blocks, buffer areas, save areas and work spaces required by POWER/VS in addition to the storage layout of the POWER/VS partition.

The first three figures act as a visual table of contents and contain references to the figure numbers where every area is shown in detail. Relationships between fundamental areas are shown in Figure 5.48 in the form of a series of examples.

Storage Descriptors

Most POWER/VS control blocks and many sections of POWER/VS code are equipped with storage descriptors which serve to rapidly locate and identify important values within a storage dump. A storage descriptor is a 16-byte alphameric character string with line alignment. Where appropriate, storage descriptors may also be addressed by internal programming. For instance, the storage descriptors of some TCBs are modified dynamically to reflect the function that the TCB is performing at any given time. For example, a storage descriptor of

```
TCBb1RDR.030.000
```

indicates the start of a task control block for an RJE reader task on RJE line number 30 invoked by the central operator. Thus, a storage descriptor identified in a dump constitutes a debugging aid.

The Position of the POWER/VS Data Areas

THE POWER/VS PARTITION STORAGE LAYOUT

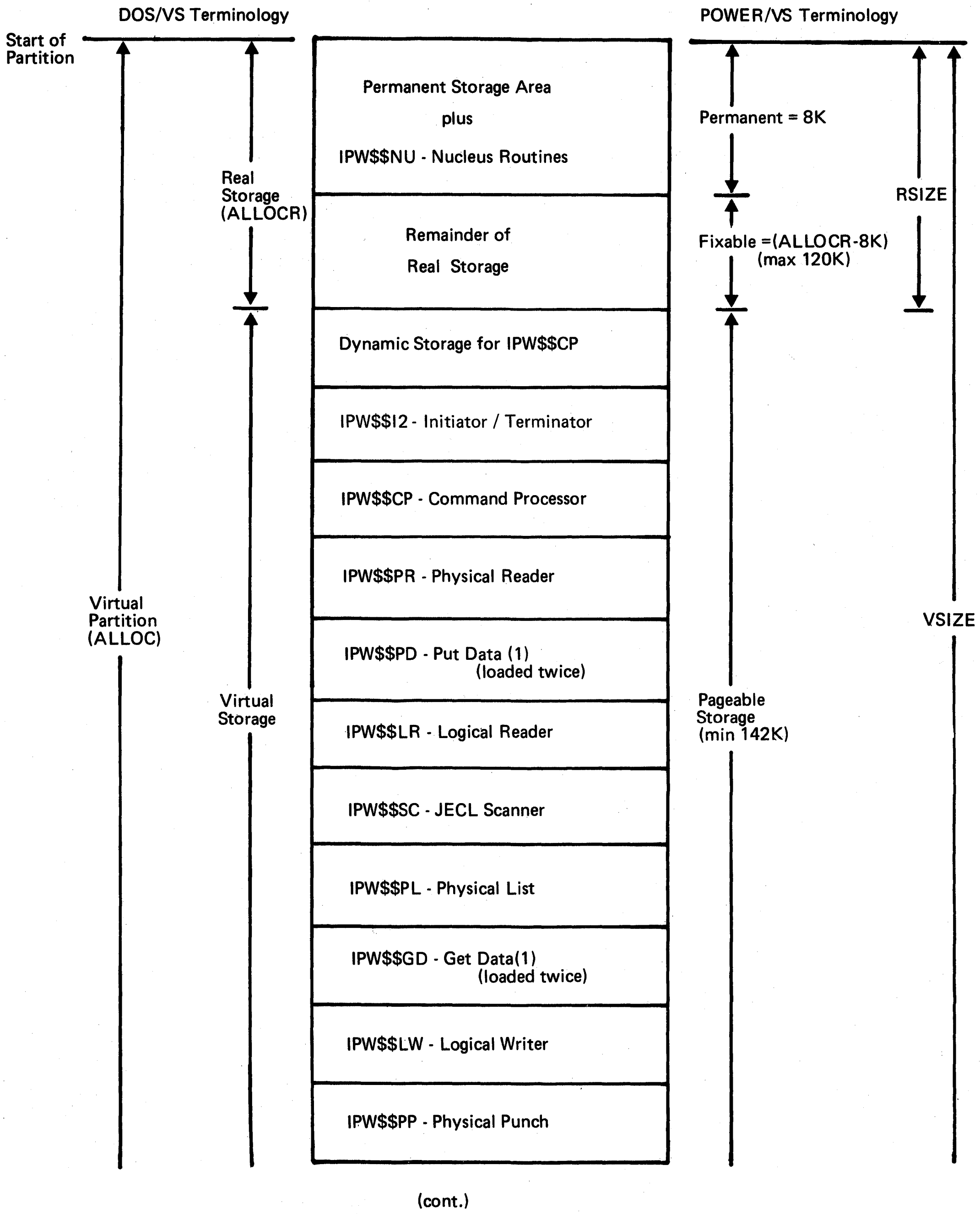
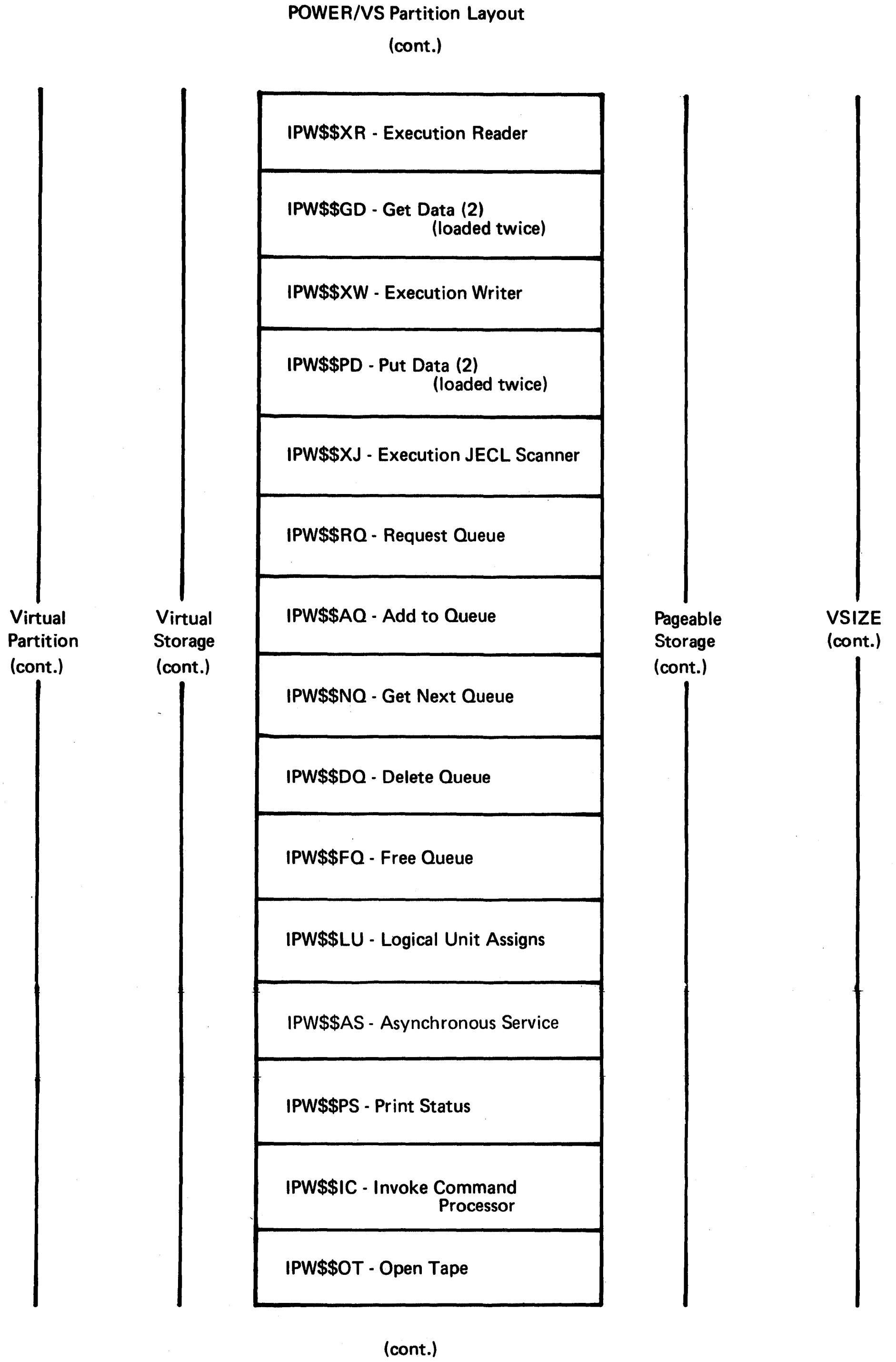


Figure 5.0. POWER/VS Partition Storage Layout (Part 1 of 4)



POWER/VS Partition Layout
(cont.)

IPW$$XR - Execution Reader

IPW$$GD - Get Data (2)
(loaded twice)

IPW$$XW - Execution Writer

IPW$$PD - Put Data (2)
(loaded twice)

IPW$$XJ - Execution JECL Scanner

IPW$$RQ - Request Queue

IPW$$AQ - Add to Queue

IPW$$NQ - Get Next Queue

IPW$$DQ - Delete Queue

IPW$$FQ - Free Queue

IPW$$LU - Logical Unit Assigns

IPW$$AS - Asynchronous Service

IPW$$PS - Print Status

IPW$$IC - Invoke Command
Processor

IPW$$OT - Open Tape

(cont.)

Figure 5.0. POWER/VS Partition Storage Layout (Part 2 of 4)

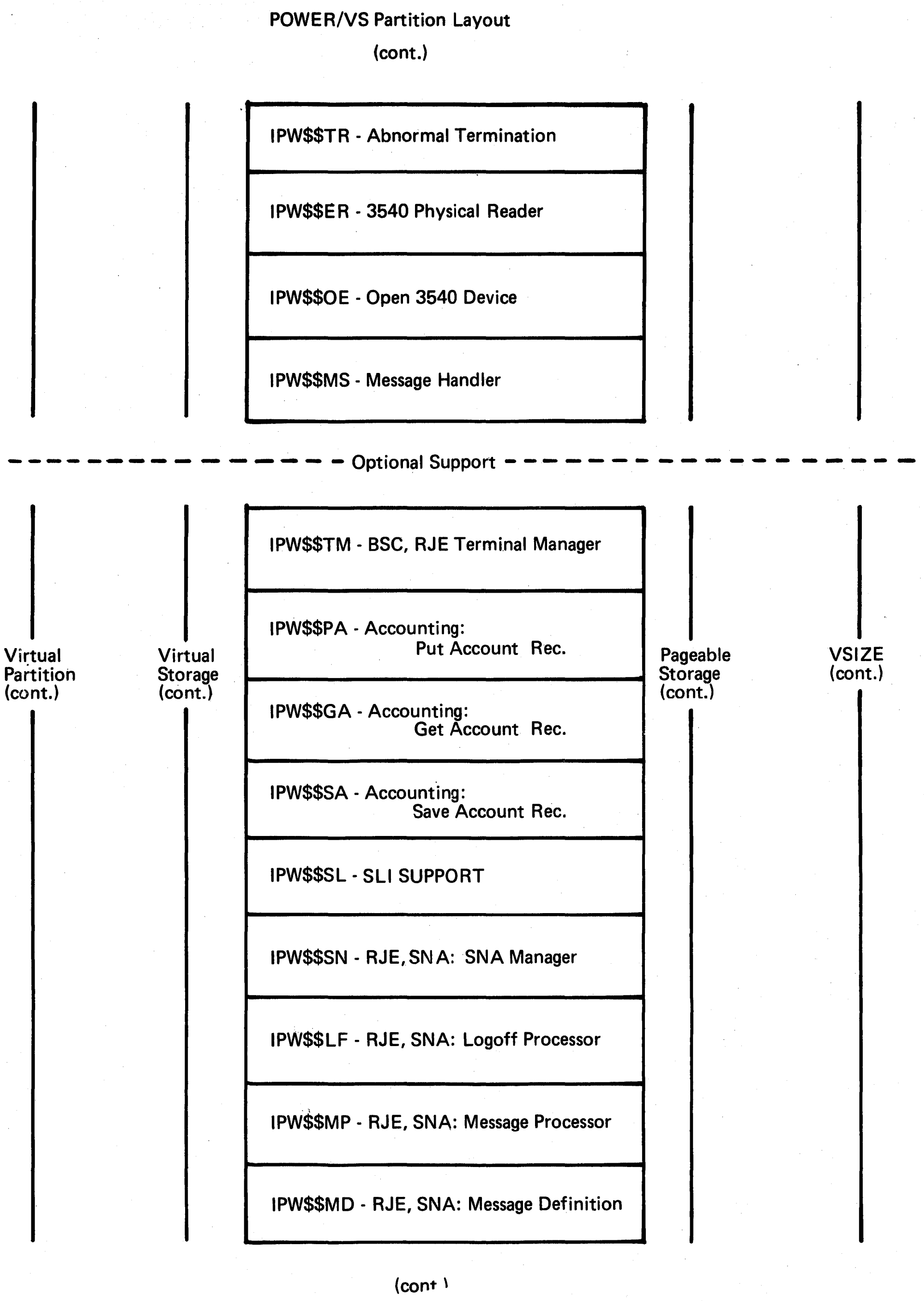

POWER/VS Partition Layout
(cont.)
IPW$$TR - Abnormal Termination
IPW$$ER - 3540 Physical Reader
IPW$$OE - Open 3540 Device
IPW$$MS - Message Handler
Optional Support
IPW$$TM - BSC, RJE Terminal Manager
IPW$$PA - Accounting: Put Account Rec.
IPW$$GA - Accounting: Get Account Rec.
IPW$$SA - Accounting: Save Account Rec.
IPW$$SL - SLI SUPPORT
IPW$$SN - RJE, SNA: SNA Manager
IPW$$LF - RJE, SNA: Logoff Processor
IPW$$MP - RJE, SNA: Message Processor
IPW$$MD - RJE, SNA: Message Definition
Virtual Partition (cont.)
Virtual Storage (cont.)
Pageable Storage (cont.)
VSIZE (cont.)
(cont.)

Figure 5.0. POWER/VS Partition Storage Layout (Part 3 of 4)

POWER/VS Partition Layout
Optional Support (cont.)

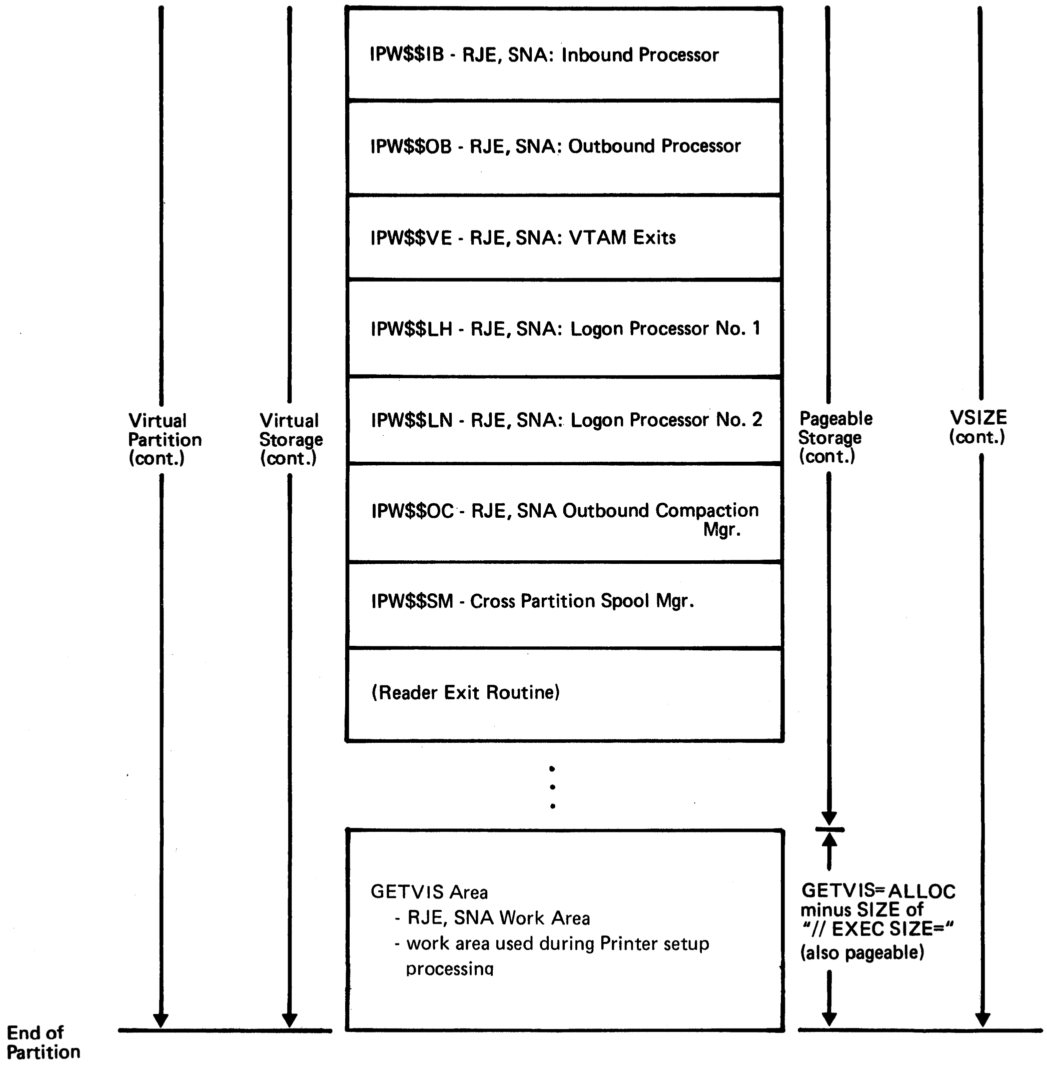

IPW$$IB - RJE, SNA: Inbound Processor
IPW$$OB - RJE, SNA: Outbound Processor
IPW$$VE - RJE, SNA: VTAM Exits
IPW$$LH - RJE, SNA: Logon Processor No. 1
IPW$$LN - RJE, SNA: Logon Processor No. 2
IPW$$OC - RJE, SNA Outbound Compaction Mgr.
IPW$$SM - Cross Partition Spool Mgr.
(Reader Exit Routine)
GETVIS Area
- RJE, SNA Work Area
- work area used during Printer setup processing
Virtual Partition (cont.)
Virtual Storage (cont.)
Pageable Storage (cont.)
VSIZE (cont.)
GETVIS=ALLOC minus SIZE of "// EXEC SIZE=" (also pageable)
End of Partition

Figure 5.0. POWER/VS Partition Storage Layout (Part 4 of 4)

THE PERMANENT AREA CONTROL TABLES

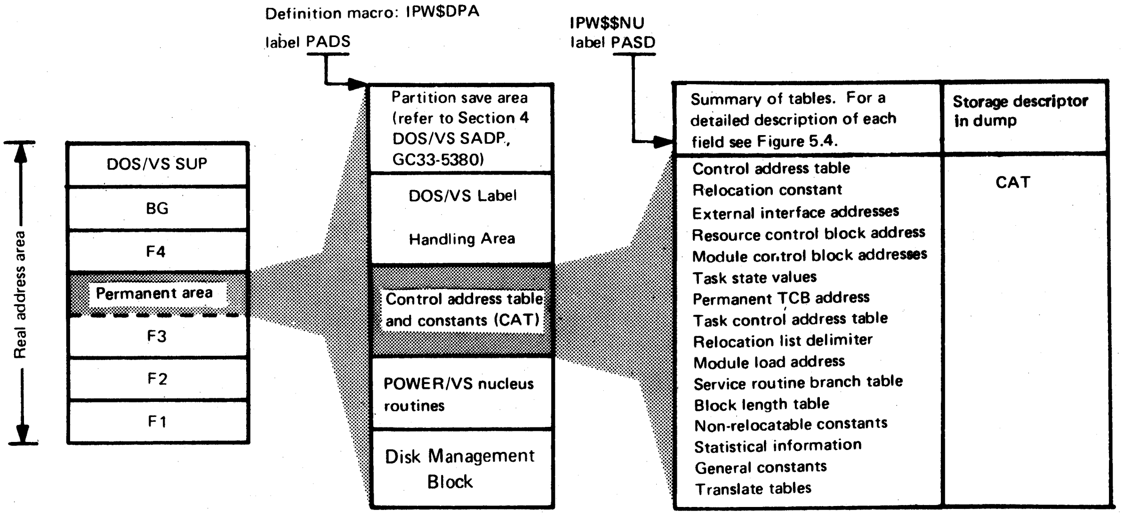

Definition macro: IPW$DPA
label PADS
IPW$$NU
label PASD
Real address area
DOS/VS SUP
BG
F4
Permanent area
F3
F2
F1
Partition save area (refer to Section 4 DOS/VS SADP, GC33-5380)
DOS/VS Label Handling Area
Control address table and constants (CAT)
POWER/VS nucleus routines
Disk Management Block
Summary of tables. For a detailed description of each field see Figure 5.4.
Storage descriptor in dump
Control address table
Relocation constant
External interface addresses
Resource control block address
Module control block addresses
Task state values
Permanent TCB address
Task control address table
Relocation list delimiter
Module load address
Service routine branch table
Block length table
Non-relocatable constants
Statistical information
General constants
Translate tables
CAT

Figure 5.1. Organization of the POWER/VS Permanent Area and the Control Address Table (CAT)

POWER/VS is shown in the F3 partition of a five partition system.

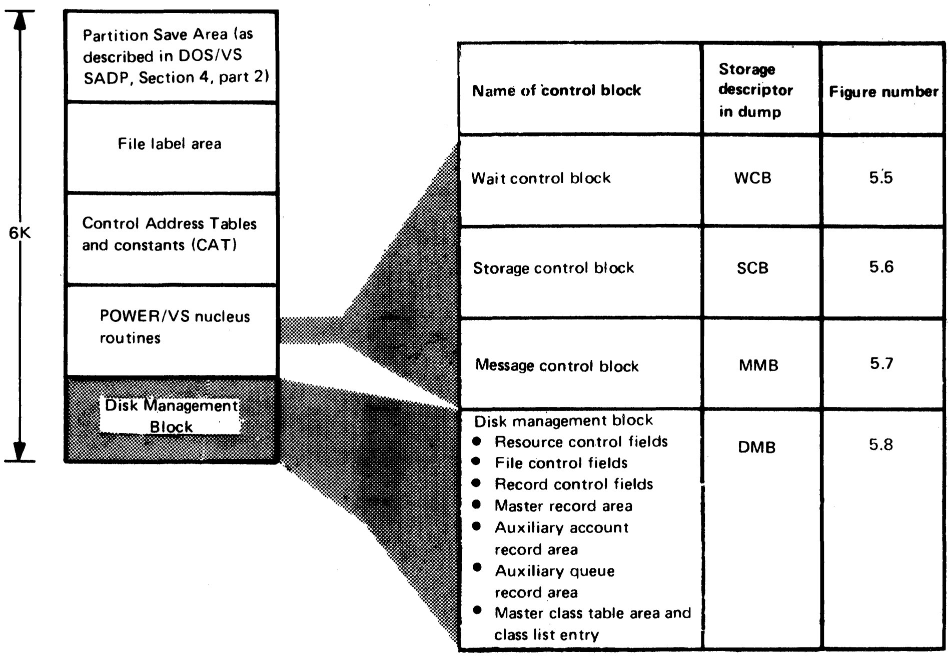
How to Locate

The Control Address Table starts at displacement X'140' from the start of the partition in which POWER/VS is initialized.

Appendix D shows the organization of the tables within this area as they are printed in a dump. It consists of a scale drawing which can be used as a template. If you have found the CAT in a dump, individual areas within the table can easily be identified by laying the table over the illustration.

THE PERMANENT AREA CONTROL BLOCKS

These control blocks are initiated by phase IPW$$I2 and remain fixed in the permanent area until POWER/VS is terminated. The location of each block is kept in the CAT. Each block has a storage descriptor enabling easy identification in a storage dump.

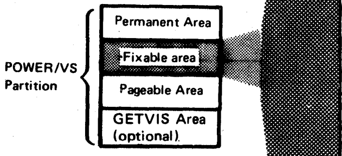

6K
Partition Save Area (as described in DOS/VS SADP, Section 4, part 2)
File label area
Control Address Tables and constants (CAT)
POWER/VS nucleus routines
Disk Management Block

Name of control block	Storage descriptor in dump	Figure number
Wait control block	WCB	5.5
Storage control block	SCB	5.6
Message control block	MMB	5.7
Disk management block ● Resource control fields ● File control fields ● Record control fields ● Master record area ● Auxiliary account record area ● Auxiliary queue record area ● Master class table area and class list entry	DMB	5.8

Figure 5.2. Organization of the POWER/VS Permanent Area with fixed control blocks

CONTROL BLOCKS DYNAMICALLY ALLOCATED IN THE FIXABLE AREA

These blocks are dynamically constructed, depending on the tasks required at any given time. The organization of the blocks relative to each other and the start of the fixable area cannot be truly illustrated. The figure, however, lists those blocks which are eligible to be in the fixable area.



Description of area	Storage descriptor	Fig.
Task Control Block • Task Management Fields • Task Register Save Area • General Task Work Area, Linkage Register Save Area, and File Control Words	TCB	5.9 5.10 5.11 5.12
Command Processor Control Block	CPB	5.13
2nd Linkage Register Save Area	None	5.14
Physical Work Space	None	5.15
Physical Data Record Area	None	
Logical Data Record Area	None	5.16
Module Control Block	MCB	5.17
Tape Control Block	TBB	5.18
Page Control Block	None	5.19
Buffer Control Word	None	5.20
Partition Control Block • Device Table Entry	PDB	5.21
Queue Record Area	None	5.22
SLI Work Space	None	5.23
Account Control Block	ACCB	5.24
Account Work Space	None	5.25
Reader Account Record	None	5.26
List Account Record	None	5.27
Punch Account Record	None	5.28
Execution Account Record	None	5.29
Line Account Record	None	5.30
Session Account Record	None	5.30
RJE,BSC Line Control Block	None	5.32
RJE,BSC Buffer Control Area	None	5.33
RJE,BSC Line Manager TCB Fields	None	5.34
RJE,SNA Control Block	SNCB	5.36
RJE,SNA Message Control Block	MSCB	5.39
RJE,SNA Manager TCB Fields	None	5.40
Generation Table	GNB	5.36
CCB	None	5.42
CCW	None	5.43
Separator Line Area	None	5.44
Diskette Work Space	OEWS	5.45
Asynchronous Service Anchor Block	ASWS	5.47.1
Service Request Block	None	5.47.2
Assign/Unassign Work Space	LUWS	5.47.3
3800 TCB Extension Area		5.47.4
Relationship Example	None	5.48

Figure 5.3. Control Blocks dynamically allocated in the Fixable area

THE GETVIS AREA CONTROL BLOCKS AND POOLS

The GETVIS area is an extension of the pageable area used to support RJE,SNA. Its space requirements are calculated by IPW$$I2, and contain the following:

a. RMCB - remote control block

plus the following GETVIS pools:

b. Logon space - containing
   one SUCB,
   one LUCB,
   one WACB with one buffer,
   and five LRCBs.

c. SUCB and LUCB - control blocks, containing one SUCB for each work station chained to a number of LUCBs (the number being equal to that of the maximum SESSLIM value specified in all PRMT macros at POWER/VS generation time.

d. WACB - work area control block

e. Compaction tables.

The start address of each pool is contained in the SNA control block (SNCB).

Another portion of the GETVIS area is used as a work area during 3800 printer setup processing. The area is released when the setup is done.

Layout of the POWER/VS Data Areas

CONTROL ADDRESS TABLE (CAT)

Included by definition macro IPW$DPA for the permanent area.

This table consists of a set of tables, addresses, and constants in the permanent area of the POWER/VS partition, used to link the component routines of the POWER/VS subsystem during execution. The format of this table as it is printed in a dump is shown below.

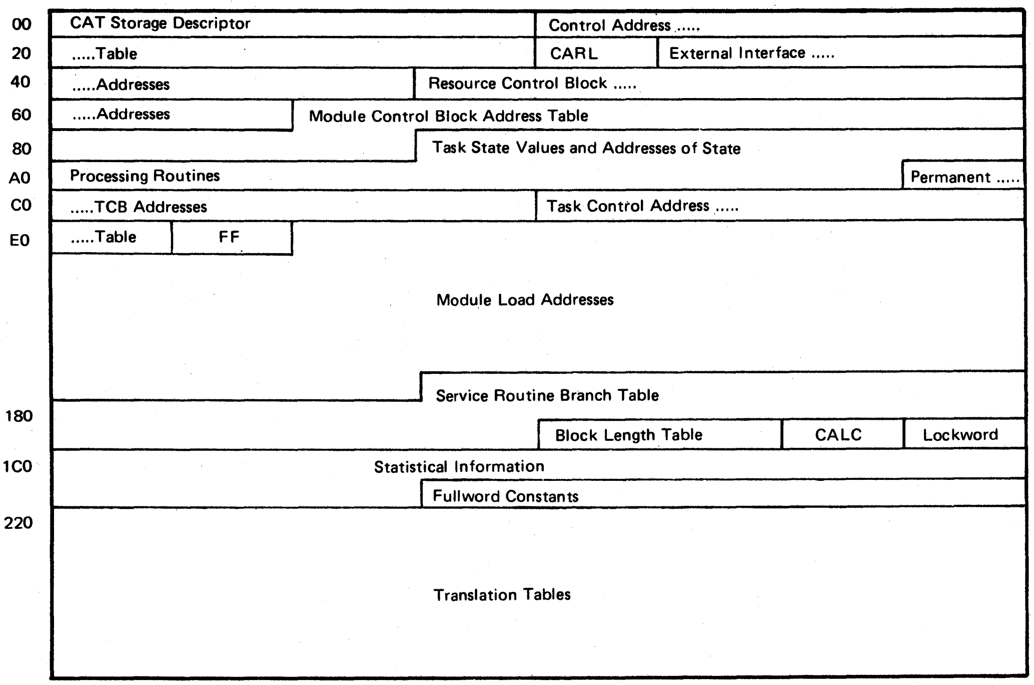


Figure 5.4. The Control Address Tables and Constants (Part 1 of 7)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Control address table	
00-15	00-0F	PASD	Storage descriptor (CAT)
16-19	10-13	PAEB	POWER/VS master ECB (refer to Appendix H)
20-23	14-17	PAPA	Start address POWER/VS partition
24-27	18-1B	PAFA	Start address fixable area
28-31	1C-1F	PAVA	Start address pageable area
32-35	20-23	PAEN	End address POWER/VS partition +1
36-39	24-27	PALS	Start address LTA
40-43	28-2B	PALE	End address LTA+1
44-47	2C-2F	CAPB	Address of POWER/VS PIB
		Relocation constant	
48-51	30-33	CARL	Relocation constant used by initiator to calculate the relocation factor for addresses in the following tables. (To enable POWER/VS to be loaded in any partition.) Value = X'170'. The number of relocatable constants is referenced by label CANN.
		External interface addresses	
52-55	34-37	CAAI	Attention interface
56-59	38-3B	CAPF	Page fault appendage
60-63	3C-3F	CAHR	Hot reader routine
64-67	40-43	CACE	RJE CE appendage
68-71	44-47	CA00	SVC 0 appendage
72-75	48-4B	CA90	SVC 90/91 appendage
		Spool management cross-partition XECB information.	
			Internal reader cross-partition XECB
76-79	4C-4F	ICXP	Internal reader XECB
80-80	50-50	IPIK	Internal RDR user's PIK/TIK
81-83	51-53	ICTA	XECBTAB ADDR of ICR XECB
			Spool/command manager cross-partition XECB
84-87	54-57	SMXP	Spool/command MGR XECB
88-88	58-58	SPIK	Spool/command MGR user's PIK/TIK
89-91	59-5B	SMTA	XECBTAB ADDR of SPM XECB
			selecting the internal RDR task and/or the spool/command manager LST task
92-95	5C-5F	ICWL	ADDR POWER'S internal reader XECB
63-99	60-63	SPWL	ADDR POWER'S spool/command MGR XBCB
100-100	64-64		WAITM list delimiter
101-103	65-67		Reserved.
		Resource control block addresses. This table is collectively referenced by label CAFR. The number of resources is referenced by label CANR.	
104-107	68-6B	CAQC	Disk management block
108-111	6C-6F	CAAC	Account control block
112-115	70-73	CASC	Storage control block
116-119	74-77	CAMM	Message control block (local)
120-123	78-7B	CARM	Message control block (remote)
124-127	7C-7F	CASM	SNA control block address
128-131	80-83	CAGP	General purpose work area
132-135	84-87	CAAB	Asynchronous service anchor block

Figure 5.4. The Control Address Tables and Constants (Part 2 of 7)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Module control block address table The addresses in this table are used by the disk services and are established when the POWER/VS disk files are opened at system start-up time.	
136-139	88-8B	CAA0	Accounting module
140-143	8C-8F	CAQ1	MCB queue file
144-147	90-93	CAD2	MCB data file module 1
148-151	94-97	CAD3	MCB data file module 2
152-155	98-9B	CAD4	MCB data file module 3
156-159	9C-9F	CAD5	MCB data file module 4
160-163	A0-A3	CAD6	MCB data file module 5
164-167	A4-A7	CAL7	MCB private SSL
168-171	A8-AB	CAL8	MCB system SSL
		Task state values and addresses of state processing routines These constants are used by the task management macro instructions to set values within the task selection fields of the task control blocks (except TMCW).	
172-175	AC-AF	TMCI	The task is inactive, task not selected Branch to TM10
176-179	B0-B3	TMCP	Page fault in process, task not selected Branch to TM10
180-183	B4-B7	TMCO	Wait for operator, task not selected Branch to TM10
184-187	B8-BB	TMCL	Wait on locked resource, test lockword Branch to TM30
188-191	BC-BF	TMCF	Wait ofn LTA/PTA, test control blocks Branch to TM55
192-195	C0-C3	TMCM	Wait on multiple posting, test control blocks Branch to TM50
196-199	C4-C7	TMCQ	Wait on class table posting, test control blocks Branch to TM50
200-203	C8-CB	TMCC	Wait on single posting, test control block Branch to TM80
204-207	CC-CF	TMCS	Wait on space posting, test control blocks Branch to TM80
208-211	D0-D3	TMCD	Immediate dispatch, dispatch the task Branch to TM90
212-215	D4-D7	TMCW	Wait state. Used for WCB only. Wait routine. Branch to TM20.
216-219	D8-DB	TMCR	The task is running, re-selection address
		Permanent TCB addresses	
220-223	DC-DF	TATM	Wait control block
224-227	E0-E3	TAOC	Command processor TCB
228-231	E4-E7	TAIT	Initialization/termination TCB
232-235	E8-EB	TALM	Line manager TCB
236-239	EC-EF	TASP	Spool manager TCB address

Figure 5.4. The Control Address Tables and Constants (Part 3 of 7)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Task control address table	
240-243	0F0-0F3	CALM	Task identifying prefix (L) and the address of the TCB of the line or SNA manager (or of the wait control block if the line manager is not present).
244-247	0F4-0F7	CASP	Task identifying prefix (J) and address of most recently attached spool management TCB.
248-251	0F8-0FB	CAOP	Task identifying prefix (O) and the address of the TCB of the most recently attached auxiliary command processor (or of the permanent command processor if no auxiliary command processor presently exists).
252-255	0FC-0FF	CARJ	Task identifying prefix (X) and the address of the TCB of the most recently attached remote (RJE) reader/writer.
256-259	100-103	CARW	Task identifying prefix (W) and the address of the TCB of the most recently attached local writer task.
260-263	104-107	CAEX	Task identifying prefix (E) and the address of the TCB of the most recently attached execution processor task.
264-267	108-10B	CARR	Task identifying prefix (R) and the address of the TCB of the most recently attached reader task.
268-271	10C-10F		X'FF000000' (list delimiter)
		Module load addresses (listed as loaded in the pageable area) The first module is referenced by label CAFM. The number of modules is referenced by label CANM. The number of SNA modules is referenced by label CANS.	
		Command processor module	
272-275	110-113	CACP	Command processor
		Reader task modules	
276-279	114-117	CAPR	Physical reader
280-283	118-11B	CAPD	Put data record function
284-287	11C-11F	CALR	Logical reader
288-291	120-123	CASN	Scan and check parameter function
		Writer task modules	
292-295	124-127	CAPL	Physical list
296-299	128-12B	CAGD	Get data record function
300-303	12C-12F	CALW	Logical writer
304-307	130-133	CAPP	Physical punch
		Execution processor modules	
308-311	134-137	CAXR	Execution reader
312-315	138-13B	CAXG	Get data record function (copy 2 refer to Section 3, XR/XW)
316-319	13C-13F	CAXW	Execution writer
320-323	140-143	CAXP	Put data record function (copy 2 refer to Section 3, XR/XW)
324-327	144-147	CAXJ	JECL analysis

Figure 5.4. The Control Address Tables and Constants (Part 4 of 7)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Queue management modules	
328-331	148-14B	CARQ	Reserve queue function
332-335	14C-14F	CAAQ	Add to queue function
336-339	150-153	CANQ	Get next from queue function
340-343	154-157	CADQ	Delete from queue function
344-347	158-15B	CAFQ	Free queue function
		Miscellaneous modules	
348-351	15C-15F	CALU	LUB/PUB update function
352-355	160-163	CAAS	Asynchronous service function
356-359	164-167	CAPS	Print queue status report
360-363	168-16B	CAIC	Invoke command processor function
364-367	16C-16F	CAOT	Open tape routine
368-371	170-173	CATR	Task terminator
372-375	174-177	CAER	3540 Physical reader
376-379	178-17B	CAOE	3540 Open routine
380-383	17C-17F	CAMS	Message handler
		Spool management option	
384-387	180-183	CASF	Spool manager
		The following modules are optional	
		Remote job entry BSC module	
388-391	184-187	CATM	Remote job entry
		Reader exit module	
392-395	188-18B	CARE	User reader exit routine
		Accounting modules	
396-399	18C-18F	CAPA	Put account function
400-403	190-193	CAGA	Get account function
404-407	194-197	CASA	Save account function
		Source library include module	
408-411	198-19B	CASL	Get SSL function
		Remote job entry SNA modules	
412-415	19C-19F	CAS0	SNA manager
416-419	1A0-1A3	CAS2	SNA logoff processor
420-423	1A4-1A7	CAS3	SNA message processor
424-427	1A8-1AB	CAS4	SNA message definition
428-431	1AC-1AF	CAS5	SNA inbound processor
432-435	1B0-1B3	CAS6	SNA outbound processor
436-439	1B4-1B7	CAS7	VTAM Exits module
440-443	1B8-1BB	CAS8	Logon processor 1 - IPW$$LH
444-447	1BC-1BF	CAS9	Logon processor 2 - IPW$$LN
448-455	1C0-1C3	CAS10	Reserved

Figure 5.4. The Control Address Tables and Constants (Part 5 of 7)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Service routine branch table The branch instructions are used to transfer control from service routine macro instructions to the appropriate service code.	
456-459	1C8-1CB	TA00	Attach task
460-463	1CC-1CF	TD00	Detach task
464-467	1D0-1D3	TM00	Task selection
468-471	1D4-1D7	TS00	Initial task entry
472-475	1D8-1DB	RM00	Reserve resource
476-479	1DC-1DF	RM50	Release resource
480-483	1E0-1E3	SM00	Reserve work space
484-487	1E4-1E7	SM50	Release work space
488-491	1E8-1EB	MM00	Message service (local)
492-495	1EC-1EF	MM50	Message service (remote)
496-499	1F0-1F3	DM00	Set write command code
500-503	1F4-1F7		Disk service
504-507	1F8-1FB	DM10	Set read command code
508-511	1FC-1FF		Disk service
512-515	200-203	TP00	Tape service
516-519	204-207	TR00	Timer service
520-523	208-20B	VA00	Validation service
		Block length table The table is used by the IPW$RSW macro instruction to identify the size of work space required to accommodate certain control blocks.	
524-527	20C-20F	BLBF	Data buffer - set by INIT (amount of storage required to accommodate the data block)
528-531	210-213	BLDB	Data block - set by INIT (size of record written to disk)
		Miscellaneous non-relocatable constants	
532-535	214-217	CALC	Line control block address
536-539	218-21B		BSC,RJE lockword
		Statistical information (refer to Appendix E)	
540-541	21C-21D	NRRE	Highest BSC remote ID
542-543	21E-21F	NRLI	Number of BSC lines
544-547	220-223	NRQR	Total number of queue records
548-551	224-227	NRQF	Number of free queue records
552-555	228-22B	NRQM	Maximum number of queue records used
556-559	22C-22F	NRTR	Total number of tracks data file
560-563	230-233	NRTW	Number of times waiting for storage
564-567	234-237	NRPG	Total number of pages allocated
568-571	238-23B	NRPC	Current number of pages allocated
572-575	23C-23F	NRPM	Maximum number of pages allocated
576-579	240-243	NRTC	Current number of tasks
580-583	244-247	NRTH	Maximum number of tasks
		Fullword	constants
584-587	248-24B	CF01	F'1'
588-591	24C-24F	CF04	F'4'
592-595	250-253	CF08	F'8'
596-599	254-257	CF10	F'10'
600-603	258-25B	CF24	F'24'

Figure 5.4. The Control Address Tables and Constants (Part 6 of 7)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Translation tables	
604-859	25C-35B	TRTB	This table is used to scan sequences of blank characters for the first non-blank character and also as a source of blank characters for various program purposes.
860-1115	35C-45B	TRTC	This table is used to scan sequences of non-blank characters for the first blank character and also as a source of zero characters for various program purposes.

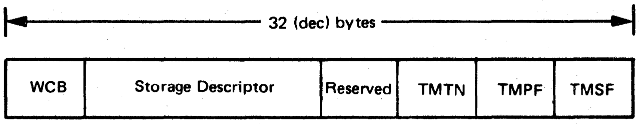
Figure 5.4. The Control Address Tables and Constants (Part 7 of 7)

How to Locate

Refer to Figure 6.1 in Section 6.

WAIT CONTROL BLOCK (WCB)

The wait control block is a skeleton task control block used to delimit the task selection list. The wait control block occupies locations in the permanent area of the POWER/VS partition.



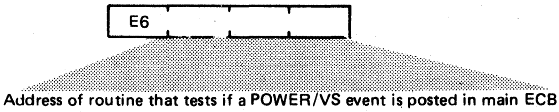
Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	TMSD	Storage descriptor (WCB)
16-19	10-13		Reserved
20-23	14-17	TMTN	Address of TCB belonging to task with highest priority in TSL
24-27	18-1B	TMPF	Page fault request word - always zero
28-31	1C-1F	TMSF	Task selection field  Address of routine that tests if a POWER/VS event is posted in main ECB. If not, it places the POWER/VS partition in wait state by issuing an SVC7.

Figure 5.5. Wait Control Block

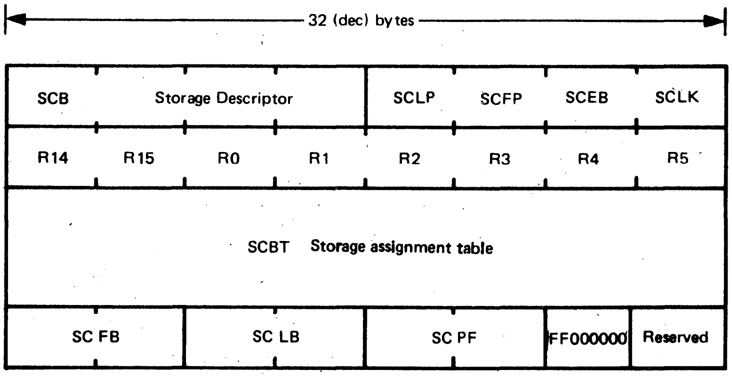
How to Locate

Refer to Figure 6.1 in Section 6.

STORAGE CONTROL BLOCK (SCB)

Definition macro: IPW$DSC

The storage control block is used to control access to the storage management routines and to allocate storage pages as required by the routines. The format of the table as it is printed in a dump is shown below:



SCB	Storage Descriptor			SCLP	SCFP	SCEB	SCLK
R14	R15	R0	R1	R2	R3	R4	R5
SCBT Storage assignment table							
SC FB		SC LB		SC PF		FF000000	Reserved

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	SCSD	Storage descriptor (SCB)
16-19	10-13	SCLP	Last permanent page
20-23	14-17	SCFP	First fixed page
24-27	18-1B	SCEB	Event control block
28-31	1C-1F	SCLK	Lockword
32-35	20-23	SCRE	Task register 14
36-39	24-27	SCRF	Task register 15
40-43	28-2B	SCR0	Task register 0
44-47	2C-2F	SCR1	Task register 1 [1]
48-51	30-33	SCR2	Task register 2
52-55	34-37	SCR3	Task register 3
56-59	38-3B	SCR4	Task register 4
60-63	3C-3F	SCR5	Task register 5
64-127	40-7F	SCBT	Storage assignment table [2]
128-135	80-87	SCFB	Constant to initialize the first BCW (see Figure 5.20) in a new fixed page in the fixable area.
136-143	88-8F	SCLB	Constant to initialize the last BCW in a new fixed page in the fixable area (see Figure 5.20)
		SCPF	Page fix/free work area [3]
144-147	90-93		Page virtual address [3]
148-151	94-97		Page length (-1) [3]
152-155	98-9B		End-of-list indicator (X'FF000000')
156-159	9C-9F		Reserved

Figure 5.6. Storage Control Block

[1] Since the storage management routines are used to provide register save areas for task use, the storage control block must contain a register save area for use by the storage management routines.

[2] The storage assignment table is like a map of the fixable area within the POWER/VS address space in which each page control byte represents a single page of address space. Each byte within the table takes one of four values.

X'00'	Page free (and not last page)
X'40'	Page free (and last page)
X'80'	Page in use (but not last page)
X'C0'	Page in use (and last page)

The storage assignment table is defined with all pages free and is properly initialized by the POWER/VS start-up routines to reflect the amount of real storage available to the POWER/VS partition at that time.

[3] Three fullwords used as a work area by the page-fix and page-free routines. The first word is used to contain the address of the first byte of the page to be fixed or freed; the second word contains binary 2047 (page size minus one); and the third word contains X'FF' in its high-order byte to act as a list terminator.

How to Locate

Refer to Figure 6.1 in Section 6.

MESSAGE CONTROL BLOCK (MMB)

Definition macro: IPW$DMM

This block provides support for the macros IPW$WTO and IPW$WTR. A routine issuing one of these macros will invoke message services. A message to be printed on SYSLOG will be passed to the MMB by means of the message request word in the TCB. The MMB also contains the channel program (CCB and CCW) to execute the I/O to the console. If a reply is necessary the channel program in the MMB will execute the necessary I/O. The message service will move the reply to an area addressed by the reply request word in the TCB for the task using the routine that issued the IPW$WTR macro. (See also TCMW and TCAW fields in the TCB, Figure 5.11.)

The format of this block as it is printed in a dump is shown below:

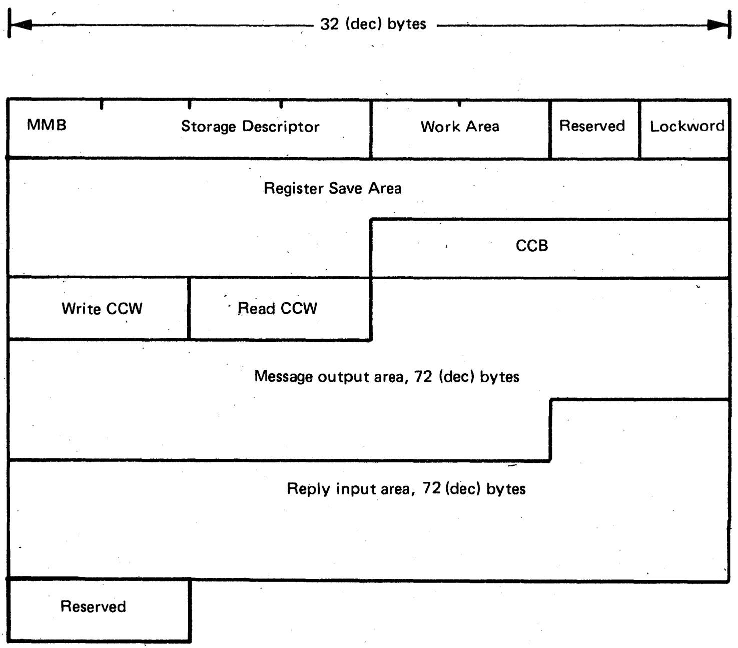


Figure 5.7. Message Control Block (Part 1 of 2)

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	MMSD	Storage descriptor (MMB)
16-23	10-17	MMWW	Work area
24-27	18-1B		Reserved
28-31	1C-1F	MMLK	Lockword
		MMSV	Register save area
32-35	20-23	MMRE	Saved register 14
36-39	24-27	MMRF	Saved register 15
40-43	28-2B	MMR0	Saved register 0
44-47	2C-2F	MMR1	Saved register 1
48-51	30-33	MMR2	Saved register 2
52-55	34-37	MMR3	Saved register 3
56-59	38-3B	MMR4	Saved register 4
60-63	3C-3F	MMR5	Saved register 5
64-67	40-43	MMR6	Saved register 6
68-71	44-47	MMR7	Saved register 7
72-75	48-4B	MMR8	Saved register 8
76-79	4C-4F	MMR9	Saved register 9
		MMCB	CCB
80-81	50-51	MMCT	Residual count
82-83	52-53	MMCM	Communication bytes
84-85	54-55	MMST	Status bytes
86-87	56-57	MMLU	LUB identifier
88	58	MMCA	Flags
89-91	59-5B		Channel program address
92-95	5C-5F		DOS/VS internal use
		MMCH	Channel program
96-103	60-67	MMWT	Write CCW
104-111	68-6F	MMRD	Read CCW
112-183	70-B7	MMMA	Message output area
184-255	B8-FF	MMMI	Reply input area
256-263	100-107		Reserved

Figure 5.7. Message Control Block (Part 2 of 2)

How to Locate

Refer to Figure 6.1 in Section 6.

DISK MANAGEMENT BLOCK (DMB)

Definition macro: IPW$DQC

The disk management block area is used to control access to the POWER/VS queue file. It is located in the permanent area of the POWER/VS partition.

The disk management block is divided into the following areas:

- Resource control fields
- File control fields
- Record control fields
- Master record area
- Auxiliary account record area
- Auxiliary queue record area
- Master class table area.

The format of the table as it is printed in a dump is shown below:

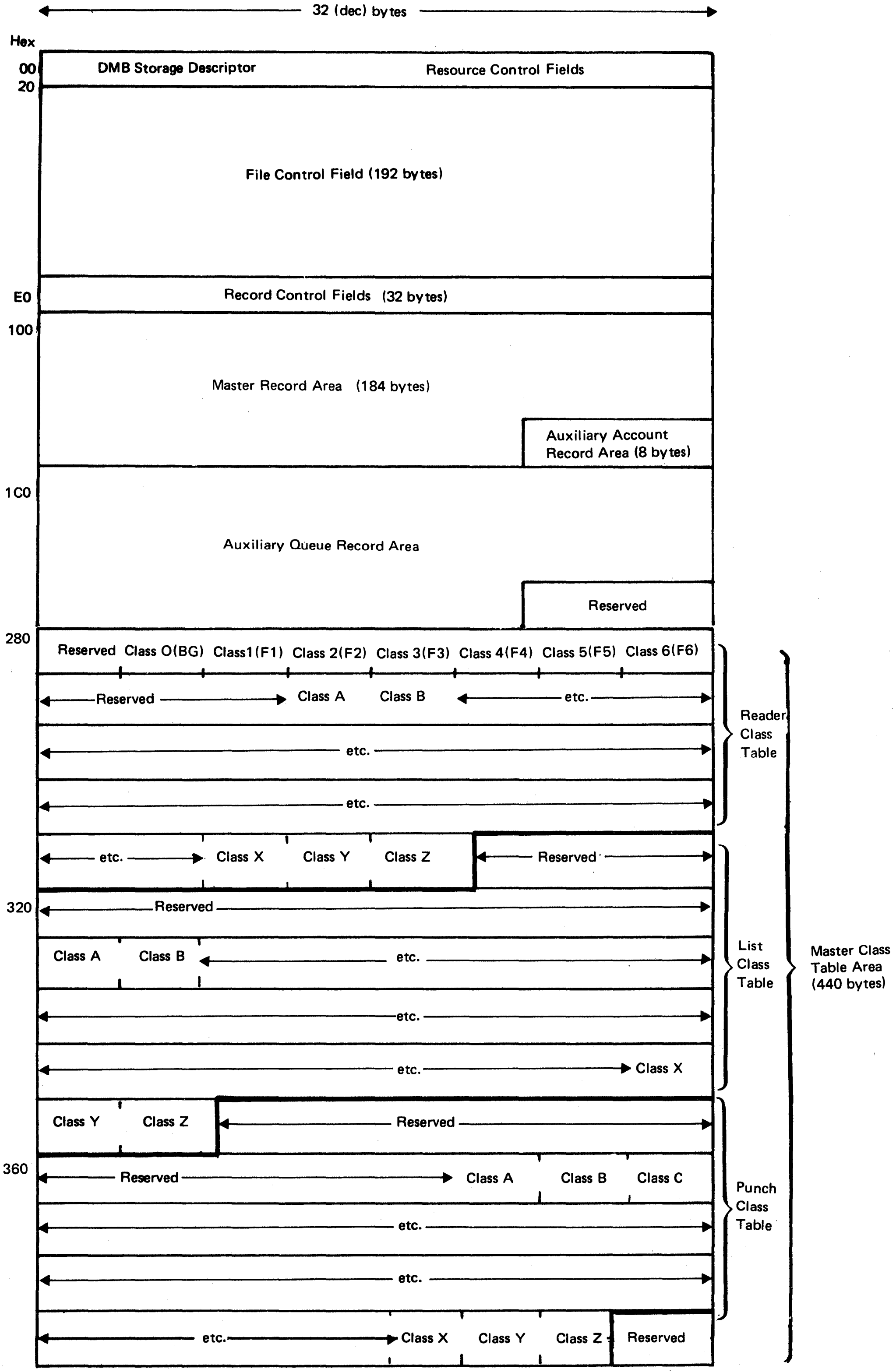

32 (dec) bytes
Hex
00
20
E0
100
1C0
280
320
360
DMB Storage Descriptor
Resource Control Fields
File Control Field (192 bytes)
Record Control Fields (32 bytes)
Master Record Area (184 bytes)
Auxiliary Account Record Area (8 bytes)
Auxiliary Queue Record Area
Reserved
Reserved Class O(BG) Class1(F1) Class 2(F2) Class 3(F3) Class 4(F4) Class 5(F5) Class 6(F6)
Reserved Class A Class B etc.
etc.
etc.
etc. Class X Class Y Class Z Reserved
Reserved
Class A Class B etc.
etc.
etc. Class X
Class Y Class Z Reserved
Reserved Class A Class B Class C
etc.
etc.
etc. Class X Class Y Class Z Reserved
Reader Class Table
List Class Table
Punch Class Table
Master Class Table Area (440 bytes)

Figure 5.8. Disk Management Block (Part 1 of 9)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Resource control fields They are used to manage the resources contained within the DMB	
00-15	00-0F	QCSD	Storage descriptor (DMB)
16-23	10-17		Reserved
24-27	18-1B	QCEB	Event control block
28-31	1C-1F	QCLK	Lockword
		File control fields They contain parameters relating to queue file, data file, and, if used, private and system SSL	
32-35	20-23	QC#R•	Number of records/track queue file [1]
36-39	24-27	QC#T•	Number of tracks/cylinder queue file [2]
40-87	28-57	QCSC	Queue file sector table
88-95	58-5F		Reserved
96-99	60-63	DC#R	Number of records/track data file [1]
100-103	64-67	DC#T	Number of tracks/cylinder data file [2]
104-135	68-87	DCTR	Track group control table
136-167	88-A7	DCSC	Data file sector table
168-223	A8-DF		Reserved for future use
		Record control fields They contain information used to read and write records to and from the master record area and auxiliary queue record area.	
224-231	E0-E7	QCMW	Master record seek address (MBBCCHHR)
232-235	E8-EB	QCMA	Real master area address
236-239	EC-EF	QCMV	Virtual master area address
240-247	F0-F7	QCQW	Queue record seek address (MBBCCHHR)
248-251	F8-FB	QCQA	Real auxiliary queue record area address
252-255	FC-FF	QCQV	Virtual queue record area address
		Master record area The master record is written as the first physical record within the queue file extent. During POWER/VS execution a copy of the master record is maintained in this area. Whenever this copy is updated a replacement master record is at once written to the queue file so that, in the event of a failure of the system, warm start information can be recovered from the direct access device in question.	
256-263	100-107	MRDY	Date These eight bytes contain the date of POWER/VS execution in the format chosen at system generation (dd/mm/yy or mm/dd/yy).

Figure 5.8. Disk Management Block (Part 2 of 9)

Bytes		Label of field	Description/function of field
Dec	Hex		
264-267	108-10B	MRST	POWER/VS start time These four bytes contain the start time of POWER/VS execution in packed decimal format.
268-271	10C-10F		Reserved
272-275	110-113	MRDB	Data block size This fullword contains a fixed-point binary value representing the data block size of the data blocks within the data file. This is the length of the physical records written to the data file.
276-279	114-117	MRTG	Track group size This fullword contains a fixed-point binary value representing the number of tracks within each track group within the data file.
280-283	118-11B	MRVM	Version and modification level four numeric characters representing the version and modification level of POWER/VS used.
284-289	11C-121		Reserved
		Programming Note: Tne following 6 switch bytes preserve the options established by the POWER/VS user at the time he generated his version.	
290	122	MRSL	Source library switch This byte contains a single alphabetic character representing the source statement sublibrary to be associated, unless otherwise specified, with any JECL SLI statements encountered in the read queue.
291	123	MRJA	Job accounting switch This byte contains a single alphabetic character; the character A indicates that POWER/VS job accounting is required; a blank character indicates that POWER/VS accounting is not required.
292	124		Reserved
293	125	MRLG	LOG option switch (set to character L if JLOG=YES and blank if JLOG=NO)
294	126	MRTT	Termination status. Contains character A for incomplete session or abnormal termination. Otherwise, it contains character N, meaning normal termination. Note: It will contain an A during the session.
295-303	127-12F		Reserved for future use
		Programming Note: The following 14 bytes contain standard POWER/VS default values used when new queue records are created.	
304-311	130-137	MRNM	Default job name These eight bytes contain the character string 'AUTONAME' used as a default job name.

Figure 5.8. Disk Management Block (Part 3 of 9)

Bytes		Label of field	Description/function of field
Dec	Hex		
312-313	138-139	MRNO	Master job number This halfword contains a fixed-point binary value representing the <u>next</u> job number to be assigned by POWER/VS. It is incremented by one each time it is used.
314	13A	MRQI	Master queue identifier This byte contains the alphabetic character M to show that this is the master record.
315	13B	MRCL	Default class attribute This byte contains the alphabetic character A representing the class attribute to be given by default to each RDR queue entry created within POWER/VS.
316	13C	MRPY	Default priority attribute This byte contains numeric character 3 which defines the priority attribute to be given by default to each queue entry created by POWER/VS.
317	13D	MRCN	Default cancel code This byte contains the hexadecimal characters X'10' representing normal end of job and task.
318-319	13E-13F		Reserved
		<u>Programming Note</u>: Next 16-byte field contains the master line table, consisting of system default values used to analyse space and skip operations during printer control carriage simulation.	
320-343	140-157	MRLT	Line table
		<u>Programming Note</u>: Next 20 bytes contain the master list values, which will be inserted by default in list queue records, unless overridden by a JECL LST statement. (Values are set by IPW$$I1 using those specified by user during POWER/VS generation [JSEP=, RBS=, STDLINE=])	
344-359	158-167	MRLV	Master list values
344	158	MROP	Option byte X'01' - 3540 feed option X'02' - Multiple channel 12 option (see POWER macro) X'80' - Clear printer at EOF X'40' - Mark form option for separator pages X'20' - No separator pages between copies
345-346	159-15A		Reserved
347	15B	LVSP	Number of separators
348-351	15C-15F	LVBS	Records before segmentation
352-355	160-163	LVBM	Records before message
356-359	164-167	LVBN	Records before next message
360-363	168-16B		Reserved
		<u>Programming Note</u>: Next 20 bytes contain the master punch values, which will be inserted by default in punch queue records, unless overridden by a JECL PUN statement. (Values set by IPW$$I1 using those specified by user during POWER/VS generation. [JSEP=, RBS=, STDCARD=])	

Figure 5.8. Disk Management Block (Part 4 of 9)

Bytes		Label of field	Description/function of field
Dec	Hex		
364-366 367 368-371 372-375 376-379 380-383	16C-16E 16F 170-173 174-177 178-17B 17C-17F	 PVSP PVBS PVBM PVBN 	Reserved Number of separators Records before segmentation Records before message Records before next message Reserved
		Programming Note: Next 10 bytes contain account file values	
384-391	180-187	MRAS	Account file seek address (MBBCCHHR) Contains the direct access storage seek address of the last record in the POWER/VS account file.
392-393	188-189	MRAZ	Account file record maximum size Binary value representing the length of the longest recora so far written to the account file.
394-407	18A-197		Reserved
		Programming Note: Next 32 bytes contain free queue pointers	
408-415	198-19F	MRQF	First record in free queue (MBBCCHHR)
416-439	1A0-1B7		Reserved
		Auxiliary account record area This area actually overlaps the auxiliary queue record area, because the account record consists of the first part of the queue record which is built in that area. All account records except execution account are transferred from here to the account file as standard variable length records.	
440-447	1B8-1BF	ACPR	Block and record length This record control field is used for sequential access method.
		Auxiliary queue record area (184 bytes) This area is required as a work space for an additional queue record (see Figure 5.22 and description Queue Record Area (QRA)). For example, for updating class chain addresses during the add to queue function. The first part (103 bytes) of the Q record contains body fields (information pertinent to this particular queue entry and the user job which created it).	
448-455	1C0-1C7	QCDY	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy)
456-459	1C8-1CB	QCST	Operation start time, in packed decimal (0HHMMSSF; F = sign)
460-463	1CC-1CF	QCET	Operation end time (0HHMMSSF; F = sign)
464-479	1D0-1DF	QCUI	16 bytes user information
480-487	1E0-1E7	QCNM	Job name Job name associated with this particular POWER/VS or DOS/VS job. If no job name is provided by the user the default value AUTONAME is set into this field.

Figure 5.8. Disk Management Block (Part 5 of 9)

Bytes		Label of field	Description/function of field
Dec	Hex		
488-489	1E8-1E9	QCNO	Job number Contains a binary job number assigned to the job upon its entry into the system and thereafter available for further identification of jobs with a common job name.

490	1EA	QCQI	Queue record identifier
			R = read queue record
			L = list queue record
			P = punch queue record
			F = free queue record
			D = dummy queue record

491	1EB	QCCN	POWER/VS cancel codes Cancel Code Condition X'10' Normal end of POWER/VS job or task [3] X'20' PCANCEL has been issued X'30' PSTOP has been issued [4] X'40' PFLUSH has been issued X'50' PDELETE has been issued X'60' PFLUSH has been issued via RDREXIT X'70' Canceled due to I/O error
492	1EC	QCRJ QCDT	Line identifier/device type
493-495	1ED-1EF	QCCU	Channel and unit (line address)
496	1F0	QCFJ	From-terminal identifier
497	1F1	QCTJ	To-terminal identifier
498	1F2	QCCL	Class (default = A)
499	1F3	QCPY	Priority (default = 3) This single byte contains the priority value (numeric 0 to 9), assigned by the user to this job operation.
500-503	1F4-1F7	QCNR	Record count Binary counter that represents the number of input or output data records associated with the read, list, or punch operation (data transfer and control operations).
504-505	1F8-1F9	QCNT	Number of tracks for output storage Binary counter recording the number of tracks within the data file used to contain data input or output for this particular job operation.
506	1FA	QCSN	Job suffix number Binary job suffix number assigned to each successive operation (read, list, or punch) performed on behalf of the job. It may be used to identify output sets produced by jobs handling segmented output.
507	1FB	QCNC	Number of copies This single byte contains a binary value indicating the number of copies of printed or punched output that are to be produced when the output is processed by the writer tasks. It has no use within input-related queue records.

Figure 5.8. Disk Management Block (Part 6 of 9)

Bytes		Label of field	Description/function of field
Dec	Hex		
508-511	1FC-1FF	QCFI	Forms identifier. Alphameric forms or card identifier of any special stationery or card stock to be used when creating the physical output from the job. A blank value indicates that no special requirement exists. The field has no use within input-related queue records.
512-515	200-203	QCNA	Number of additional records
516-517	204-205	QCNP	Number of pages (number of skips to channel 1)
518-519	206-207	QCNE	Number of extra pages
520-523	208-20B	QCLC	Line/card counter (data transfers only, see QCNR)
524-527	20C-20F	QCRR	Restart page counter (used when PRESTART command given)
528	210	QCCR	Copies remaining (used when PRESTART command given)
529	211	QCDI	Not used
530	212	QCDP	Disposition (default = D)
531	213	QCSP	Number of separators. Binary value indicating the number of printed output separators to be produced. It has no use within input-related queue records.
532-535	214-217	QCBS	Number of records before segmentation (count driven segmentation)
536-539	218-21B	QCBM	Records before message. Binary value representing the maximum number of list or punch data records that is to be tolerated by this job. When the record count exceeds the maximum value a warning message is output to the system operator.
540-543	21C-21F	QCBN	Records before next message. Additional number of list or punch data records that is to be tolerated by the job each time the record count exceeds the maximum value specified in the preceding field and the system operator elects to continue execution of the job.
544-545	220-221	QCER	Physical 3540 device address (packed)
546-547	222-223	QCJ#	Saved job number for accounting
548-551	224-227	QCCP	Compaction table name
			3800 Printer Control Information
552-555	228-22B	QCFL	Flush identifier
556-563	22C-233	QLCG	Copy groups
564	234	QLTC	Total number of transmission
565	235	QLC1	Current copy group index (restart purposes)
566	236	QLPS	Paper status (3800 only)
		QLBR	C'B' - burst threading
567	237	QLOP	Option byte
		QLCS	X'20' - no separator pages between copies
568-571	238-23B		Reserved for future use
572-583	23C-247		Unused

Figure 5.8. Disk Management Block (Part 7 of 9)

Bytes		Label of field	Description/function of field
Dec	Hex		
		The second portion (56 bytes) of the queue record contains control fields (information relating to the status of the queue record and to its position within the POWER/VS queues).	
584	248	QCXS	Execution switch X = job in execution b = job not in execution
585	249	QCFS	First in set switch

586	24A	QCSG	Segmentation type
			C = count driven segmentation
			P = program driven segmentation
			D = data driven segmentation
			b = no segmentation

587-599	24B-257		Reserved
600-607	258-25F	QCNS	Next record in set. (MBBCCHHR) [5] M = index in module control block address table in CAT
608-615	260-267	QCQP	Pointer to previous queue record (MBBCCHHR) [5] M = index in module control block address table in CAT The meaning of this pointer depends on the value of the contents in field QCFS. See Figure 5.48, part 9.
616-623	268-26F	QCQN	Pointer to next queue record (MBBCCHHR) [5] M = index in module control block address table in CAT The meaning of this pointer depends on the value of the contents in field QCFS. See Figure 5.48, part 9.
624-631	270-277	QCDF	Seek address of first data block (MBBCCHHR) M = index in module control block address table in CAT Seek address of the first read, list, or punch data block associated with the input or output described by this queue record.
632-639	278-27F		Reserved
		Master class table area Defines the status of the POWER/VS queues.	
640-1023	280-3FF	QCCT	
640-787	280-313	CTRT	Reader class area (37 entries, that is, 1 dummy entry and 36 entries 0-9 and A-Z)
788-937	314-3A7	CTLT	List class area (37 entries, that is, 11 dummy entries and 36 entries A-Z)
938-1083	3A8-43B	CTPT	Punch class area (37 entries, that is, 11 dummy entries and 36 entries A-Z)
1084-1088	43C-43F		Reserved
			The area layouts are shown in the dump format figure.

Figure 5.8. Disk Management Block (Part 8 of 9)

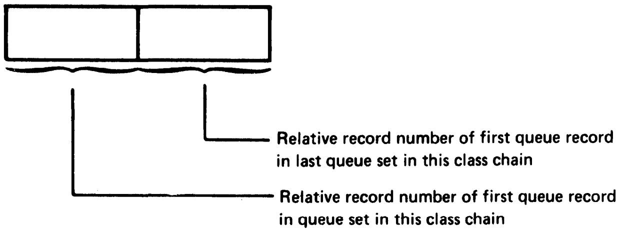
Bytes		Label of field	Description/function of field
Dec	Hex		
			Each entry is defined as a class list entry (DSECT=CTDS) and consists of the following two 2-byte fields:  Relative record number of first queue record in last queue set in this class chain Relative record number of first queue record in queue set in this class chain The high-order bit in the last field indicates whether there is a queue entry in this class that can be dispatched. (See Figure 5.48 for an illustration of this relationship.) Entries in the table also act as ECBs for the class chains (refer to Appendix D).

Figure 5.8. Disk Management Block (Part 9 of 9)

[1] This fullword contains a fixed-point binary value, representing the number of records per track characterising the DASD on which the file is located.

[2] This fullword contains a fixed-point binary value, representing the number of tracks per cylinder characterising the DASD on which the file is located.

[3] This code indicates that the corresponding queue entry was not affected by an abnormal POWER/VS termination. The DOS/VS jobs associated with the queue entry, however, could have been canceled via DOS/VS.

[4] The PSTOP cancel code will not be stored in an account record if the EOJ option was specified with the PSTOP command.

[5] Refer to note ([2]) in the description of the queue record area, where the contents of equivalent fields QRFS, QRNS, QRQP, and QRQN are explained.

TASK CONTROL BLOCK (TCB)

Definition macro: IPW$DTC

Each POWER/VS task is equipped with a task control block which is created in fixed storage and is used to establish the identity of the task and to preserve its status when it is not in active control of the central processor.

The TCB is divided into the following main areas:

- Task management fields
- Task register save area
- General task work area, linkage register save area, and file control words

Format of the TCB as it appears in a dump:

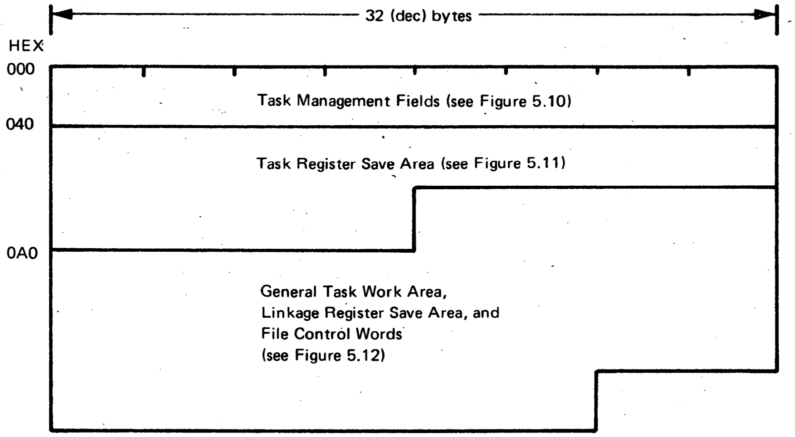

32 (dec) bytes
HEX
000
040
0A0
Task Management Fields (see Figure 5.10)
Task Register Save Area (see Figure 5.11)
General Task Work Area,
Linkage Register Save Area, and
File Control Words
(see Figure 5.12)

Figure 5.9. Task Control Block (Part 1 of 2)

When the TCB belongs to a command processor task, the general task work area, linkage register save area, and file control words are replaced by command processor control fields (see Figure 5.14), when it belongs to a BSC line manager task, these fields are replaced by BSC line manager control fields (see Figure 5.34), and when it belongs to an SNA manager task, these fields are replaced by SNA manager control fields (see Figure 5.40).

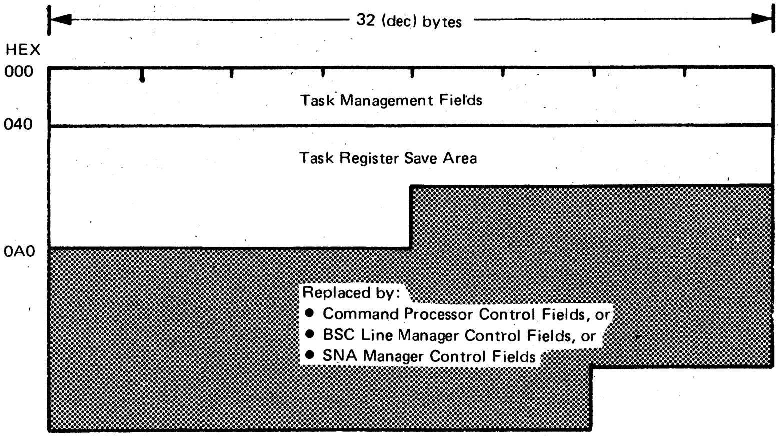
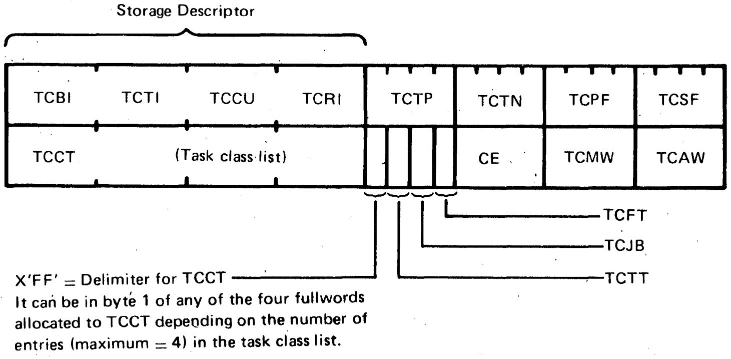

32 (dec) bytes
HEX
000
040
0A0
Task Management Fields
Task Register Save Area
Replaced by:
• Command Processor Control Fields, or
• BSC Line Manager Control Fields, or
• SNA Manager Control Fields

Figure 5.9. Task Control Block (Part 2 of 2)

<u>How to Locate</u>

Refer to Figure 6.1 in Section 6.

TCB - TASK MANAGEMENT FIELDS


Storage Descriptor
TCBI
TCTI
TCCU
TCRI
TCTP
TCTN
TCPF
TCSF
TCCT
(Task class list)
CE
TCMW
TCAW
TCFT
TCJB
TCTT
X'FF' = Delimiter for TCCT
It can be in byte 1 of any of the four fullwords allocated to TCCT depending on the number of entries (maximum = 4) in the task class list.

Bytes in TCB		Label of field (Note 1)	Description/function of Task management fields	
Dec	Hex			
		TCSD or ITSD or OCSD or TNSD or TPSD or TCBSD	The first 16 bytes contain the storage descriptor	
00-03	00-03	TCBI	Storage descriptor block ID (TCB)	
04-07	04-07	TCTI	Task ID	
			X'D6' (Q)	TCB belongs to a command processor task. Remaining 3 bytes are 'bCP'
			X'C9' (I)	TCB belongs to an initiator/terminator task. Remaining 3 bytes are 'bIT'
			X'E3' (T)	TCB belongs to the terminator task. Remaining 3 bytes are 'bTT'
			X'D9' (R)	TCB belongs to a local reader.
			X'E6' (W)	TCB belongs to a local writer.
			X'C5' (E)	TCB belongs to an execution processor task. In this case the next byte contains X'40', and the remaining bytes in the field indicate the partition that requested the task. For example, X'C6F1' = foreground 1 partition.
			X'F1'-'F9' (1-9)	TCB belongs to an RJE task. In this case the three remaining bytes will indicate the type of task. (RDR, LST, PUN, LGN, LGF, or MSG.)
			X'D3D9D3D4' ('LRLM')	TCB belongs to a line manager task.
			X'D740D7E2' ('PbPS')	TCB belongs to a status task.
			X'40C1C3E3' ('bACT')	TCB belongs to an account task.
			X'D1' (J)	TCB belongs to a spool management task, the three remaining bytes indicating the type of task (RDR, LST, or SPM).

Figure 5.10. Task Management Fields (Part 1 of 6)

Bytes in TCB		Label of field	Description/function of Task Management fields
Dec	Hex		
08-11	08-0B	TCCU	Physical device ID Physical unit address. If byte 0 of the task ID field = X'F1' - X'F9' (1-9), then TCCU contains the RJE line number, or 'SNA' for all RJE,SNA TCBs. 'PSP' (for RDR task) and 'GSP' (for LST task) are used respectively for PUTSPOOL and GETSPOOL/CTLSPOOL processing.
12-15	0C-0F	TCRI	Terminal ID Byte 0 = ID in binary format Byte 1 = ID in character decimal format Identifies the terminal ID requiring the task. If TCRI = binary zeros (0000), then task started as result of command invoked by the central operator.
			The following two fields form part of the task selection list (TSL). Task selection list is described in Section 2 (see also Appendix D).
16-19	10-13	TCTP	Address of task control block belonging to previous task in task selection list.
20-23	14-17	TCTN	Address of task control block belonging to the next task in task selection list. If the present is the last task control block in the chain, the address in TCTN is that of the wait control block.
24-27	18-1B	TCPF	Page fault request word. Contains page fault request information resulting from a page fault interrupt. Contents of GPR 13, passed from DOS/VS supervisor and saved for page management in the event of a page fault occurring during execution of the task. The field is set to binary zeros when no page fault request condition is present; hence, it will contain binary zeros during the time that the task is in control of the central processor.
28-31	1C-1F	TCSF	Task selection field. Byte 0 (the first byte in the field) = Task State Value <u>Task State Values</u> At any time, each task within the POWER/VS must be in one or another of a set of task states. The state of each task is defined by the single alphameric character in byte 28 of the associated task control block, and this in turn determines what action the task management routines must take when the task is examined for dispatch. Task states are normally set by the task itself whenever one of the task management macros is issued. The task management routines, the command processing task and the execution reader tasks are privileged, however, in that they may modify the task state of tasks other than themselves. <u>Note</u>: Task states can also be set by the page fault appendage routine.

Figure 5.10. Task Management Fields (Part 2 of 6)

<table>
<tr><th colspan="2">Bytes in TCB</th><th rowspan="2">Label of field (Note [1])</th><th rowspan="2">Description-function of Task Management fields</th></tr>
<tr><th>Dec</th><th>Hex</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><th>Task states</th><th>Hex</th><th>Char</th><th>Task condition</th><th>Label or Routine</th></tr>
<tr><td>Not dispatchable</td><td>C9</td><td>I</td><td>Task is inactive</td><td>TM10</td></tr>
<tr><td></td><td>D7</td><td>P</td><td>Page fault in process</td><td>TM10</td></tr>
<tr><td></td><td>D6</td><td>O</td><td>Waiting for operator response</td><td>TM10</td></tr>
<tr><td>Conditionally dispatchable</td><td>D3</td><td>L</td><td>Waiting for locked resource</td><td>TM30</td></tr>
<tr><td></td><td>C6</td><td>F</td><td>Waiting for the LTA</td><td>TM55</td></tr>
<tr><td></td><td>D4</td><td>M</td><td>Wait on multiple CCB or ECB posting [1]</td><td>TM50</td></tr>
<tr><td></td><td>D8</td><td>Q</td><td>As for M state, except event may never occur</td><td></td></tr>
<tr><td></td><td>C3</td><td>C</td><td>Wait on single CCB or ECB posting [2]</td><td>TM80</td></tr>
<tr><td></td><td>E2</td><td>S</td><td>As for C state, except event may never occur</td><td></td></tr>
<tr><td>Immediately dispatchable</td><td>C4</td><td>D</td><td>Dispatch task immediately</td><td>TM90</td></tr>
<tr><td>Running</td><td>D9</td><td>R</td><td>Task is running</td><td>Not appl.</td></tr>
<tr><td colspan="5">[1] or for an RJE task, waiting for a single ECB posting.

[2] or for an RJE task, waiting for a multiple ECB posting.</td></tr>
</table>
Bytes 1-3 = Address of the routine in the nucleus that tests for the condition indicated by the task state.</td></tr>
</table>

Figure 5.10. Task Management Fields (Part 3 of 6)

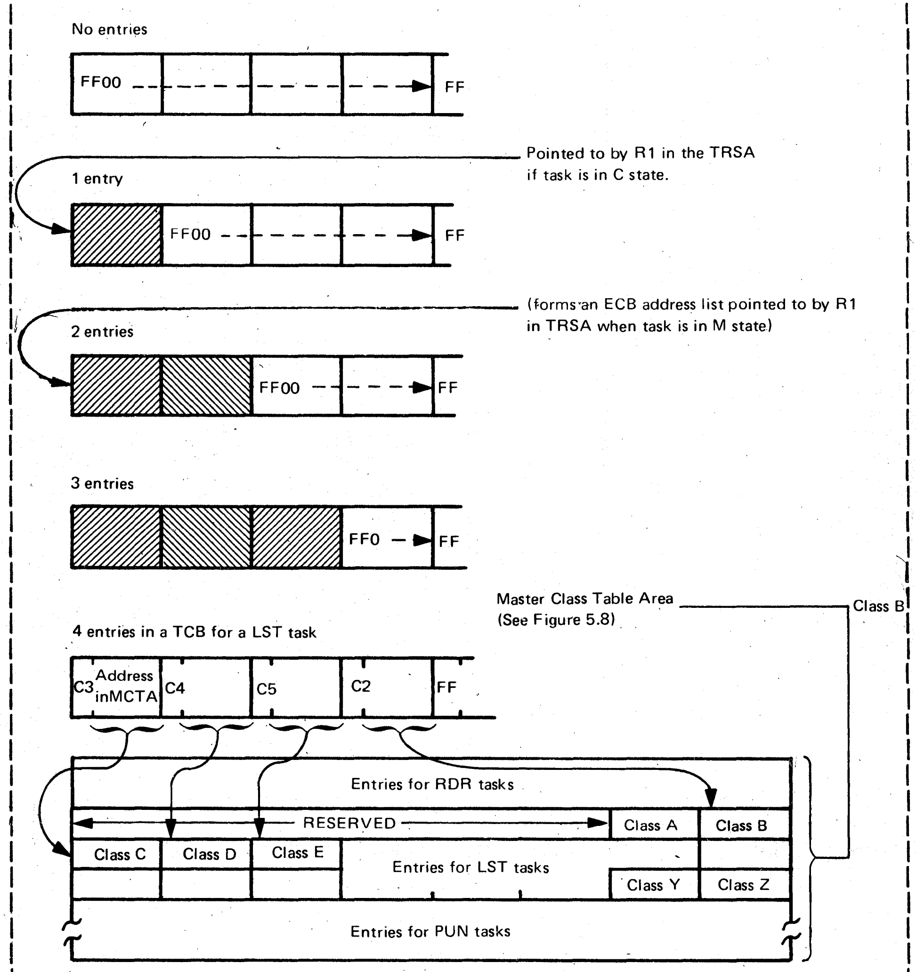
Bytes in TCE		Label of field	Description/function of Task Management fields
Dec	Hex		
32-48	20-30	TCCT	Task class list (plus a 1-byte field of X'FF') Up to four different classes can be specified simultaneously for any task, except RDR task. For each class identifying character an entry is made in the TCCT field in the TCB for that task. The first byte of each entry contains the class, and the remaining three bytes contain an address of an ECB in the master class table area (in DMB).  Pointed to by R1 in the TRSA if task is in C state. (forms an ECB address list pointed to by R1 in TRSA when task is in M state) Master Class Table Area (See Figure 5.8)
		Fields used by Spool Management Task	
43	2B	TCSS	Spool Management Switch
		TCIW	C'I' - Logical writer already initialized
		TCOW	C'O'-Open Logical Writer
		TCCW	C'C'-Close Logical Writer
44-47	2C-2F	TCER	Address of user's cross-partition XECB

Figure 5.10. Task Management Fields (Part 4 of 6)

<table>
<tr><th colspan="2">Bytes in TCB</th><th rowspan="2">Label of field</th><th rowspan="2">Description/function of Task Management fields</th></tr>
<tr><th>Dec</th><th>Hex</th></tr>
<tr><td>49</td><td>31</td><td>TCTT</td><td>Termination type
<table>
<tr><th>Hex</th><th>Char</th><th></th></tr>
<tr><td>40</td><td>(b)</td><td>Normal - continue execution</td></tr>
<tr><td>E4</td><td>(U)</td><td>Unrecoverable I/O error</td></tr>
<tr><td>E7</td><td>(X)</td><td>TASK cancel condition</td></tr>
<tr><td>C3</td><td>(C)</td><td>PCANCEL command issued</td></tr>
<tr><td>C6</td><td>(F)</td><td>PFLUSH command issued</td></tr>
<tr><td>C5</td><td>(E)</td><td>Stop at end of job</td></tr>
<tr><td>E2</td><td>(S)</td><td>Stop immediately</td></tr>
<tr><td>C8</td><td>(H)</td><td>PFLUSH with hold issued</td></tr>
<tr><td>D9</td><td>(R)</td><td>Stop immediately and restart</td></tr>
</table></td></tr>
<tr><td>50</td><td>32</td><td>TCJB</td><td>Job boundary switch

FF = Start of job
00 = Job boundary
80 = No job started yet</td></tr>
<tr><td>51</td><td>33</td><td>TCFT</td><td>Function track indicator.

This indicator is used by the task terminator phase (TR) to determine the appropriate action in case of an I/O error on the queue file or the data file. The following entries are possible:

<u>On input</u>:

X'D5' N - Get next from queue
X'C9' I - Open for input
X'C7' G - Get in process
X'C4' D - Delete in process
X'C3' C - Free pending
X'C6' F - Free in process
X'C5' E - End of queue action, awaiting accounting action
X'D3' L - Put account record in process
X'00' 0 - No entry active
or
X'40' b - No entry active

<u>On output</u>:

X'D9' R - Reserve queue in process
X'D6' O - Open for output
X'D7' P - Put in process
X'C1' A - Add to queue
X'C5' E - End of queue action, awaiting accounting action
X'D3' L - Put account record in process
X'00' 0 - No entry active
or
X'40' b - No entry active</td></tr>
</table>

Figure 5.10. Task Management Fields (Part 5 of 6)

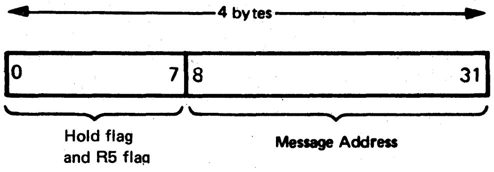
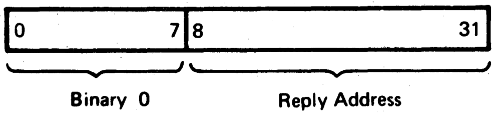
Bytes in TCB		Label of field	Description/function of Task Management fields
Dec	Hex		
52-55	34-37	TCEB	Task event control block (see Appendix D)
52	34	TCDB TCB2	Double buffering indicator C'2' - two buffers in use flag
53	35	TCCB TCAS	Function communication byte X'40' - asynchronous service user
54	36	TCEP TCEP	Event post byte X'80' - event post bit on setting
55	37		Reserved
		Each POWER/VS task that needs to perform input or output operations addressed to the system console must specify the operation required in the form of a message request word or a reply request word. These control fields are used to pass the necessary parameters for the operation of the message service routines.	
56-59	38-3B	TCMW	Message request word (see note [2] for message formats).  4 bytes 0 7 8 31 Hold flag and R5 flag Message Address The message address field contains the virtual address of the message control byte, that is, the byte that immediately precedes the text of the message to be output.
60-63	3C-3F	TCAW	Reply request word (see note [2] for message formats).  The reply address field contains the virtual address of the reply control byte, that is, the byte that immediately precedes the input area into which the reply is to be read. If no reply is to be made to the message, this field must contain binary zeros.

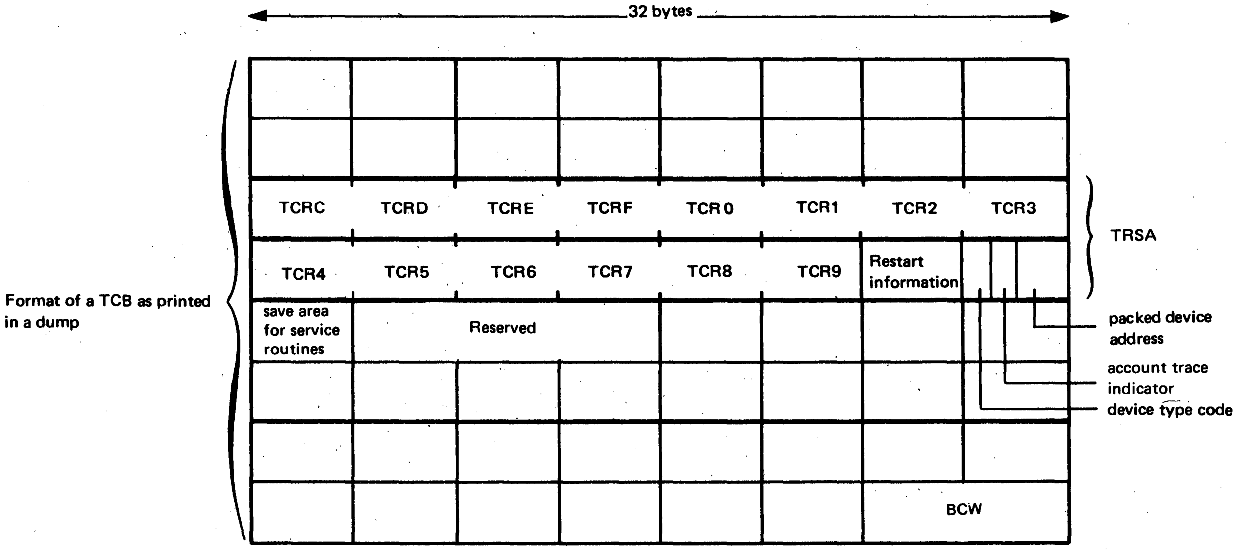
Figure 5.10. Task Management Fields (Part 6 of 6)

How to Locate

Refer to Figue 6.1 in Section 6.

TCB - TASK REGISTER SAVE AREA (TRSA)

The fields in this area in a TCB record the contents of registers 12 through 9 whenever entry is made to task selection. If the task state is set to R (running) the values in the fields record the contents of the registers when the task was most recently given control. If the task state is set to any other value the fields contain the current contents of the registers associated with the task.



Bytes in TCB		Label of field (Note 1)	Description/function of fields in TRSA
Dec	Hex		
		TCTR or TNTR or ITTR or OCTR or TPTR or TCBTR	
64-67	40-43	TCRC	Register 12 - asynchronous address register ('task PSW') Register 12 contains the address of the first instruction to be executed when the task is despatched. The first byte contains the condition code and the program mask bits in the form in which they are loaded by BAL instructions. (This is also true when the information is provided by the page fault appendage routines.)
68-71	44-47	TCRD	Register 13 - save area register Register 13 may contain the address of either the first (or only) or second linkage register save area depending on the hierarchy level of the caller.

Figure 5.11. Task Register Save area (Part 1 of 3)

Bytes in TCB		Label of field	Description/function of fields in TRSA
Dec	Hex		
72-75	48-4B	TCRE	Register 14 - linkage register Register 14 is used to contain the linkage address, that is, the address to which return is to be made when an exit linkage is next performed. When not required for this purpose the register is available for general use.
76-79	4C-4F	TCRF	Register 15 - entry point register Register 15 is used to address the entry point of the routine to be entered when an entry linkage is performed. This address is normally that of the storage descriptor which precedes the routine to be executed. The register may be conveniently used as the base register for the function to be executed. When not required for this purpose the register is available for general use.
80-83	50-53	TCR0	Register 0 - parameter and work register Register 0 is used to pass parameters to and from invoked routines. When not required for this purpose the register is available for general use.
84-87	54-57	TCR1	Register 1 - parameter and work register Register 1 may address a control block or control block list on which the task is at present waiting. For a task in C or S state it will point to a conventional DOS/VS CCB or a POWER/VS ECB. For a task in M or Q state, it will point to an ECB or CCB list. (Refer to Appendix D.)
88-91	58-5B	TCR2	Register 2 - linkage and work register Register 2 is used by service routines to retain the return address of the requesting task. It also has machine usage when a translate and test instruction is executed. When not required for these purposes the register is available for general task use.
92-95	5B-5F	TCR3	Register 3 - resource address register Register 3 may contain the address of a resource control block on which the task is at present waiting (task in L state). When not required for this purpose the register is available for general task use.
96-99	60-63	TCR4	Register 4 - work register
100-103	64-67	TCR5	Register 5 - work register If the task owns queue space, this register will address the queue record.
104-107	68-68	TCR6	Work register (may address the DMB). In an execution processor task, it addresses the partition control block.

Figure 5.11. Task Register Save area (Part 2 of 3)

Bytes in TCB		Label of field	Description/function of fields in TRSA
Dec	Hex		
108-111	6C-6F	TCR7	Work register. In an execution processor task this register addresses the user CCB.
112-115	70-73	TCR8	Work register. In an execution processor task this register addresses current channel command. In a physical routine, it points to PWS.
116-119	74-77	TCR9	Base register for highest level of code used by task.
120-123	78-7B	TCRS	Restart information This field contains an action type code in byte 0 and a value in bytes 1-3, as follows:
		TCRX	byte 0: (restart function index) X'04' restart at specified record (card or page) * X'08' skip forward specified number of records * X'0C' skip back specified number of records * (* set by PRESTART command) X'10' print specified number of pages (set by PSETUP command) X'14' restart at specified record (card or page) (set by PSTOP with RESTART option)
		TCRP	bytes 1-3: (restart page count) The number of records (cards or pages) to be acted upon.
124	7C	TCDT	Device type code of device in TCCU field
125	7D	TCAT	Account trace indicator: This indicator is used by the task terminator phase (TR) to determine the appropriate action in case of an I/O error on the account file. It can contain the following: X'D6' O - Open for reading account file X'C1' A - Caller active X'C7' G - Get in process X'C3' C - Close in process X'D2' K - Keep account file in process X'C5' E - Erase account file in process X'00' 0 - No entry active or X'40' b - No entry active
126-127	7E-7F	TCDE	Packed device address (of TCCU field)
128-131	80-83	TCRG	Save area for service routines
132-135	84-87	TCRH	Save area for service routine
136-139	88-8B	TC3E	Address of TCB extension area
140-143	8C-8F	TCPL	Address of Spool parameter list

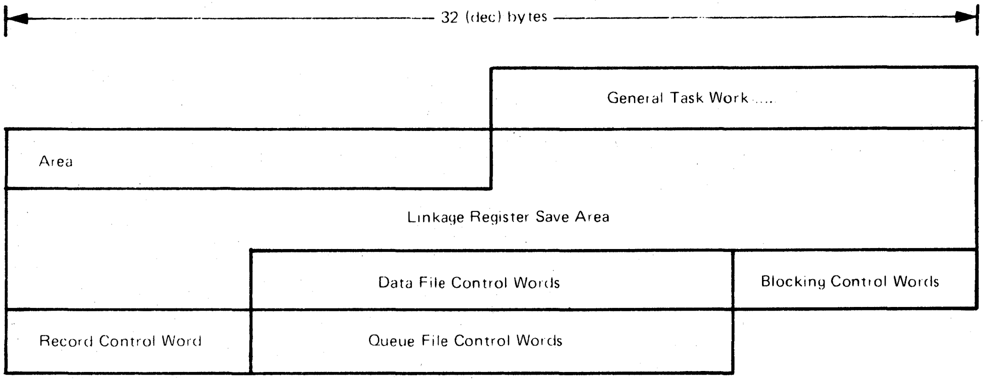
Figure 5.11. Task Register Save area (Part 3 of 3)

How to Locate

Refer to Figure 6.1 in Section 6.

TCB - GENERAL TASK WORK AREA, LINKAGE REGISTER SAVE AREA, AND FILE CONTROL WORDS

When the TCB belongs to a command processor task, this part of the TCB is replaced by command processor control fields (see Figure 5.13), when it belongs to a BSC line manager task, it is replaced by BSC line manager control fields (see Figure 5.34), and when it belongs to an SNA manager task, it is replaced by SNA manager control fields (see Figure 5.40).



Bytes in TCB		Label of field (Note 1)	Description/Function of Fields
Dec	Hex		
144-175	90-AF	General Task Work Area	
			This area may be broken into fields in whatever way is required by a task (for example, logical reader and writer work areas). It can also contain the 3540 communication byte (LWER): X'01' = card reader with a 3540 attached X'02' = reading from 3540 X'04' = 3540 data file processing
144-159	90-9F	TCGW	Used by logical routines
160-163	A0-A3	TCW1	Used by SNA routines as SUCB pointer
164-167	A4-A7	TCW2	Used by SNA routines as work area pointer
168-171	A8-AB	TCW3	Used by SNA routines as save area for register 13
172-175	AC-AF	TCW4	
160-167	A0-A7	TCJN	Spool management job name
168-171	A8-AB	TCXA	Address of error exit return in IPW$$SM
172	AC	TCSW	Switch byte
			Redefinition of the General Work Area Used by the Logical Writer
144-147	90-93	LWFI	Current forms identifier
148	94	LWNC	Current copy/transmission count
149	95	PPEB	Empty block indication
150	96	LWEJ	Restart switch X'80' restart at EOJ time
151	97	LWFT	Separator page flag C'1' - separator pages indicator
152-155	98-9B	LWAW	Address account counter workspace
156	9C	LWLC	Last command code
156-159	9C-9F	LWAD	Address of buffer for separator pages
160-167	A0-A7	LWFB	Current FCB name
168-170	A8-AA		Unused
171	AB	LWPS	Current paper threading (3800 only)
172-175	AC-AF	LWFH	Current flush identifier (3800 only)
		TCSV	Linkage Register Save Area (LRSA)

Figure 5.12. General Task Work Area, Linkage Register Save Area and File Control Words (Part 1 of 3)

Bytes in TCB		Label of field (Note [1])	Description/function of fields
Dec	Hex		
176-179	B0-B3	SVTC[4]	Task control block address
180-183	B4-B7	SVSV[4]	Previous save area address points to second of double LRSA
184-187	B8-BB	SVRE[4]	Saved register 14
188-191	BC-BF	SVRF[4]	Saved register 15
192-195	C0-C3	SVR0[4]	Saved register 0
196-199	C4-C7	SVR1[4]	Saved register 1
200-203	C8-CB	SVR2[4]	Saved register 2
204-207	CC-CF	SVR3[4]	Saved register 3
208-211	D0-D3	SVR4[4]	Saved register 4
212-215	D4-D7	SVR5[4]	Saved register 5
216-219	D8-DB	SVR6[4]	Saved register 6
220-223	DC-DF	SVR7[4]	Saved register 7
224-227	E0-E3	SVR8[4]	Saved register 8
228-231	E4-E7	SVR9[4]	Saved register 9
			Data File Control Words
232-239	E8-EF	TCDW	Data file seek address (MBBCCHHR) M = index into the module load address table. For tape spooling, this 8 byte field is defined as follows: Bytes 0 Tape flag (X'80') 1 Reserved 2-3 Length field 4-7 Address of tape control block
240-243	F0-F3	TCDA	Real data area address (see Note 3)
244-247	F4-F7	TCDV	Virtual data area address
			Blocking Control Words
248-251	F8-FB	TCBC	Residual block count
252-255	FC-FF	TCPR	Previous record address
		TCRW	Record Control Word (formed from CCW)
256	100	TCCC	Record command code
257-259	101-103	TCRV	Record address (virtual)
260	104	TCGP	General purpose byte (see Note 5) X'00' = normal record X'02' = 3540 data record X'04' = end of data X'08' = break record X'10' = end of block X'20' = end of 3540 data
261	105	TCG2	General purpose byte X'01' = TCB for read only operations X'08' = device end occurred (reader)
262-263	106-107	TCRL	Record length
			I/O (disk or tape) Request Word for Queue File

Figure 5.12. General Task Work Area, Linkage Register Save Area and File Control Words (Part 2 of 3)

Bytes in TCB		Label of field (Note[1])	Description/function of fields
Dec	Hex		
264-271	108-10F	TCQW	Queue file seek address (MBBCCHHR) M = X'04' For tape spooling this 8 byte field is defined as follows: Bytes 0 Tape flag 1 Reserved 2-3 Length field 4-7 Address of tape control block
272-275	110-113	TCQA	Real queue space address (see Note 3)
276-279	114-117	TCQV	Virtual queue space address
			Data File Control Words for 2nd Buffer (Note: 6)
			This extension of the TCB exists only when a list task has been started with 2 data file buffers.
280-287	118-11F	TC2DW	Data file seek address for 2nd buffer (MBBCCHHR). M = index into the module load address table.
288-291	120-123	TC2DA	Real data area address (see Note 3)
292-295	124-127	TC2DV	Virtual data area address
296-311	128-137		Reserved for future use
		TC2LN	Length of extended control block

Figure 5.12. General Task Work Area, Linkage Register Save Area and File Control Words (Part 3 of 3)

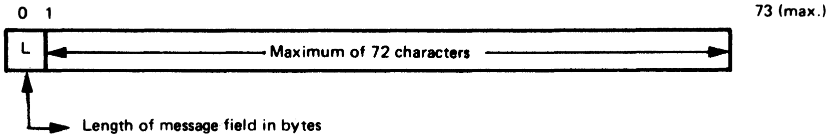
How to Locate

Refer to Figure 6.1 in Section 6.

Note [1]. The first characters of the labels in the control block vary according to the generated DSECT or declaration (PL/S).

- TC = Current TCB
- IT = Initiator/terminator TCB (used within the CSECT of NU).
- OC = Operator command processor (used within the CSECT of NU).
- TN = Used to address a TCB other than the task's own TCB. (To enable a task to address the TCB of another task.)
- TP = Used to address a TCB other than the task's own TCB. (To enable a task to address the TCB of another task.)
- TCB= Used to address a TCB other than the task's own TCB in the PL/S listings.

Note [2]. Message formats



The message field may contain variables for which the message modification routine in the message module (IPW$$MS) will substitute the appropriate fields.

Note [3]. The high-order byte of this field will contain the command code of the current or last executed operation.

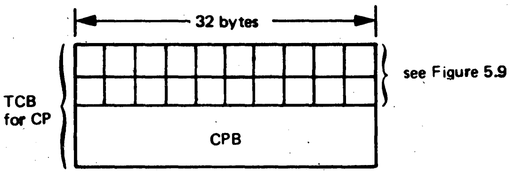
Note [4]. These labels refer to fields in a second LRSA described in Figure 5.14. The second LRSA has a format identical to that of the LRSA in the TCB.

Note [5]. Bit 7 of this byte may be set to 0 (to indicate no data transfer or card motion), or to 1 (to indicate data transfer or card motion).

Note [6]. The command processor has acquired a larger storage area for the extended TCB.

COMMAND PROCESSOR CONTROL BLOCK (CPB)

This block replaces part of a command processor TCB, when a command is entered via the console keyboard by the central operator, and of its associated temporary command processor TCB when linkage is made via the IPW$ICP macro.



CPB replaces file control fields, general task work area, and LRSA of standard TCB.

The contents of the CPB, in both cases, are described below:

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	CPSD	Storage descriptor (CPB)
16	10	CPID	RJE-userid (0 for local)
17-23	11-17	CPCM	Command code
24-95	18-5F	CPOP	Operands (free format)
96-103	60-67	CPNO	Sequence number (RJE only)
104-107	68-6B	CPEA	Address of caller ECB
108-135	6C-87		Reserved

Figure 5.13. Command Processor Control Block

How to Locate

Refer to Figure 6.1 in Section 6.

SECOND LRSA

Included by definition macro IPW$DSA for the save area.

A second LRSA is required by some tasks to link routines within the tasks. This second LRSA has the same format as the LRSA described in the TCB (see Figure 5.12).

RELATIONSHIP BETWEEN LRSAS

Linkage from a Physical Routine to a Logical Routine

Execution of the IPW$OLI macro instruction causes the creation of a second LRSA. The first LRSA is associated with the physical routine issuing the macro instruction (physical save), and the second LRSA is associated with the logical routine invoked by the macro instruction (logical save). The first fullword of each save area is initialized to address the TCB of the issuing task. The second fullword of each save area is initialized to address the other save area. The address of the internal routine entry point is stored in the third word of the internal routine save area.

The first and second LRSAs are collectively called a double linkage register save area (DLRSA). Linkage between the first and second LRSA in a DLRSA is shown in Figure 5.14, parts 1 and 2. Refer to DOS/VS POWER/VS Logic Part 2 (phases IPW$$PR, IPW$$PL, IPW$$PP) for the contents of the registers in the first LRSA.

Double Linkage Register Save Area (DLRSA)

Case 1 - where task is executing code in the physical routines (PR, PL, PP).

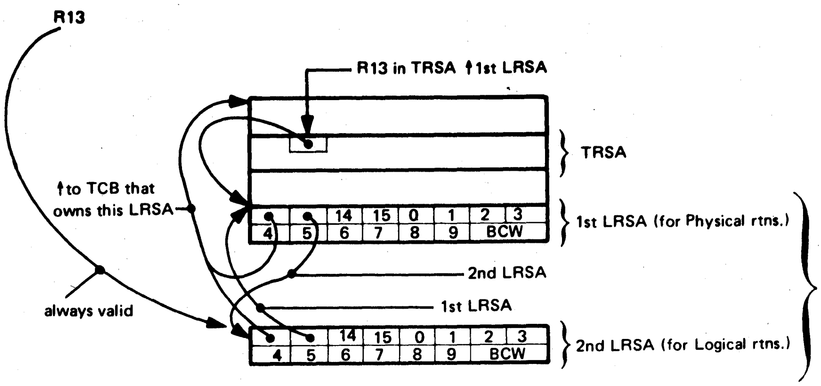


Figure 5.14. Part 1: Linkage between the two LRSAs in a Double Linkage Register Save Area (case 1)

Case 2 - where the task is executing code in the logical routines (LR, LW).

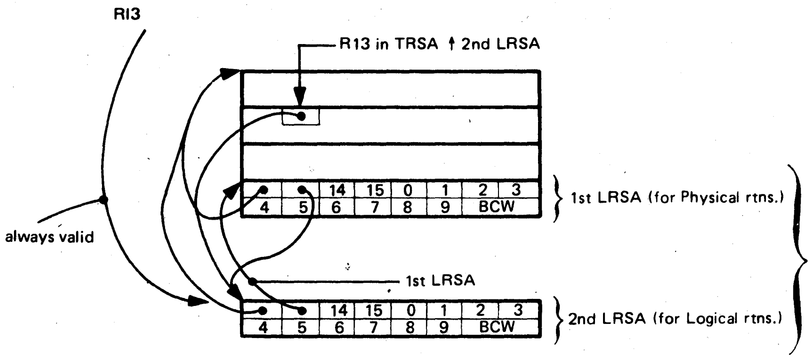


Figure 5.14. Part 2: Linkage between the two LRSAs in a Double Linkage Register Save Area (case 2)

Execution of the IPW$CLI macro instruction causes destruction of the interface linkage previously established by the IPW$OLI macro instruction and release of the second LRSA acquired by that instruction. Once the IPW$CLI macro instruction has been issued no further IPW$GLR or IPW$PLR macro instructions may be issued until the next IPW$OLI macro instruction is issued.

Linkage from a Logical Routine to a Function

Each POWER/VS function is coded as a unique control section. The first sixteen bytes of each control section consist of an alphameric control section descriptor. A fullword address constant containing the address of each control section is contained in the control address table.

Linkage to a function is achieved by loading register 15 with the address of the appropriate control section and then executing a Branch and Link instruction in the form BAL 14,16(15). Thus, entry is made to the control section at the first byte following the control section descriptor, the task return address being preserved in register 14.

Immediately upon entry the contents of registers 14 through 9 are saved in words 3 through 14 of the LRSA provided by the calling routine and addressed by register 13.

On return from a function, registers 14 through 9 are restored from the LRSA addressed by register 13. A branch is then made to the return address now contained in register 14. Refer to DOS/VS POWER/VS Logic Part 2 phases IPW$$LR and IPW$$LW for the contents of the registers in the second LRSA.

Linkage to a Function from an Execution Processor Routine

The LRSA of the TCB is also required by execution processors (XR, XW) in order to save the registers if the execution processor routine needs a function. Refer to Section 3 phases XR, XW descriptions for the contents of the first LRSA.

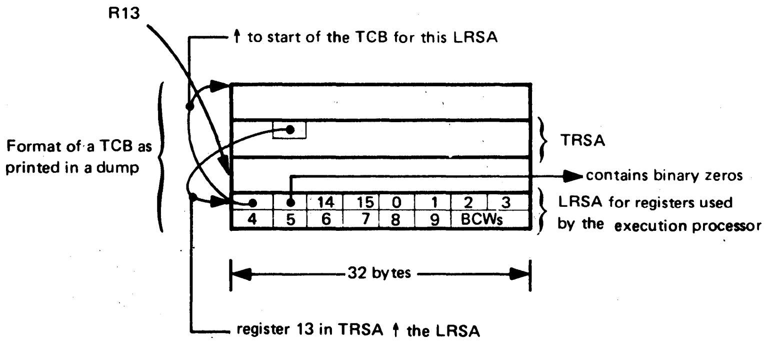

R13
↑ to start of the TCB for this LRSA
Format of a TCB as printed in a dump
TRSA
contains binary zeros
14 15 0 1 2 3
4 5 6 7 8 9 BCWs
LRSA for registers used by the execution processor
32 bytes
register 13 in TRSA ↑ the LRSA

Figure 5.14. Part 3: Linkage from an Execution Processor XR/XW routine

A different situation is when an execution processor task is executing code in XJ. In this case, a second LRSA is required as shown below:

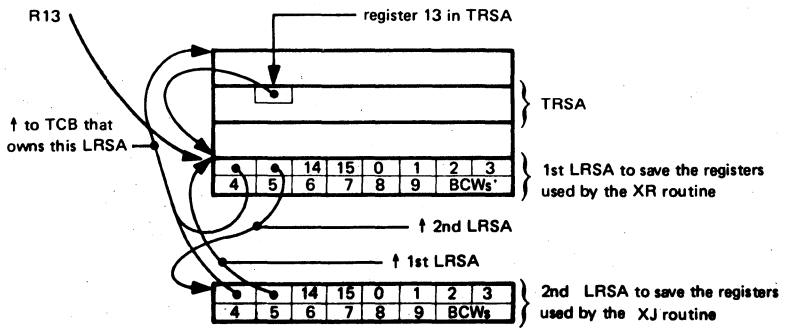

R13
register 13 in TRSA
TRSA
↑ to TCB that owns this LRSA
14 15 0 1 2 3
4 5 6 7 8 9 BCWs'
1st LRSA to save the registers used by the XR routine
↑ 2nd LRSA
↑ 1st LRSA
14 15 0 1 2 3
4 5 6 7 8 9 BCWs
2nd LRSA to save the registers used by the XJ routine

Figure 5.14. Part 4: Linkage from an XJ JECL Analysis routine

Refer to DOS/VS POWER/VS Logic Part 2 (phase XJ description) for the contents of the registers in the second LRSA.

Linkage to a Service

No registers are saved, other than in the TCB, when going from an

- external routine to a service, or from an

- internal routine to a service, or fiom a

- function to a service, except in the case of calling storage management, when registers 14 through 5 are stored in the SCB (see Figure 5.6), and in the case of calling message service wher registei 5 is stored in the MMB.

Note: Any service may use registers 0 through 3 destructively, (consult comments in program listings and/or the descriptions in DOS/VS POWER/VS Logic Part 2 for the service entered.)

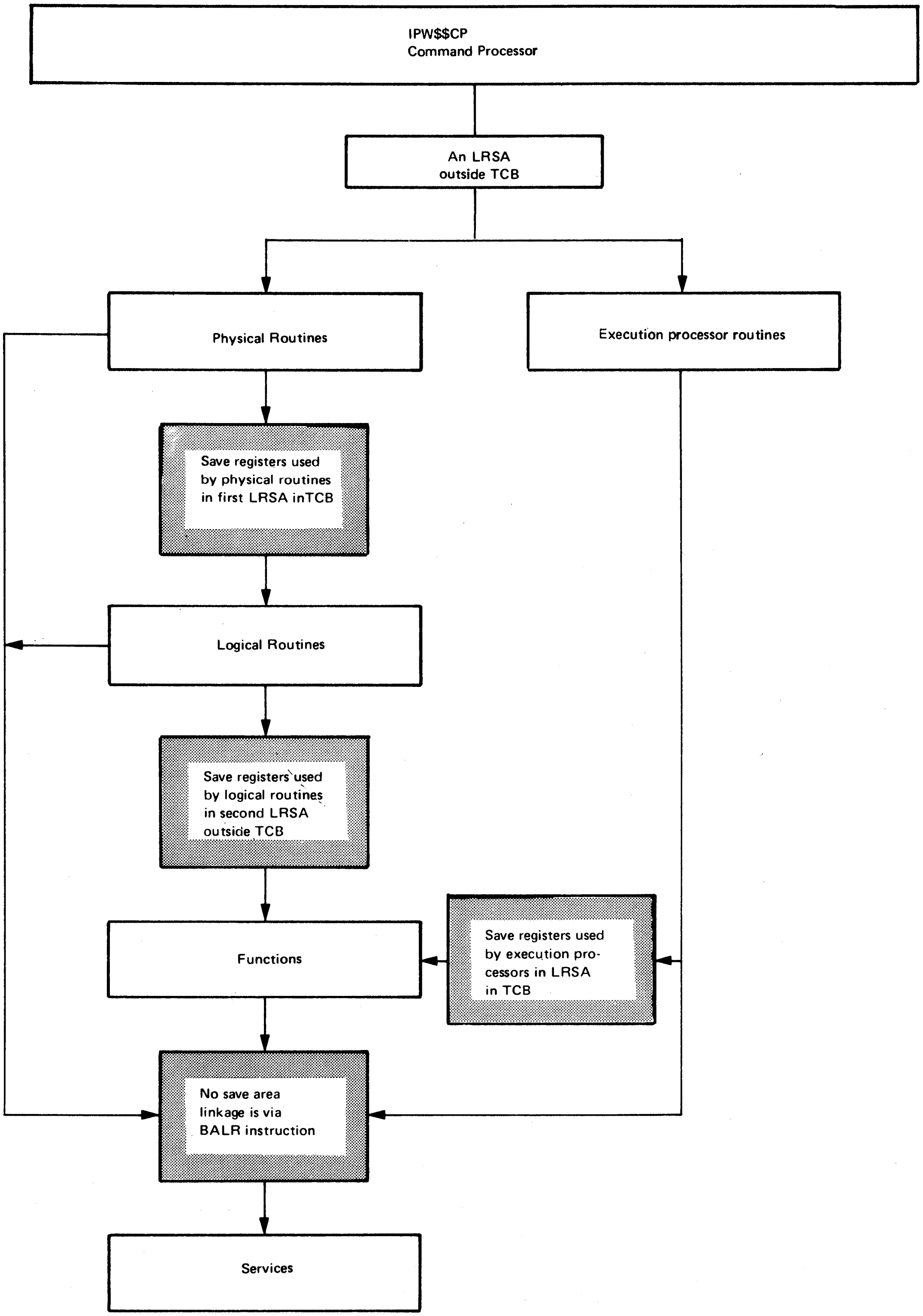


Figure 5.14. Part 5: Summary of Linkage Register Save areas

PHYSICAL WORK SPACE (PWS)

Definition macro: IPW$DPW

The physical work space is used to address and save the information necessary for reentrance of the physical reader/vriter. The area for PWS is reserved by the physical routine. It records information that points to a physical data area.

The relationship between the PWS and the PDA is illustrated in Figure 5.48.

Note: There is no PWS for an RJE task; it is replaced by information contained in the BCA (see Figure 5.33).

Bytes		Label of field	Description/function of field
Dec	Hex		
00-03	00-03	PBV1	Virtual address of the first PDA
04-07	04-07	PBR1	Real address of the first PDA
08-11	08-0B	PBV2	Virtual address of the second PDA
12-15	0C-0F	PBR2	Real address of the second PDA
16-19	10-13	PWVE	Virtual address of the active PDA
20-23	14-17	PWRE	Real address of the active PDA
24-25	18-19	PWLC	Displacement of last CCW in string from beginning of PDA
26-27	1A-1B	PWRL	Physical record length: to update the record pointer in the deblock routine
28-31	1C-1F	PWDI	Device type information
28	1C	PWDB	1 byte = single/double buffering (contains number of buffers)
29	1D	PWDT	1 byte = device type of unit record device
30-31	1E-1F	PWLU	2 bytes = LUB number
32-35	20-23	PWDV	Virtual address of end of PDA
36-39	24-27	PWDA	Real address of end of PDA
40-43	28-2C	PWCA	Real address of the first CCW
44	2C	PWOT	Operation byte
		PWWC	X'80' wait for completion request
45	2D	PWME	Message reply length
46-53	2E-34	PWRA	Message reply area
54-55	35-37		Reserved for future use

Figure 5.15. Physical Work Space (Part 1 of 2)

Bytes		Label of field	Description/function of field
Dec	Hex		
			3540 Extension
56-57	38-3B	PERA	Real address of the physical work space.
		PEDI	Device type indication.
60	3C		Reserved.
61	3D	PEDT	Device type.
62-63	3E-3F	PELU	Programmer logical unit.
		PEDP	Diskette parameters from PSTART.
64-71	40-47	PEFI	File identification.
		PEPS	PSTART parameters.
72	48		Reserved.
73	49	PEND	Number of diskettes to be read.
74	4A	PESC	Sequence check required.
75	4B	PEVE	Verify requested.
76-79	4C-4F	PECD	Displacement between real and virtual CCB addresses.
80-83	50-53	PECV	Address of 3540 CCB or physical data area.
84-87	54-57	PEDV	Virtual address of first 3540 data buffer.
88-91	58-5B	PEDA	Real address of first 3540 data buffer.
92-95	5C-5F	PEVN	Virtual address of second data buffer.
96-99	60-63	PERN	Real address of second data buffer.
100-103	64-67	PEBS	Real address of forced pre-SEEK CCW.
104-107	68-6B	PESK	Seek address (00CCHHRR).
108-111	6C-6F	PESO	Overlap seek address (00CCHHRR).
112-115	70-73	PELO	Extent lower limit (00CCHHRR).
116-119	74-77	PEED	Next sector address (00CCHHRR).
120-121	78-79	PERL	Record length.
122-123	7A-7B	PENN	Number of buffers allocated in second data buffer.
		PESI	Sequence identification.
124	7C	PEMI	Multivolume identification.
125	7D	PESN	Volume sequence number.
126	7E	PEOC	Open return code.
127	7F	PEOD	Number of opened diskettes.
128-135	80-87	PEDW	Double word for conversion purposes.
136-151	88-97		Reserved.

Figure 5.15. Physical Work Space (Part 2 of 2)

How to Locate

Refer to Figure 6.1 in Section 6.

PHYSICAL DATA RECORD AREA (PDA)

Space for this area is reserved during the execution of a physical reader/writer routine. The size of the area depends on the specifications in the DBLK parameter. It consists of a CCB and a CCW string which constitutes the channel program, followed by 80 byte areas which contain the input or output data records.

Note: For an RJE task the CCB and the channel program is in the BCA.

During a read operation the area is initialized by calculating the amount of data records and their CCWs that will fit in the area. Then an SVC 0 is issued to commence the I/O operation to read cards or 80 byte records into it. When it is full, the data is transferred to the logical data area by the function IPW$PLR and is ready for output to the spooling device assigned as the data file. Queue records are constructed on the queue file to record the seek addresses of the data on the data file.

During a write operation, the reverse occurs. Data is read from the spooling device to the LDA from where it is transferred to this PDA ready for the physical routine to print or punch the data.

Figure 5.48 illustrates this relationship for an RJE task and for a local task.

How to Locate

Refer to Figure 6.1 in Section 6.

LOGICAL DATA RECORD AREA (LDA)

This area is used to hold data which is to be written to the data file (read operation) and read from the data file (write operation). Its size is set by the DBLK parameter.

Records are transferred to the IDA one at a time from the PDA for read and for write operations. When the LDA is full, or there is no more room for a complete record, the information is written to or read from the data file. It is addressed via the I/O request word in the TCB, and each record is addressed via the channel program in the MCB for the data file. Figure 5.48 illustrates this relationship.

The format of a data record is shown below:

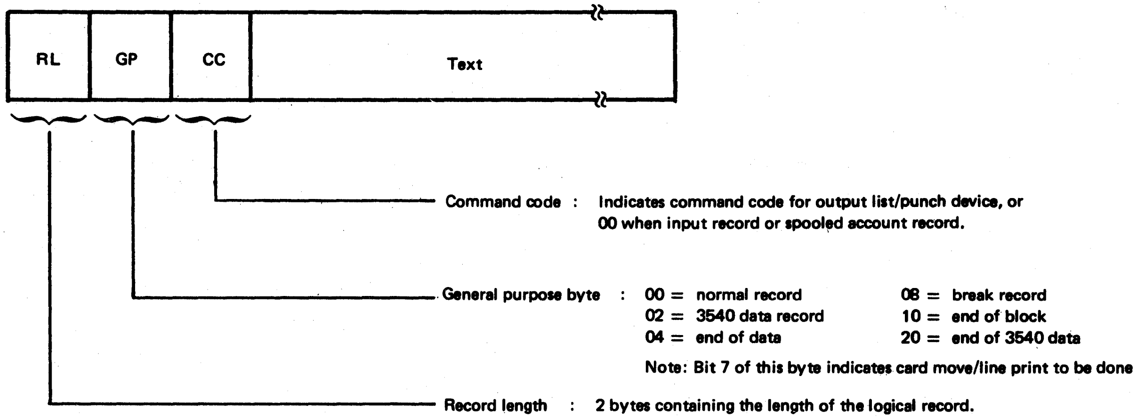

RL
GP
CC
Text
Command code : Indicates command code for output list/punch device, or 00 when input record or spooled account record.
General purpose byte : 00 = normal record
02 = 3540 data record
04 = end of data
08 = break record
10 = end of block
20 = end of 3540 data
Note: Bit 7 of this byte indicates card move/line print to be done
Record length : 2 bytes containing the length of the logical record.

Figure 5.16. Logical Data Record

How to Locate

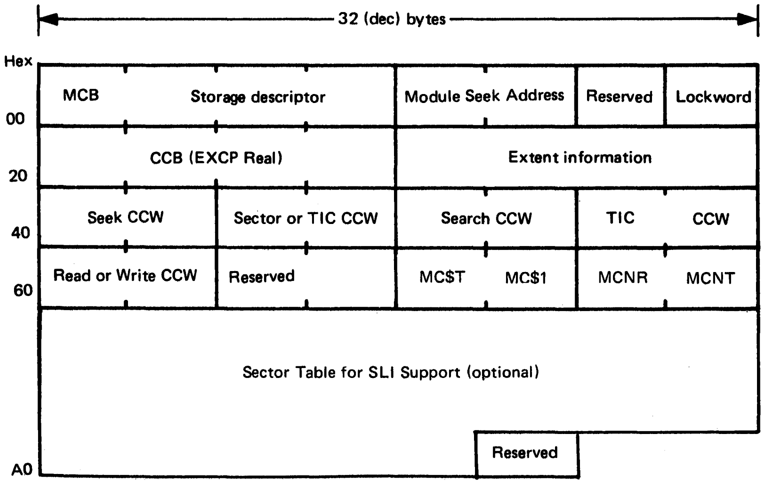
Refer to Figure 6.1 in Section 6.

MODULE CONTROL BLOCK (MCB)

Definition macro: IPW$DMC

Each module (an extent, always 1 for queue file and at least 1 for the data file) requires an MCB. The format and type of information contained in any MCB is identical.

The format of a module control block as it is printed in a dump is shown below:



Bytes		Label of field (Note [1])	Description/function of field
Dec	Hex		
00-15	00-0F	MCSD[1] or	Storage descriptor MCB QFILE1 cuu (SYS001)
		D2SD[1]	Storage descriptor MCB DFILE2 cuu (SYS002)
		MCSD[1]	Storage descriptor MCB DFILE3 cuu (SYS003)
		MCSD[1]	Storage descriptor MCB DFILE4 cuu (SYS004)
		MCSD[1]	Storage descriptor MCB DFILE5 cuu (SYS005)
		MCSD[1]	Storage descriptor MCB DFILE6 cuu (SYS006)
		MCSD[1]	Storage descriptor MCB LFILE7 cuu (SSL)
		MCSD[1]	Storage descriptor MCB LFILE8 cuu (PVTSSL)
16-23	10-17	MCSA	Module seek address (MBBCCHHR)[2]
24-27	18-1B		Reserved
28-31	1C-1F	MCLK	Lockword
		MCCB	Command control block
32-33	20-21	MCCT	Residual count
34-35	22-23	MCCM	Communication bytes
36-37	24-25	MCST	Device status
38-39	26-27	MCLU	EXCP real plus LUB index (logical unit)
40-43	28-2B	MCCA	CCW address
44-47	2C-2F	None	CCW address in CSW
		MCXT	Extent information
48-51	30-33	MCLO	Low limit (CCHH)
52-55	34-37	MCHI	High limit (CCHH)
56	38	MCSE	Sector value
57-59	39-3B		Reserved
60-63	3C-3F	MCSX	Sector table address
		MCCH	Channel program

Figure 5.17 (Part 1 of 2). Module Control Block

Bytes		Label of field (Note [1])	Description/function of field
Dec	Hex		
64-71	40-47	MCSK	Seek CCW
72-79	48-4F	MCSS	Set sector or TIC CCW
80-87	50-57	MCSH	Search CCW
88-91	58-5B	MCTI	TIC CCW
92-95	5C-5F	MCTV	Virtual address of buffer
96-103	60-67	MCRW	Read or write CCW
104-107	68-6B	MC$T	Owner of i/o request (↑ TCB)
108-111	6C-6F	MC$1	Save area for register 1
112-179	70-1B7		SLI support
112-115	70-73	MCNR	SSL - No. of records/track
116-119	74-77	MCNT	SSL - No. of tracks/cylinder
120-179	78-B3	MCTS	Extension to sector table SSL (optional)
180-183	B4-B7		Reserved

Figure 5.17. Module Control Block (Part 2 of 2)

[1] The labels in this control block vary according to the generated DSECT or declaration. The first characters are Q1 for the queue file MCB, D2 for the DFILE2 MCB, and MC for all other MCBs.

[2] Seek and search address required by the channel program. Whenever an input or output operation is to be performed it is updated from the seek address pointer in the I/O Request Word (see Figure 5.12) that controls the operation.

How to Locate

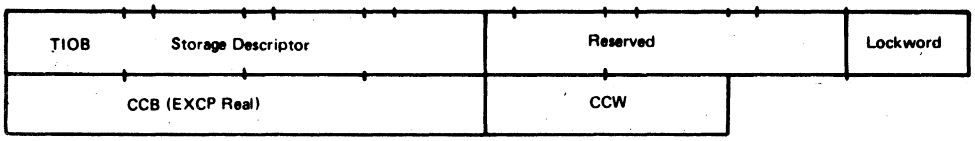
Refer to Figure 6.1 in Section 6.

TAPE CONTROL BLOCK (TBB)

Definition macro: IPW$DTB

This block is dynamically created to satisfy requirements of POWER/VS tasks utilizing tape as intermediate storage. Its format as it is printed in a dump is shown below:

Dump format:



Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	TBSD	Storage descriptor (TBB)
16-27	10-1B		Reserved
28-31	1C-1F	TBLK	Lockword
		TBCB	Command control block
32-33	20-21	TBCT	Residual count
34-35	22-23	TBCM	Communication bytes
36-37	24-25	TBST	Channel and device status
38-39	26-27	TBLU	EXCP real plus LUB index
40-43	28-2B	TBCA	CCW address
44-47	2C-2F	None	CCW address in CSW
48-55	30-37	TBCH	Read or write CCW

Figure 5.18. Tape Control Block

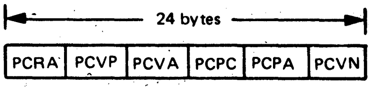
How to Locate

Refer to Figure 6.1 in Section 6.

PAGE CONTROL BLOCK (PCB)

Definition macro: IPW$DPC

Each page in the fixable area starts with this control block, which occupies the first 24 bytes of the page. The format of the PCB as it is printed in a dump is shown below:



Bytes		Label of field	Description/function of field
Dec	Hex		
00-03	00-03	PCRA	Page real storage address Real storage address of the page described by this PCB.
04-07	04-07	PCVP	Previous page virtual address Virtual storage address of the previous page in the fixed page list. If the present page is the first page in the fixed page list the word is set to binary zeros.
08-11	08-0B	PCVA	This page virtual address This fullword contains the virtual storage address of the page described by this page control block.
12-15	0C-0F	PCPC	Page control byte address Contains the virtual storage address of the byte within the storage assignment block in the storage control block which corresponds to the present page.
16-19	10-13	PCFA	First buffer address This fullword contains the virtual storage address of the first storage buffer within the present page.
20-23	14-17	PCVN	Next page virtual address This fullword contains the virtual storage address of the next page in the fixed page list. If the present page is the last page in the fixed page list the word is set to binary zeros.

Figure 5.19. Page Control Block

This is a useful aid to convert the virtual address of a page to real when analyzing a standalone dump.

For example, assume the contents of register 1 in a TCB within a page starting at real address E800 is 41AF4. Assume also that field PCVA of the page control block contains 41800. To find the real address contained in register 1:

1. 41AF4 lies within the page starting at virtual address 41800.

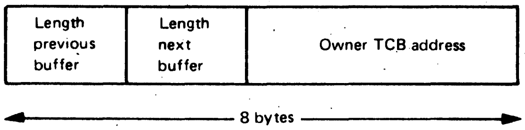
2. Subtract that address (41800) from 41AF4 and add the result to the address in field PCRA of the page control block, (E800).

3. The result is the corresponding real address of the contents of register 1 (EAF4).

BUFFER CONTROL WORD (BCW)

When a page is fixed in the fixable area, storage management assigns the first and last buffer control words. The first buffer control word is placed immediately after the page control block at the start of the page in real storage, and the last buffer control word is placed in the last two words of the page.

Real storage within the page is allocated by storage management from the last buffer control word. When storage is allocated to a buffer, the last buffer control word is updated to reflect the size of the buffer, and a new buffer control word is created to immediately precede the buffer. The newly created buffer control word will be used by storage management next time it requires space in the fixable area. This is more fully described in Section 2 of this manual and DOS/VS POWER/VS Logic Part 2. Its format as it is printed in a dump is shown below:



Bytes		Label of field	Description/function of field
Dec	Hex		
00-01	00-01	None	Length of previous buffer This halfword contains the binary length of the immediately-preceding storage buffer. If the buffer is in use its length is stored in twos complement form. If the buffer is not in use its length is stored in normal form. If the present buffer is the first in the page the word is set to binary zeros.
02-03	02-03	None	Length of next buffer This halfword contains the binary length of the present storage buffer, that is, the buffer which immediately follows this buffer control word in storage. If the buffer is in use its length is stored in twos complement form. If the buffer is not in use its length is stored in normal form. If the preceding buffer is the last in the page the word is set to binary zeros.
04-07	04-07	None	Owner (TCB virtual address) of next buffer. This fullword contains the address of the TCB belonging to the task which issued the request for buffer space. If a TCB is contained in the buffer, the owner address is that of the task which built the TCB.

Figure 5.20. Buffer Control Word (Part 1 of 2)

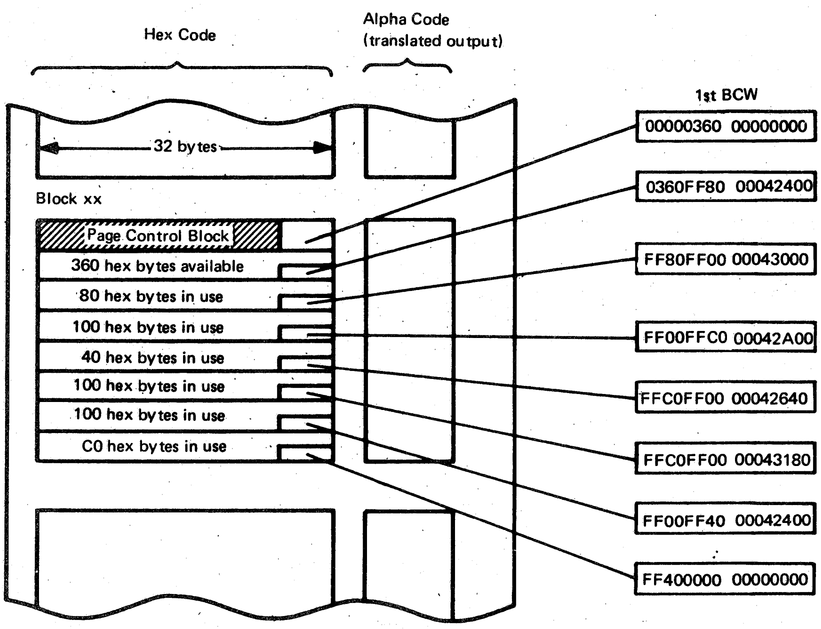


Figure 5.20. Buffer Control Word (Part 2 of 2)

This figure shows how to interpret BCWs in a standalone dump output. It illustrates a page containing seven buffer areas which contain control blocks that existed in the fixable area at the time the dump was executed. The size of each buffer in use can be seen recorded in twos complement form in the BCWs, as well as the task which issued the request for buffer space.

PARTITION CONTROL BLOCK (PDB)

Definition macro: IPW$DPD

A partition control block is dynamically created for each partition to be controlled by POWER/VS. In addition to general partition information, the block contains an entry for each device that is to be spooled. The format of these entries is described by the IPW$DTL macro instruction.

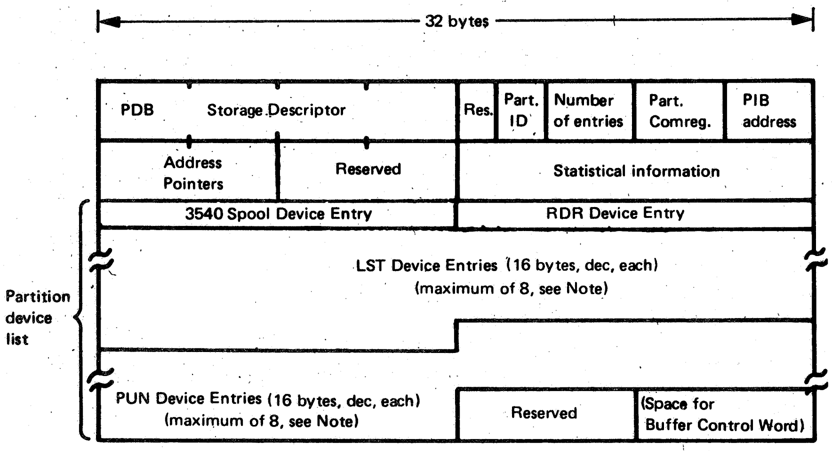


Figure 5.21. Partition Control Block (Part 1 of 3)

Note: The number of entries in the LST and PUN device entry sections of this block depend on the number specified in response to the message:

```
1QxxI PLEASE SPECIFY SPOOL DEVICES
```

after entering the PSTART command to start a partition.

Bytes		Label of field[1]	Description/function of field
Dec	Hex		
00-15	00-0F	PSSD	Storage descriptor (PDB)
16-17	10-11		Reserved
18-19	12-13	PDPI	Partition identifier
20-23	14-17	PDNE	Number of entries
24-27	18-1B	PDCM	Partition comreg address
28-31	1C-1F	PDPB	PIB address
32-35	20-23	PDPA	First entry address
36-39	24-27	PDBB	Boundary box entry pointer
40-47	28-2F		Reserved.
		Statistical information This information is destined for the execution account record and there is a pointer to the SLI work area.	
48-51	30-33	PDSL	Pointer to SLI work area
52-55	34-37	PD#L	Number of lines spooled
56-59	38-3B	PD#C	Number of cards spooled
60-61	3C-3D	PD#P	Number of pages spooled
62	3E	PDOC	Default output class
63	3F	PDMT	Multitasking indicator
		3540 spool device entry (same format as RDR device entry)	
64-79	40-4F	PDER	
		RDR device entry (maximum = 1)	
80-83	50-53	PDPU	Address of entry in the DOS/VS PUB for a card reader device
84-87	54-57	PDTC	Address of execution reader TCB
88-91	58-5B	PDCB	CCB address. The first byte of this field is the SVC code: X'00' = SVC 0: I/O request by user program X'90' = SVC 90: accounting request by PA X'91' = SVC 91: accounting request by JCL
92	5C	PDDT	Device type code
93	5D	PDCL	Device class code can be R = normal reader, or C = console
94-95	5E-5F	PDRQ	Requestor ID
		LST device entry (maximum = 8)	
96-99	60-63	TLPU[2]	Address of entry in the DOS/VS PUB for a printer device
100-103	64-67	TLTC[2]	Address of the execution list TCB
104-107	68-6B	TLCB	CCB address
108	6C	TLDT[2]	Device type code
109	6D	TLCL	For list device entry this can be L = device is being spooled, N = device is not being spooled.
110-111	6E-6F	TLRQ	Requestor ID

Figure 5.21. Partition Control Block (Part 2 of 3)

Bytes		Label of field	Description/function of field
Dec	Hex		
Depends on number of LST entries		PUN device entry (maximum = 8) (same format as LST device entry)	
		TLPU	Address of entry in the DOS/VS PUB for a punch device
		TLTC	Address of the execution punch TCB
		TLCB	CCB address
		TLDT	Device type code
		TLCL	For punch device entry this can be P = device is being spooled, N = device is not being spooled.
		TLRQ	Requestor ID

Figure 5.21. Partition Control Block (Part 3 of 3)

How to Locate

Refer to Figure 6.1 in Section 6.

QUEUE RECORD AREA (QRA)

Definition macro: IPW$DQR

This area is used in conjunction with the auxiliary queue record area in the disk management block. Each task that processes a queue record acquires a QRA to contain the record.

The format as it is printed in a dump is shown below:

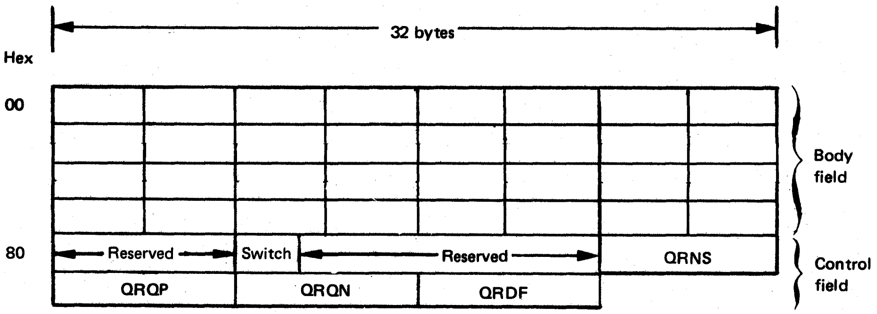


Figure 5.22 Queue Record area (Part 1 of 3)

Bytes		Label of field[1]	Description/function of field. Refer to the Disk Management Block (DMB) auxiliary queue record area for a fuller description of the individual entries.
Dec	Hex		
		Body Fields (first 136 bytes) referenced by label QRBF The body of the queue record contains information pertinent to this particular queue entry and the user job which created it.	
00-07	00-07	QRDY[1]	Date
08-11	08-0B	QRST	Operation start time
12-15	0C-0F	QRET	Operation end time
16-31	10-1F	QRUI	User information
32-39	20-27	QRNM	Job name
40-41	28-29	QRNO	Job number
42	2A	QRQI	Queue record identifier
43	2B	QRCN	POWER/VS cancel code
44	2C	QRRJ	Line identifier or device type
45-47	20-2F	QRCU	Channel and unit (line address)
48	30	QRFJ	From terminal identifier
49	31	QRTJ	To terminal identifier
50	32	QRCL	Class
51	33	QRPY	Priority
52-55	34-37	QRNR	Record count
56-57	38-39	QRNT	Number of tracks
58	3A	QRSN	Job suffix number
59	3B	QRNC	Number of copies
60-63	3C-3F	QRFI	Forms identifier
64-67	40-43	QRNA	Number of additional records
68-69	44-45	QRNP	Number of pages
70-71	46-47	QRNE	Number of extra pages
72-75	48-4B	QRLC	Line/card counter
76-79	4C-4F	QRRR	Restart page count
80	50	QRCR	Copies remaining
81	51	QRDI	Not used
82	52	QRDP	Disposition
83	53	QRSP	Number of separators
84-87	54-57	QRBS	Number of records before split
88-91	58-5B	QRBM	Maximum value of count
92-95	5C-5F	QRBN	Additional count value
96-97	60-61	QRER	Physical 3540 device address (packed)
98-99	62-63	QRJ#	Save job number for accounting
100-103	64-67	QRCP	Compaction table name
			3800 Printer Control Information
104-107	68-6B	QRFL	Flush identifier
108-115	6C-73	QRCG	Copy groups
116	74	QRTC	Total number of transmissions
117	75	QRCI	Current copy group index (restart purposes)
118	76	QRPS	Paper status (3800 only)
		QRBR	C'B' burst threading
119	77	QROP	Option byte
		QRCS	X'20' no separator pages between copies
120-123	78-7B		Reserved for future use
124-135	7C-87		Unused

Figure 5.22 Queue Record area (Part 2 of 3)

[1] The labels in a queue record vary according to the generated DSECT. The first two characters are queue record in a present queue record, QN in a "next" queue record, and QP in a "previous" queue record.

Bytes		Label of field[1]	Description/function of field. Refer to the Disk Management Block (DMB) auxiliary queue record area for a fuller description of the individual entries.
Dec	Hex		
		Control Fields (48 bytes) referenced by label QRCF The control portion of the queue record contains information relating to the status of the queue record and to its position within the POWER/VS queues.	
136	88	QRXS	Execution switch
137	89	QRFS	First in set switch [2]
138	8A	QRSG	Segmentation type
139-151	8B-97		Reserved
152-159	98-9F	QRNS	Next record in set [2]
160-167	A0-A7	QRQP	Previous set in queue [2]
168-175	A8-AF	QRQN	Next set in queue [2]
176-183	B0-B7	QRDF	First block of data

Figure 5.22 Queue Record area (Part 3 of 3)

[2] The contents of field QRFS can be X'00' or X'01'.

X'01' indicates that the queue record is the first in the queue set.
X'00' indicates that the queue record is not the first in the queue set.

This determines the meaning of the fields QRNS, QRQP, and QRQN as follows:

QRFS = X'00' (This queue record is not first in Q set)

Label of Field	Field contains zero	Field does not contain zero
QRNS	This queue record is the last in this queue set.	It is seek address of next queue record in this queue set.
QRQP	Cannot be zero.	It is the seek address of the first queue record in this queue set.
QRQN	Must be zero.	Must be zero.

QRFS = X'01' (This queue record is first in queue set)

Label of Field	Field contains zero	Field does not contain zero
QRNS	This queue record is last in same queue set.	It is the seek address of the next queue record in this queue set.
QRQP	This queue record is the first in the first queue set in this class chain.	It is the seek address of the first queue record of the previous queue set.
QRQN	This queue record is the first queue record of the last queue set in this class chain.	It is the seek address of the next queue set in this class chain.

This is illustrated in Figure 5.48, Part 9

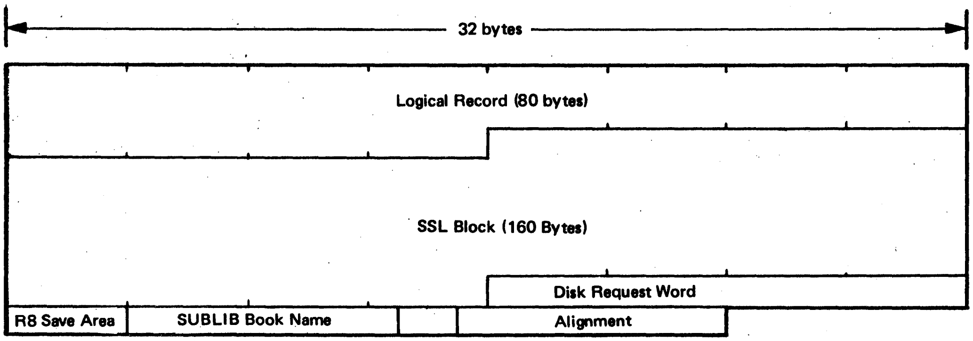
How to Locate

Refer to Figure 6.1 in Section 6.

SLI WORK SPACE (SLW)

Definition macro: IPW$DSL

This work space is reserved and used by phase IPW$$SL and provides storage to read blocks from a source statement library and to deblock the records. The format of the block as printed in a dump is shown below:



Bytes		Label of field in IPW$$NU	Description/function of field
Dec	Hex		
00-79	00-49	SLLR	Logical record work area
80-239	50-E9	SLPB	SSL block work area
		SLRW	Disk request word
240-247 248-251 252-255	F0-F7 F8-FB FC-FF	SLSW SLSA SLSV	Seek address (MBBCCHHR) M = index in module control block address table in CAT. Real address read-in area Virtual address read-in area
256-259	100-103	SLR8	Register 8 save area
		SLNM	SLI sublibrary and bookname
260 261-268	104 105-10C	SLSL SLBM	Sublibrary name Bookname
269 270	10D 10E	SLRS SLRR	Read SSL switch Read RDR switch
271-279	10F-117		Filler for alignment

Figure 5.23. SLI Work Space

How to Locate

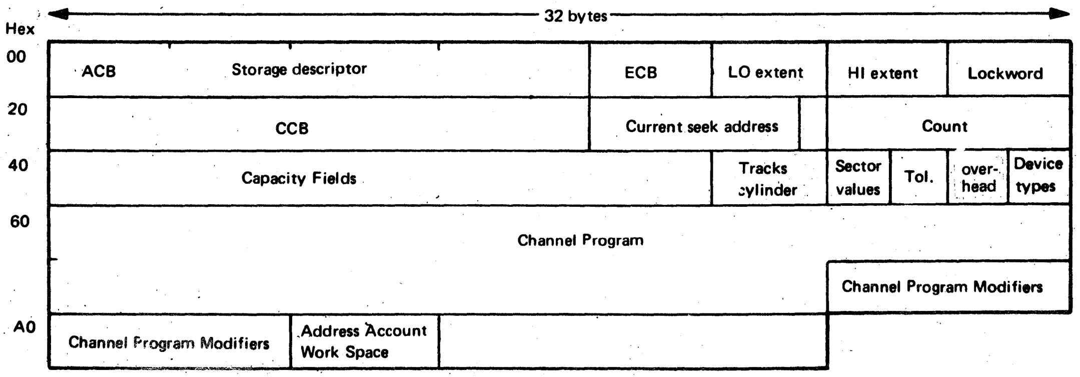
Refer to Figure 6.1 in Section 6.

ACCOUNT CONTROL BLOCK (ACB)

Definition macro: IPW$DAC

The ACB is used only by job accounting support. It is used to control account records contained on the account file 'IJAFILE' (SYS000).

The ACB is initialized for PUT mode. The mode is changed into GET mode when the save account task issues a IPW$OAF macro. The format of the block as printed in a dump is shown below:



Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	ACSD	Storage descriptor (ACB)
16-19	10-13	ACEB	Event control block This ECB is posted when the account file is emptied.
20-23	14-17	ACLO	Extent lower limit
24-27	18-1B	ACHI	Extent upper limit
28-31	1C-1F	ACLW	Lockword
		Command Control Block (referred to by label ACCB)	
32-33 34-35 36-37 38-39 40 41-43 44 45-47	20-21 22-23 24-25 26-27 28 29-2B 2C 2D-2F	ACCT ACCM ACST ACLU None ACCA None ACCS	Residual count Communication bytes Device status Logical unit Reserved for LIOCS CCW real address Reserved for PIOCS CCW address in CSW
48-54 55	30-36 37	ACSA None	Current seek address (BBCCHHR) Reserved
56-63	38-3F	ACCF	Count field
64-67	40-43	ACMC	Maximum account file capacity
68-71	44-47	ACEC	20% limit residual capacity

Figure 5.24. Account Control Block (Part 1 of 2)

Bytes		Label of field	Description/function of field
Dec	Hex		
72-75	48-4B	ACAC	Current residual capacity
76-79	4C-4F	ACMT	Maximum track capacity
80-83	50-53	ACLC	Residual capacity on current track
84-87	54-57	AC#T	Number of tracks per cylinder
88-89	58-59	ACSE	Sector values
90-91	5A-5B	ACTL	Tolerance
92-93	5C-5D	ACOH	Overhead
94	5E	ACPB	PUB device type code
95	5F	ACDT	DTFPH device type code
		Channel Program (referred to by label ACCH)	
96-103	60-67	ACSK	Seek CCW
104-111	68-6F	ACSS	Set sector or TIC * +8 CCW
112-119	70-77	ACSH	Search ID equal CCW
120-127	78-7F	ACTI	TIC * -8 CCW
128-143	80-8F	ACRW	WCKD CCWs (WCOUNT and WDATA)
144-151	90-97	ACRS	Read sector CCW or not used
152-167	98-A7	ACPM	Channel program modifiers RDATA and RCOUNT CCWs
168-171	A8-AB	ACWA	Virtual address account work space buffer
172-183	AC-B7	None	Not used

Figure 5.24. Account Control Block (Part 2 of 2)

How to Locate

Refer to Figure 6.1 in Section 6.

ACCOUNT WORK SPACE (AWS)

DSECTname: LADS

The account work space contains counters used for building account records by the logical writer IPW$$LW.

Bytes		Label of field	Description/function of field
Dec	Hex		
00-03	00-03	LAER	Extra lines/cards
04-07	04-07	LACR	Current line/card
08-11	08-0B	LARC	Restart current page/card
12-13	0C-0D	LAEP	Extra pages
14-15	0E-0F	LATP	Total pages from data file
16-17	10-11	LACP	Current page
18	12	LASR	START after PSTOP cuu, restart indicator
19	13	LAWS	Work field for copy count
20-23	14-17	LAST	Task start time (OHHMMSSF)

Figure 5.25. Account Work Space

READER ACCOUNT RECORD

A reader account record is created for each read queue entry that is entered into the POWER/VS system. Whether or not the queue entry has actually been placed in the queue file is indicated by the POWER/VS cancel code. The record is copied from the first 58 bytes of the corresponding queue record.

The record shown is the logical record. On the account file IJAFILE and after saving from account file onto tape or disk they are contained as SAM physical records (including 8-byte record control field as shown in the DMB auxiliary account record area).

When saved on punched cards they will appear in columns 2-72. Column 1 of every card contains a copy of the record identifier (byte 42) of the record, columns 73-78 contain the sequential account record number, and columns 79-80 contain the sequential card number within that record (=01).

Bytes	Description	Format[1]
00-07	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy)	a
08-11	Start time of read, in packed decimal (0HHMMSSF; F = sign)	p
12-15	Stop time of read (0HHMMSSF; F = sign)	p
16-31	16 bytes of user information	a
32-39	POWER/VS job name	a
40-41	Job number assigned by POWER/VS	b
42	Record identifier (R)	a
43	POWER/VS cancel code (see DMB)	b
44	Reserved	b
45-47	Reader device address, or SNA	a
48	FROM remote ID	b
49	TO remote ID	b
50	Input class	a
51	Input priority number	a
52-55	Number of records read (including records added or deleted by an RDR exit routine)	b
56-57	Number of tracks for input storage	b

[1] a stands for alphameric
b stands for binary
p stands for packed decimal.

Note: The TO remote-id is a dummy entry in the Reader Account Record and in the Execution Account Record. It is copied from the FROM remote-id.

Figure 5.26. Reader Account Record

LIST ACCOUNT RECORD

A list account record is created for each list queue entry that is processed by a list task. It is a copy of the first 72 bytes of the corresponding queue record.

The record shown is the logical record. On the account file IJAFILE and after saving from account file onto tape or disk they are contained as SAM physical records (including 8-byte record control field as shown in the DMB auxiliary account record area).

When saved on punched cards they will appear in columns 2-72. Column 1 of every card contains a copy of the record identifier (byte 42) of the record, columns 73-78 contain the sequential account record number, and columns 79-80 contain the sequential card number within that record.

Bytes	Description	Format[3]
00-07	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy)	a
08-11	Start time of list, in packed decimal (0HHMMSSF; F = sign)	p
12-15	Stop time of list (0HHMMSSF; F = sign)	p
16-31	16 bytes of user information from * $$ JOB card	a
32-39	POWER/VS job name	a
40-41	Job number assigned by POWER/VS	b
42	Record identifier (L)	a
43	POWER/VS cancel code (see DMB)	b
44	Reserved	b
45-47	Printer device address, or SNA	a
48	FROM remote ID	b
49	TO remote ID	b
50	Printed output class	a
51	Printed output priority number	a
52-55	Number of lines printed	b
56-57	Number of tracks for output storage [1]	b
58	Job suffix number assigned by POWER/VS [4]	b
59	Number of printed copies [2]	b
60-63	Print forms identification	a
64-67	Number of extra records printed due to PRESTART or PSETUP or separator lines	b
68-69	Number of pages printed (skips to channel 1)	b
70-71	Number of extra pages printed due to PRESTART or PSETUP or separator cards	b

[1] Only for spooling to disk. When tape spooling, field is zero.

[2] If more than one copy is provided, the statistics are totals for all copies. When a 3800 Printer has been used, indicates the total transmission count.

[3] a stands for alphameric
b stands for binary
p stands for packed decimal.

[4] Will be X'FF' for last or only segment of job.

Figure 5.27. List Account Record

PUNCH ACCOUNT RECORD

A punch account record is created for each punch queue entry that is processed by a punch task. It is a copy of the first 64 bytes of the corresponding queue record.

The record shown is the logical record. On the account file IJAFILE and after saving from account file onto tape or disk they are contained as SAM physical records (including 8-byte record control field as shown in the DMB auxiliary account record area).

When saved on punched cards they will appear in columns 2-72. Column 1 of every card contains a copy of the record identifier (byte 42) of the record, columns 73-78 contain the sequential account record number, and columns 79-80 contain the sequential card number within that record.

Bytes	Description	Format[3]
00-07	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy)	a
08-11	Start time of punch, in packed decimal (0HHMMSSF; F = sign)	p
12-15	Stop time of punch, (0HHMMSSF; F = sign)	p
16-31	16 bytes of user information from * $$ JOB card	a
32-39	POWER/VS job name from * $$ JOB card	a
40-41	Job number assigned by POWER/VS	b
42	Record identifier (P)	a
43	POWER/VS cancel code (see DMB)	b
44	Reserved	b
45-47	Punch device address, or SNA	a
48	FROM remote ID	b
49	TO remote ID	b
50	Punched output class	a
51	Punched output priority number	a
52-55	Number of records punched	b
56-57	Number of tracks for output storage [1]	b
58	Job suffix number assigned by POWER/VS [4]	b
59	Number of punched copies [2]	b
60-63	Punch forms identification	a
64-67	Number of additional cards punched due to restart or separator cards	a

[1] Only for spooling to disk. When tape spooling, field is zero.
[2] If more than one copy is provided, the statistics are totals for all copies.
[3] a stands for alphameric
b stands for binary
p stands for packed decimal.
[4] Will be X'FF' for last or only segment of job.

Figure 5.28. Punch Account Record

EXECUTION ACCOUNT RECORD

DSECT name: AEDS

An execution account record is created for each queue set (user job step) that has been processed by POWER/VS.

The record shown is the logical record. On the account file IJAFILE and after saving from account file onto tape or disk they are contained as SAM physical records (including 8-byte record control field as shown in the DMB auxiliary account record area).

When saved on punched cards they will appear in columns 2-72. Column 1 of every card contains a copy of the record identifier (byte 42) of the record, columns 73-78 contain the sequential account record number, and columns 79-80 contain the sequential card number within that record.

Bytes	Label	Description	Format[2]
00-07	AEDY	Date of execution in format specified at SYSGEN (mm/dd/yy or dd/mm/yy)	a
08-11	AEST	Start time of job (0HHMMSSF; F = sign)	p
12-15	AEET	Stop time of job (0HHMMSSF; F = sign)	p
16-31	AEUI	16 bytes of user information	a
32-39	AENM	Current POWER/VS job name	a
40-41	AENO	Job number assigned by POWER/VS	b
42	AERI	Record identifier (E)	a
43	AECN	POWER/VS cancel code (see DMB)	b
44	AERJ	Reserved	b
45-47	AECU	Reserved	b
48	AEFJ	FROM remote ID	b
49	AETJ	TO remote ID	b
50	AECL	Class	a
51	AEPY	Priority	b
52-55	-	Number of lines spooled	b
56-59	-	Number of cards spooled	b
60-61	-	Number of pages spooled	b
62-63	-	Length of SIO table	b
64-65	-	Length of total account record	b
66-71	-	Reserved	
72-79	-	DOS/VS job name from // JOB card	a
80-95	-	16 bytes of user information from // JOB card	a
96-97	-	Partition-id in EBCDIC format	a
98	-	DOS/VS cancel code	b
99	-	Type of record; S = job step, L = last step	a
100-103	-	Reserved	
104-111	-	Phase name, taken from execute card	a
112-115	-	End address of active program phase, from COMREG	b
116-119	-	CPU time elapsed in a job step; counted in 300ths of a second	b
120-123	-	Overhead time; elapsed time not charged to any partition, in 300ths of a second	b
124-127	-	All-bound time; system wait state time divided between running partitions, in 300ths of a second	b
128-	-	SIO tables: 6 bytes for each device specified by SYSGEN options, as follows: two bytes for device address (0cuu), four bytes for count of SIOs in current job step [1]	b
-		Overflow byte: normally X'20', but X'30' if more devices are used within a partition than specified by SYSGEN options	b
[3]		User account information (provided via user PUTACCT macro)	

[1] POWER/VS will update the SIO tables in the execution account record with the number of I/Os it has intercepted for spooling purposes.

[2] a stands for alphameric
b stands for binary
p stands for packed decimal

[3] Maximum length of execution account record is 2008 bytes.

Figure 5.29. Execution Account Record

RJE ACCOUNT RECORDS

A line account record is created for each RJE,BSC user session when signoff or line stop is processed. It is a copy of the first 56 bytes of the Line Control Block.

The record shown is the logical record. On the account file IJAFILE and after saving from account file onto tape or disk they are contained as SAM physical records (including 8-byte record control field as shown in the DMB auxiliary account record area).

When saved on punched cards they will appear in columns 2-57. Column 1 of every card contains a copy of the record identifier (byte 50) of the record, columns 73-78 contain the sequential account record number, and columns 79-80 contain the sequential card number within that record.

Bytes hex	Description	Format[1]
00-07	Date in format specified at system generation mm/dd/yy or dd/mm/yy	a
08-0B	Signon time in packed decimal (0MMMMSSF; F=sign)	p
0C-0F	Signoff time (0MMMMSSF; F=sign)	p
10-1F	16 bytes user information	a
20-27	Line password	a
28-29	Reserved	b
2A	Record identifier (T)	a
2B	Signoff code X'01' - Normal signoff X'02' - Signoff forced due to central stop X'04' - Signoff forced due to excessive idle time X'08' - Signoff forced due to unrecoverable I/O error	b
2C	Terminal error count	
2D-2F	Line address	
30	Remote identifier	
31		
32-33	Transmission count per session	
34-35	Timeout count per session	
36-37	Error count per session	

[1] a stands for alphameric
b stands for binary
p stands for packed decimal

Figure 5.30. Part 1: RJE,BSC Line Account Record

A session account record is created when an RJE,SNA user session is terminated. It is a copy of bytes 32-79 of the SNA unit control block.

The record shown is the logical record. On the account file IJAFILE and after saving from account file onto tape or disk they are contained as SAM physical records (including 8-byte record control field as shown in the DMB auxiliary account record area).

When saved on punched cards they will appear in columns 2-57. Column 1 of every card contains a copy of the record identifier (byte 42) of the record, columns 73-78 contain the sequential account record number, and columns 78-80 contain the sequential card number within that record.

Bytes dec	Description	Format[1]
00-07	Date in format mm/dd/yy	a
08-11	Signon time (0HHMMSSF; F=sign)	p
12-15	Signoff time (0HHMMSSF; F=sign)	p
16-31	16 bytes of user information	a
32-39	Logical unit name	a
40-41	Reserved	
42	SNA record identifier (S)	a
43	Session termination code X'01' - normal termination (LOGOFF or SIGNOFF) X'02' - abnormal termination	b
44-47	Remote identifier	b

[1] a stands for alphameric
b stands for binary
p stands for packed decimal

Figure 5.30. Part 2: RJE,SNA Session Account Record

RJE CONTROL BLOCKS

The major control blocks used by POWER/VS RJE are the Line Control Block (LCB) and the Buffer Control Area (BCA).

Line Control Block (LCB)

The line control block describes the line and its status. It contains an entire line account record, which is completed and written to the account file at SIGNOFF time. It also contains the terminal characteristics, which are copied from the remote table in virtual storage at SIGNON time.

When the line is started by the central operator, an LCB is built for that line in real storage. It is not released before the line is stopped. One LCB is always corresponding to each active line, independent of the number of reader and writer tasks operating on the line.

Buffer Control Area (BCA)

The buffer control area contains the CCB, a CCW string, and all other information used to perform a line operation, such as mode bytes and sense information. Like the LCB, the BCA is built in real storage when the line is started, and released when the line is stopped. For hardware terminals, one BCA is used for all line operations.

The format of an LCB with 1 BCA as printed in a dump are shown below:

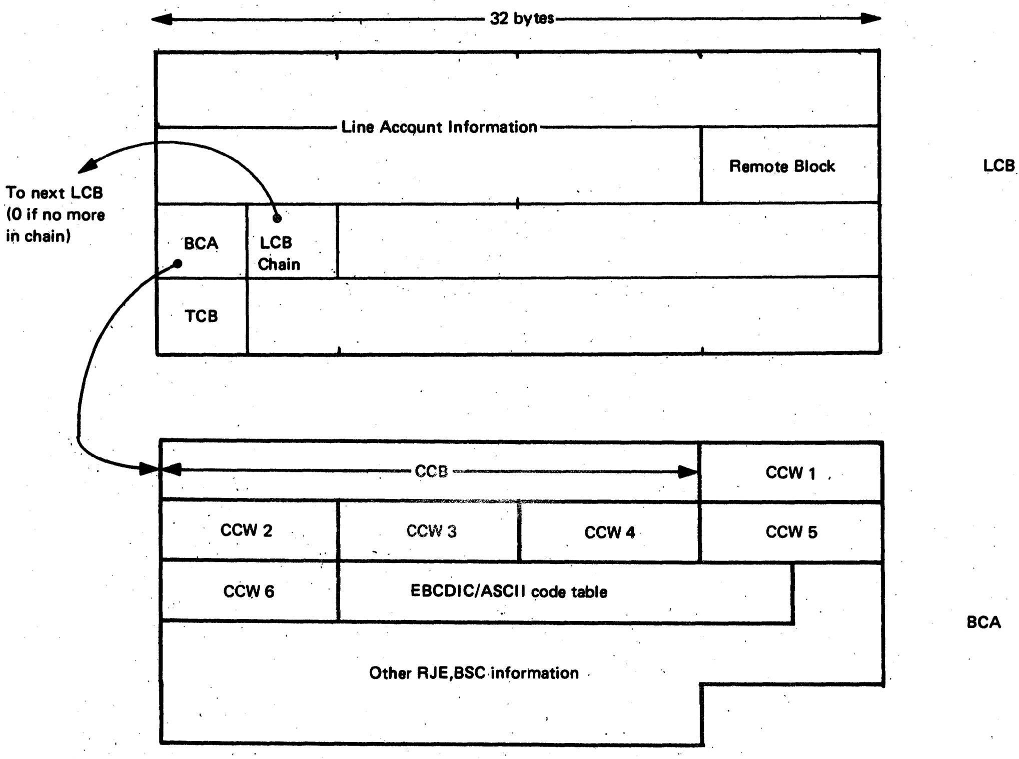


Figure 5.31. RJE Line Control Block and Buffer Control area

LINE CONTROL BLOCK (LCB)

Definition macro: IPW$DLC

Bytes		Label of field[1]	Description/function of field
Dec	Hex		
00-07	00-07	LCBHEAD	LCB header (LCBbbcuu)
08-15	08-0F	LCBDATE	System date in format specified at SYSGEN
16-19	10-13	LCBTIMON	SIGNON time in format 0HHMMSSF; F = sign in packed decimal
20-23	14-17	LCBTIMOF	SIGNOFF time in format 0HHMMSSF; F = sign in packed decimal
24-39	18-27	LCBUSER	User information
40-47	28-2F	LCBPSWD	Line password
48-49	30-31		Reserved
50	32	LCBRECID	Line account record identifier (T)
51	33	LCBSCODE	SIGNOFF code
52	34	LCBERRCT	Terminal error count. This count is only maintained for intervention required and for specific timeouts. When the count reaches 10 a record is written to SYSREC (error recorder file). Then the count is reset to zero. When it reaches 10 again, the same sequence occurs.
53-55	35-37	LCBDEVAD	Line address (in alpha)
56	38	LCBID	Remote identifier (in binary)
57	39		Remote identifier (for compatibility)
58-59	3A-3B	LCBTSMIT	Transmission count per session
60-61	3C-3D	LCBIDLE	Timeout count per session
62-63	3E-3F	LCBERROR	Error count per session
64-67	40-43	LCBBUFAD	Corresponding BCA address
68-71	44-47	LCBCHAIN	LCB chain pointer. A chain of LCBs is maintained and is continuously scanned by the line manager. Up to 25 LCBs can be chained, the last LCB in the chain has 0 in this field.
72-75	48-4B	LCBWORK	Work field

Figure 5.32. Line Control Block (Part 1 of 4)

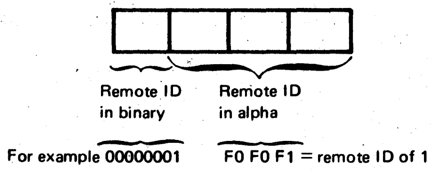
Bytes		Label of field[1]	Description/function of field
Dec	Hex		
76-79	4C-4F	LCBREMID	Remote identifier This 4-byte field consists of:  Remote ID in binary Remote ID in alpha For example 00000001 F0 F0 F1 = remote ID of 1
80-83	50-53	LCBLIST	List output classes. Each byte in this 4-byte field contains a hex displacement in the LST part of the MCTA in the DMB. The displacement contained in one of the bytes therefore corresponds to a class. Up to four classes may be specified, being delimited by X'FF' in a similar manner to that in the TCCT field in the TCB. For example, the command * .. START LST,B,C,D would result in the displacements 18, IC, 20 being placed in this field. (Byte 3 would be X'FF'). If no START LST command is entered the first byte of this field is FF.
84-87	54-57	LCBPUNCH	Punch output classes. Each byte in this 4-byte field is used in the same way as in field LCBLIST, displacements being in the PUN part of the MCTA.
88-89	58-59	LCBTMCNT	Timeout counter. This field counts the number of timeouts (1 every 3 seconds) as long as the terminal is idle (no data transfer). When information is transmitted on the line it is set to zero. The count is compared with the timeout limit specified in the PLINE macro.
90-91	5A-5B	LCBTMOUT	Timeout limit. The value in this field is specified by the TIMEOUT parameter in the PLINE macro. If the user specifies 1 (one minute) the value set in this field is binary 20. The maximum timeout that may be specified is 255 minutes. If the idle time on the line as counted in the field LCBTMCNT exceeds the value of field LCBTMOUT the terminal is signed off by POWER/VS. If TIMEOUT=NO is specified, this field is filled with binary zeros.
92	5C	LCBFEAT	Line features B'1xxx xxxx' - ASCII code line B'x1xx xxxx' - Transparent feature B'xxxx xxx1' - Switched line B'xxYYYYYx' - ooYYYYYo is mode byte for 2701
93	5D	LCBFLAGS	LCB flags X'80' Line stop X'40' Line start/restart X'20' An ETX has been received X'10' No messages wanted [4] X'08' Remote is signed on X'02' SIGNOFF card has been read X'01' Signoff (processing finished)

Figure 5.32. Line Control Block (Part 2 of 4)

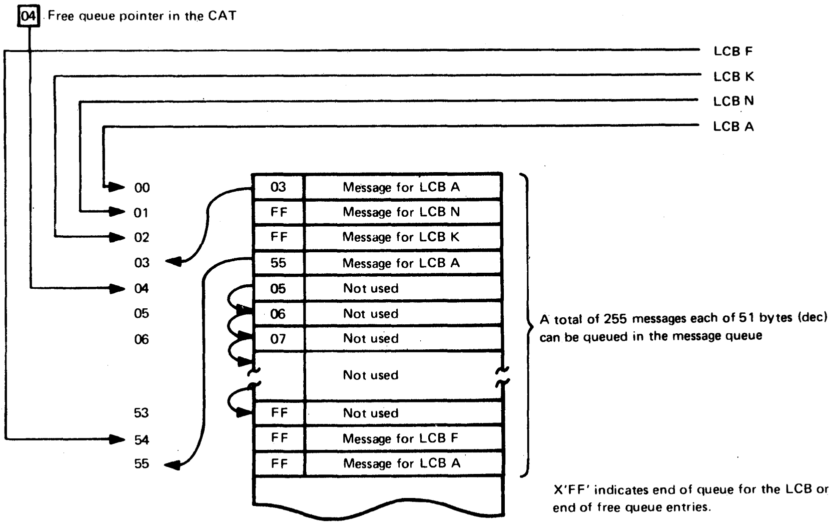
Bytes		Label of field[1]	Description/function of field
Dec	Hex		
94	5E	LCBOUT	Output switches X'80' = list output is ready and dispatchable for any of the classes started. X'08' = punch output is ready and dispatchable for any of the classes started. On START LST, field LCBOUT is initiated with X'80'. On START PUN, field LCBOUT is initiated with X'08'.
95	5F	LCBMSG	Message subchain index. This byte contains the entry number in the message queue that contains the first message queued for this LCB. The illustration below shows three messages queued for a line owning what we shall call LCB A. So, field LCBMSG in LCB A will contain 00, field LCBMSG in LCB N will contain 01, field LCBMSG in LCB K will contain 02, field LCBMSG in LCB F will contain 54.  04 Free queue pointer in the CAT LCB F LCB K LCB N LCB A 00 01 02 03 04 05 06 53 54 55 03 Message for LCB A FF Message for LCB N FF Message for LCB K 55 Message for LCB A 05 Not used 06 Not used 07 Not used Not used FF Not used FF Message for LCB F FF Message for LCB A A total of 255 messages each of 51 bytes (dec) can be queued in the message queue X'FF' indicates end of queue for the LCB or end of free queue entries.
96-99	60-63	LCBTCBAD	Reader TCB address. This field contains the reader TCB address as long as the reader is active. It is set to binary zero when an RDR TCB is detached. If an EOF is encountered in the middle of a job, TCB space is not released, and the field is not set to zero. After the user responds by placing more cards in the reader the reader TCB will be reactivated. To summarize: If the field is zero, a new TCB is created. If the field is nonzero, the original TCB is reactivated.

Figure 5.32. Line Control Block (Part 3 of 4)

Bytes		Label of field[1]	Description/function of field
Dec	Hex		
100-103	64-67	LCBLFORM	List forms
104-107	68-6B	LCBPFORM	Punch forms
108-109	6C-6D	LCBPUB	Corresponding PUB address
110-111	6E-6F	LCBMSCTR	Message counter. This field contains the number of messages for the user of this LCB in the remote message queue. Since this remote message queue contains 255 entries an overflow condition may occur. Should an overflow occur, the LCB with the highest value in this field is located, all messages for this user are removed from the message queue and replaced by message 1R20I.
112-119	70-77		Not used
Remote Block			
120	78	LCBPROUT	Default punch routing
121	79	LCBLROUT	Default list routing
122-123	7A-7B	LCBBFSIZ	Terminal buffer size Size of this buffer depends on type of terminal.
124	7C	LCBPLINE	Terminal type plus line features X'80' Log every channel end [2] X'40' Transparency [3] X'20' ASCII code X'02' 3780 X'01' 2780 or 3741 X'00' 2770 or 3780 with component select
125	7D	LCBPRMT	Terminal features X'20' Hardware compress feature X'10' Multiple-record feature X'08' Horizontal format control X'02' Variable length records X'01' Blocked records
126	7E	LCBPRLEN	Remote printer width
127	7F		Reserved

Figure 5.32. Line Control Block (Part 4 of 4)

[1] In PL/S listings the first characters of these labels are LC instead of LCB.

[2] The byte is set X'80' if TRACE=YES has been specified in the REMOTE macro. It enables a wraparound I/O trace in the phase IPW$$TM. (See Section 6 for details.)

[3] Transparency enables user to transmit object decks over line. If no transparency is supported codes X'00' through X'40' are converted to binary zeros for output to the terminal.

Transparency on input (read) is determined by a switch on the terminal unit.

For print output to the terminal, non-transparency if forced by POWER/VS.

For punch output, transparency depends on whether it has been specified in the PLINE macro and the PRMT macro. If either one has not been specified for transparency, non-transparency is forced by POWER/VS.

4 X'10' in this byte indicates that the command *b..bSTOP MSG has been issued. This prevents system and broadcast messages from being transmitted and printed on output which must be free from such messages, such as salary slips.

Messages are always queued and issued when an ACK is received from the terminal (PRINTER ready). The STOP MSG command causes all messages to be lost, that is, they are not queued.

Messages are requeued and output as normal by issuing the command *b..bSTART MSG. See field LCBMSG for an explanation of message queues.

BUFFER CONTROL AREA (BCA)

Definition macro: IPW$DBC

Bytes		Label of field	Description/function of field
Dec	Hex		
		RJE CCB (label BCACCB) Initialized with a complete sense CCW to read sense information into the two sense bytes in the BCA. Byte 12 is initialized with bits 1 and 2 on, indicating that channel-end appendages and private unit-check routines are being used.	
00-01	00-01	CCBCNT	Residual count
02	02	CCBCOM1	Communication byte
03	03	CCBCOM2	Communication byte
04-05	04-05	CCBSTA	Status bytes from CSW
06-07	06-07	CCBLUB	Logical unit number
08-11	08-0B	IOBSTART	First CCW address
12	0C	CCBBY3	Communication byte
13-15	0D-0F	CCBAPP	Channel appendage address
16-23	10-17	BCACCW0	RJE sense CCW
		RJE CCW string, dynamically set up by MCCWINIT routine	
24-31	18-1F	IOBCCW1	These six CCW fields constitute various channel programs that depend on the operation required. For example, a READ program consists of • An enable CCW • A write response CCW • A read text CCW A WRITE program has a different CCW string and a PREP program consisting of: • A disable CCW • A set mode CCW • An enable CCW • A write enquiry CCW • A read response CCW These CCW strings are described in Sections 2 and 3.
32-39	20-27	IOBCCW2	
40-47	28-2F	IOBCCW3	
48-55	30-37	IOBCCW4	
56-63	38-3F	IOBCCW5	
64-71	40-47	IOBCCW6	
		EBCDIC/ASCII Code Table This table is moved from virtual storage at OPEN time for RDR, LST, or PUN to reflect one of the following four conditions: EBCDIC code transparency EBCDIC code non-transparency ASCII code transparency ASCII code non-transparency	

Figure 5.33. Buffer Control Area (Part 1 of 3)

Bytes		Label of field	Description/function of field
Dec	Hex		
72-73	48-49	MSOHENQ	Multileaving sign-on sequence
74-75	4A-4B	MSTXSEQ	Start of text sequence
76	4C	METBSEQ	End of text block sequence
77	4D	MBSCETB	End of text block character This 1-byte field is the second byte of the previous 2-byte field.
78-79	4E-4F	METXSEQ	End of text sequence
80	50	MACK0SEQ	Even acknowledgement sequence
81	51	MBSCACK0	Even acknowledgement character This 1-byte field is the second byte of the previous 2-byte field.
82	52	MACK1SEQ	Odd acknowledgement sequence
83	53	MBSCACK1	Odd acknowledgement character This 1-byte field is the second byte of the previous 2-byte field.
84	54	MNAKSEQ	Negative acknowledgement sequence
85	55	MBSCNAK	Negative acknowledgement character This 1-byte field is the second byte of the previous 2-byte field.
86	56	MBSCACKX	Acknowledgement conversion character
87	57	MBSCCWCH	CCW chaining character
88	58	MBSCENQ	Enquiry character
89	59	MBSCEOT	End of transmission character
90	5A	MBSCWACK	Wait before transmit
91-92	5B-5C	MEOTSEQ	DLE-EOT characters
		Other RJE Information	
93-94	5D-5E	LCBWRCNT	Saved write count
95	5F	BCAUCCNT	Maximum retry count
96-99	60-63	IOBRESTR	Restart address of channel program
100	64	TPBLCCC	Last remote output command code
101-103	65-67	TPBLCCAD	Address of the last remote carriage control
104	68	TPBRECNT	Current remote output record count. Incremented at each PUT during a WRITE operation. The count is compared with the value in the field TPBMXREC. If they match, the PDA is written out.
105-107	69-6B	TPBFDATA	Remote data pointer. Address of the record in the PDA currently being processed.
108-111	6C-6F	BUFEWF	Address of last data byte read. Address of the last byte read into the PDA (not necessarily the last byte in the PDA).
112-115	70-73	BUFDCT	Address of the corresponding LCB
116-119	74-77	LASTCCW	Address of the last CCW executed plus 8 (stored by the RJE channel appendage routine on every I/O interrupt).
120-123	78-7B	BCADISP	Displacement between the real and virtual address of the BCA (used to construct real addresses for the channel program).
124-127	7C-7F	BUFCHAIN	Address of next CCB completed. BCA chain pointer, set up by channel appendage routine, and processed by line manager.
128-131	80-83	BCATCBAD	Address of a list TCB or punch TCB as long as an RJE LST or PUN task is active. Otherwise, the field is 0. (Has same function as field LCBTCBAD in the LCB.)
132-135	84-87	RLBUFAD	Real address of PDA. Address of current TP buffer. Updated whenever a new buffer is obtained.
136-139	88-8B	TPBUFAD	Virtual address of PDA. Address of current TP buffer. Updated whenever a new buffer is obtained.
140-143	8C-8F	BCAFRMCB	Address of last TCB, as long as a form change is needed. Otherwise, the field is 0.

Figure 5.33. Buffer Control Area (Part 2 of 3)

Bytes		Label of field	Description/function of field
Dec	Hex		
144	90	LCBMCB	Remote mode byte (SDA mode byte). Set at line initialization time to X'04' for 2701, and to X'00' for 2703 TP control unit or ICA.
145	91	LCBACK	Remote next acknowledgement. For write response CCW.
146-147	92-93	LCBRCB	Remote response control block. Used for the write response to the terminal, and to read the response from the terminal.
148	94	IOBSENS0	First sense byte (see TP manual)
149	95	IOBSENS1	Second sense byte (always 0)
150	96	TPBMXREC	Maximum output record count. Contains a value equal to the maximum record count for the output buffer at the terminal. Its value depends on the terminal type.
151	97	BCAFLAGS	BCA flags X'01' end of transmission X'02' end of forms X'04' second entry to put routine

Figure 5.33. Buffer Control Area (Part 3 of 3)

BSC LINE MANAGER TCB FIELDS

These fields replace the General Task Work Area, Linkage Register Save Area, and the File Control Words of the TCB in case it belongs to an RJE,BSC Line Manager task.

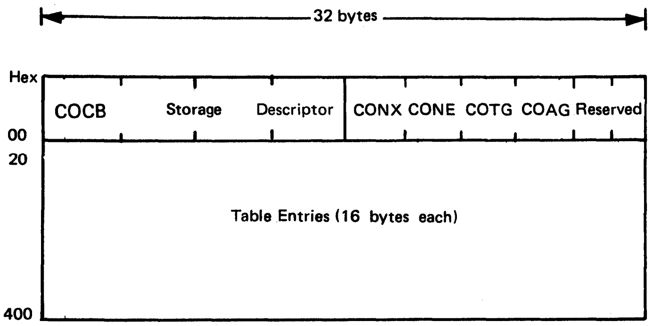
Bytes		Label of field	Description/function of fields
Dec	Hex		
00-27	00-1B		Reserved
28-31	1C-1F	TCSR	SYSREC header
32-87	20-57	TCLM	BSC line manager save area
88-103	58-67		Reserved

Figure 5.34. BSC Line Manager TCB Fields

SNA COMPACTION TABLE CONTROL BLOCK (COCB)

Definition macro: IPW$DCO

The address (COAD) in a COCB entry will be used for retrieving the FMH3, and fetching the compaction table for use in the compaction algorithm.



Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	COSD	Storage descriptor (COCB)
16-19	10-13	CONX	Address next COCB
20-21	14-15	CONE	Number of entries in COCB
22-23	16-17	COIG	Maximum number of GETVIS table entries (1K each)
24-25	18-19	COAG	Actual number of GETVIS
26-31	1A-1F		Reserved
32-47	20-2F	First Compaction Table Entry	
32-35	20-23	CONA	Compaction table name
36-39	24-27	COAD	Compaction table address
40	28	COID	Compaction table identifier
41	29		Reserved
42-43	2A-2B	COUS	Compaction table use counter
44-45	2C-2D	COLN	Compaction table length
46-47	2E-2F		Reserved
48-1023	30-3FF	Remaining Table Entries	

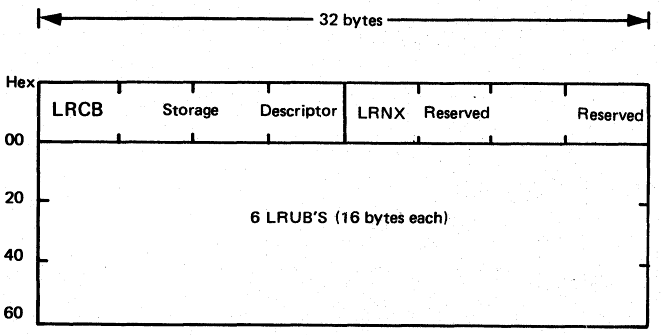
Figure 5.34A. SNA Compaction Table Control Block

SNA LOGON REQUEST CONTROL BLOCK (LRCB)

Definition macro: IPW$DLR

A logon request control block contains information for 6 logon requests to the POWER/VS application. All logon request control blocks are chained. The pointer to the first LRCB is contained in the SNA control block (SNCB).

Information about logon requests are stored in the LRCB by the logon exit of the SNA manager. The logon processor processes the logon requests to build SUCB/LUCBs.



Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	LRSD	Storage descriptor (LRCB)
16-19	10-13	LRNX	Pointer to next LRCB
20-23	14-17		Reserved
24	18	LRLC	Length of one LRCB
25	19	LRLB	Length of one LRUB
26	1A	LRAL	No. of total LRUBs in LRCB
27	1B	LRUS	No. of active LRUBs in LRCB
28-31	1C-1F		Reserved
32-47	20-2F	First LRUB entry	
32-35	20-23	LRAC	ACB address
36-43	24-2B	LRLU	LU-name
44	2C	LRST	Status (X'01' indicates active entry)
45-47	2D-2F	LRLM	Length of logon message
48-127	30-7F	Remaining LRUB entries	

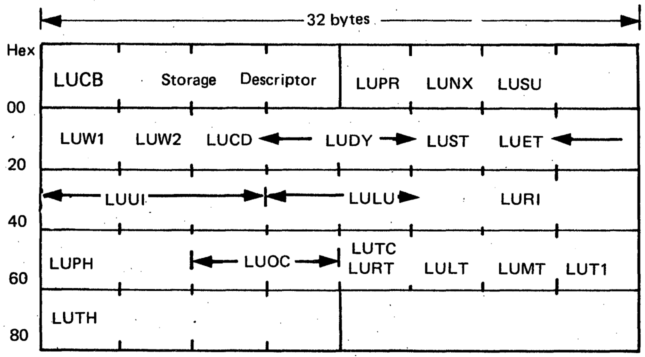
Figure 5.34B. SNA Logon Request Control Block

SNA LOGICAL UNIT CONTROL BLOCK (LUCB)

Definition macro: IPW$DLU

A logical unit control block (LUCB) is created for each logical unit logon to the POWER/VS application.

All logical unit control blocks within one work station are chained together. The pointer to the first LUCB within one work station is contained in the SUCB, which describes this work station.



Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	LUSD	Storage descriptor
16-19	10-13	LUPR	Previous address LUCB (pointer)
20-23	14-17	LUNX	Address of next LUCB (pointer)
24-27	18-1B	LUSU	Address of SUCB
28	1C	LUSL	Select indicator
			C'S'-select
29	1D	LUTT	Termination type (C'S' - immediate)
30	1E	LUTX	Termination type, set by exits
31	1F	LUFS	Free session indicator
			X'80' - session in use
32-35	20-23	LUW1	Reader 1 work space address
36-39	24-27	LUW2	Reader 2 work space address
40-43	28-2B	LUCD	VTAM CID
		Session Accounting Information	
44-51	2C-33	LUDY	Date = c'MM/DD/YY'
52-55	34-37	LUST	Signon time = X'0HHMMSSF'
56-59	38-3B	LUET	Signoff time = X'0HHMMSSF'
60-75	3C-4B	LUUI	User information
76-83	4C-53	LULU	LU name
84-85	54-55		Reserved
86	56	LUAI	c'S' - identifier for account record.
87	57	LUCN	Session termination code
			X'01' - Normal (SIGNOFF or LOGOFF)
			X'02' - Abnormal
88-91	58-5B	LURB	Remote identifier
88	58	LURB	Binary format
89-91	59-5B	LURC	Character format

Figure 5.34C. SNA Logical Unit Control Block (Part 1 of 3)

Bytes		Label of field	Description/function of field
Dec	Hex		
		Restart Information	
92-95	5C-5F	LURS	Restart
92	5C	LURX	Restart function index
93-95	5D-5F	LURD	Restart page count
		List and Punch Characteristics	
96-99	60-63	LUPH	Pointer to device in SUCB
100	64	LULO	List output support
			X'00' - List output support
			X'80' - ASCII
			X'40' - Compression
			X'20' - Transparency
			X'10' - Spanning
			X'08' - Inter-record separator
			X'01' - Compaction
101	65	LUPO	Punch output support
			X'00' - Punch output support
			X'80' - ASCII
			X'40' - Compression
			X'20' - Transparency
			X'10' - Spanning
			X'08' - Inter-record separator
			X'01' - Compaction
102	66	LUPD	PDIR information byte
			X'80' - PDIR outbound allowed
103	67	LUAD	Card/document flow
			X'00' - Card/document flow
			X'80' - Card inbound allowed
			X'40' - Card outbound allowed
			X'08' - Document inbound allowed
			X'04' - Document outbound allowed
104-111	68-6F	LUOC	Current compaction table in use by outbound processor
104-107	68-6B	LU01	Compaction table name
108-111	6C-6F	LU02	Pointer to CDCB entry
		Process Control Section	
112-115	70-73	LUTC	Start address of TCBs for LU
112-115	70-73	LURT	RDR, LGN, LGF, TCB address
116-119	74-77	LULT	LST, PUN TCB address
120-123	78-7B	LUMT	Message TCB address
124-127	7C-7F	LUTI	RDR2 TCB address
128-131	80-83	LUTH	LGH TCB address
132	84	LUA1	Action byte
			X'80' - Request logon
			X'40' - Request start reader
			X'20' - Request interrupt LST/PUN on signal
			X'10' - Request interrupt LST/PUN for outbound message
			X'08' - Request interrupt inbound for inbound
			X'04' - Request stop session
			X'02' - Request GO/SETUP command
			X'01' - Request for restart command
133	85	LUP1	Process byte
			X'80' - Logoff in process

Figure 5.34C. SNA Logical Unit Control Block (Part 2 of 3)

Bytes		Label of field	Description/function of field
Dec	Hex		
134	86	LUS1	Status byte one X'80' - BB reject indicator X'40' - Logon completed X'01' - 1 - BB reject by IPW$$OB C - BB reject by IPW$$MP
135	87	LUS2	Status byte two X'80' - Change direction X'40' - LST/PUN suspended for msg X'20' - LST/PUN suspended for inbound X'10' - Inbound suspended for inbound
136	88	LUBR	Bracket state
137	89		Unused
138-139	8A-8B	LUBS	Buffer size
140-141	8C-8D	LUBSL	Buffer size logon process
142-159	8E-9F		Unused

Figure 5.34C. SNA Logical Unit Control Block (Part 3 of 3)

SNA REMOTE CONTROL BLOCK (RMCB)

Definition macro: IPW$DRM

The SNA remote control block consists of:

- General information that is not required in real storage for RJE,SNA processing
- A general work space to be used by any SNA routine that cannot obtain virtual storage via the DOS/VS GETVIS macro
- Translate tables to convert EBCDIC characters to ASCII and vice versa
- Remote entries for each remote ID specified in the PRMT macro at DOS/VS POWER/VS generation.

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	RMSD	Storage descriptor (RMCB)
16-19	10-13		Reserved
20-22	14-16		Reserved
23	17	RMAL	Length of ACB password
24-31	18-1F	RMAP	ACB password
32-1055	20-41F	RMGP	General purpose work space This area is serially accessible by SNA tasks that cannot obtain virtual storage via the DOS/VS GETVIS macro. Access is regulated by a lockword (SNRL) located in the SNA control block (SNCB).
1056-1311	420-51F	RMEA	Translate table to convert EBCDIC characters to ASCII
1312-1439	520-59F	RMAE	Translate table to convert ASCII characters to EBCDIC
			SNA Manager save area
1440-1567	5A0-61F	RMSNSS	Subtask save area
1440-1447	5A0-5A7	RMSNSN	C 'IPW$$SN' - Subtask name
1448-1567	5A8-61F	RMSNSR	Register save area
1568-1639	620-667	RMSNRA	VTAM RECEIVE Any save area
1640-1711	668-6AF	RMSNST	VTAM SETLOGON save area
1712	6B0	RMSR	No. of SNA remote entries
1713	6B1	RMFR	First SNA remote ID
1714	6B2	RMHR	Last SNA remote ID
1715	6B3	RMNC	Total no. of 1K blocks in GETVIS pool for compaction tables including COCB.
1716-	6B4-	RMRM	Remote entries (The number of remote entries, which are 16 bytes long, depends on the number of SNA remote units specified in the PRMT macro at POWER/VS generation.)

Figure 5.35. SNA Remote Control Block (Part 1 of 2)

Bytes		Label of field	Description/function of field
Dec	Hex		
			The following is a layout of a remote entry.
		RMPR	Punch routing remote ID (1 byte)
		RMLR	List routing remote ID (1 byte)
		RMRI	Remote reference information
		RMBS	Buffer size (1 byte)
		RMTT	Terminal type (1 byte)
		RMTF	Terminal features (1 byte)
		RMCS	Console specified (1 bit)
			Reserved (7 bits)
			Reserved (1 byte)
		RMPL	Length of password (1 byte)
		RMPW	Password (8 bytes)
		RMCN	Compaction table name (4 bytes)
		RMNL	Number of LU names (1 byte)
		RMLA	Address of first LU name in list (3 bytes)
		RMSL	Session limit (2 bytes)
			Reserved (6 bytes)

Figure 5.35. SNA Remote Control Block (Part 2 of 2)

How to Locate

Refer to Figure 6.1 in Section 6.

SNA CONTROL BLOCK (SNCB)

Definition macro: IPW$DSN

The SNA control block contains general information that is required in real storage for RJE,SNA processing.

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	SNSD	Storage descriptor (SNCB)
16	10	SNTT	SNA termination type
17	11	SNTX	Termination type set by SNA exit routines
18	12	SNFL	Flag byte:
			X'80' - SNA stop requested (SNSS)
			X'40' - Kill SNA requested (SNKS)
			X'20' - Subtask detach requested (SNST)
			X'10' - Subtask quiesce requested (SNRQ)
19	13	SNSU	Maximum number of logical units
20-23	14-17	SNFS	Address of first active SNA unit control block (SUCB)
24-27	18-1B	SNSP	Address of SNA unit control block space
28-31	1C-1F	SNLW	SNA control block lockword
32-35	20-23	SNRM	Address of SNA remote control block (RMCB)
36-39	24-27	SNRL	Lockword for general purpose work space (RMGP) in SNA remote control block
40-43	28-2B	SNTC	Address of SNA manager TCB
	2C-2F	SNSB	Subtask ECB
	30-33	SNEB	SNA manager work ECB
	30-31		Unused
	32		Post byte
	33		Unused
	34-3B		Reserved
	3C-7B	SNAC	VTAM ACB + 4 words for expansion
	7C-7F	SNSL	Pointer to logon space pool for SUCB/LUCB, WACB and LRCBs
	80-83	SNCP	Compaction table pool address
	84-87	SNCR	Address first logon request control block (LRCB)
	88-8B	SNWS	Address logon SUCB
	8C-8F	SNCA	Address of compaction table
	90-93	SNEC	LRCB chain - lockword 1
	94-97	SNFC	LRCB chain - lockword 2
	98-9B	SNCL	Compaction table lockword
	9C		IPW$$LH process byte
			X'80' - Request for IPW$$LH
			X'40' - IPW$$LH is active
	9D-B7		Unused

Figure 5.36. SNA Control Block

How to Locate

Refer to Figure 6.1 in Section 6.

SNA UNIT CONTROL BLOCK (SUCB)

Definition macro: IPW$DSU

An SNA unit control block is created for each logical unit logged on to the POWER/VS application. All SNA unit control blocks are chained together.

Bytes		Label of field	Description/function of field
Dec	Hex		
0-15	00-0F	SUSD	Storage descriptor (SUCB)
16-19	10-13	SUNX	Address of next SUCB
			General Accounting Information
20-31	14-1F	SUAR	General Information
20-27	14-1B	SUDY	Date = c'MM/DD/YY'
28-31	10-1F	SURI	Remote Identifier
28	1C	SURB	- Binary format
29-31	1D-1F	SURC	- Character format
			List, Punch and Reader Device Characteristics
32	20	SULP	List routing remid
33	21	SUPP	Punch routing remid
			Device status values for the following devices:
			X'80' - Device started
			X'40' - Device available
			X'20' - Output available
36-39	24-27	SUL1P	Printer 1 - C'LST1'
40	28	SUL1S	Device status
41-43	29-2B	SUL1L	Pointer to LUCB
44-47	2C-2F	SUL1F	Forms ID
48-51	30-33	SUL1C	List output classes
52-55	34-37	SUL2P	Printer 2 - C'LST2'
56	38	SUL2S	Device status
57-59	39-3B	SUL2L	Pointer to LUCB
60-63	30-3F	SUL2F	Forms ID
64-67	40-43	SUL2C	List output classes
68-71	44-47	SUL3P	Printer 3 - C'LST3'
72	48	SUL3S	Device status
73-75	49-4B	SUL3L	Pointer to LUCB
76-79	40-4F	SUL3F	Forms ID
80-83	50-53	SUL3C	List output classes
84-87	54-57	SUP1P	Punch - C'PUN1'
88	58	SUP1S	Device status
89-91	59-5B	SUP1L	Pointer to LUCB
92-95	5C-5F	SUP1F	Forms ID
96-99	60-63	SUP1C	Punch output classes

Figure 5.37. SNA Unit Control Block (Part 1 of 2)

Bytes		Label of field	Description/function of field
Dec	Hex		
100-103	64-67	SUR1P	Reader - C'RDR1'
104	68	SUR1S	Device status
105-107	69-6B	SUR1L	Pointer to LUCB
108-111	6C-6F	SUR1F	Forms ID
112-115	70-73	SUR1C	Reader classes
116-119	74-77	SUC1P	Console - C'CON1'
120	78	SUC1S	Device status
121-123	79-7B	SUC1L	Pointer to LUCB
124-127	7C-7F	SUC1C	Forms ID
128-131	80-83	SUC1C	Console classes
132	84	SUHD	Device List delimiter
133-134	85-86		Reserved
135	87	SUDLS	Device select indicator
136-143	88-8F	SUOC	Default compaction table for outbound
136-139	88-8B	SUO1	Name of default table
140-143	8C-8F	SUO2	Address of default table virtual
144	90	SUAD	Card/document flow
			X'80' - Card inbound allowed
			X'40' - Card outbound allowed
			X'08' - Document inbound allowed
			X'04' - Document outbound allowed
			Message Control Section
145	91	SUMR	Message request status
			X'80' - Message processor for work station is active
			X'40' - Request to interrupt IPW$$OB for outbound message was issued
146-148	92-94	SUMRL	Pointer to the LUCB with the suspending IPW$$OB
149	95		Unused
150-151	96-97	SUMN	No. of messages
152	98	SUMC	Subchain index
153	99	SUMD	Temporary delete chain index
154	9A	SUTY	Terminal type
155	9B	SUTF	Terminal features
			X'80' - Console specified
			Miscellaneous
156-159	9C-9F	SUWLW	Address of work station lockword
160-163	A0-A3	SUW1	Inbound work space address
164	A4		Reserved
165-167	A5-A7	SUWSL	Pointer to LUCB. If set then workspace is in use by the LUCB being pointed to.
168-171	A8-AB	SUPL	Pointer to first LUCB
172-173	AC-AD	SUN1	No. of attached LUCBs
174-175	AE-AF	SUN2	No. of active LUCBs
176-191	B0-BF		Unused

Figure 5.37. SNA Unit Control Block (Part 2 of 2)

How to Locate

Refer to Figure 6.1 in Section 6.

SNA WORK AREA (WACB)

Definition macro: IPW$DWA

This work space is reserved for and used by each logical unit processing routine (RDR, LST, PUN, and MSG).

Bytes		Label of field	Description/function of field
Dec	Hex		
		WALN	Fixed part of SNA work space
00-15	00-0F	WASD	Storage descriptor (WACB)
		WABC	Buffer control fields
16-19	10-13	WARC	Residual count in buffer
20-23	14-17	WACR	Current position in buffer
24-27	18-1B	WABI	Address of buffer to SEND/RECEIVE
28-31	1C-1F	WABP	Address of buffer in process (FILL)
32-35	20-23	WARL	Logical record length
36-171	24-AB	WALR	Logical record
172	AC	WASW	Processing switches:
			X'02' - End of file
			X'01' - Logical interface open (WALI)
			As used by SNA inbound processor:
			X'80' - Processing console (WAIC)
			X'40' - Processing reader (WAIR)
			X'20' - Unconditional end bracket (WAUB)
			X'10' - Immediate termination required (WATI)
			X'08' - Resume data stream state requested (WASR). A resume FMH was received.
			As used by SNA outbound processor:
			X'80' - EOF reached (WAOF)
			X'40' - EOJ or chain reached (WAOJ)
			X'20' - End of logical record reached (WAOL)
			X'10' - End of RU reached (WAOR)
			X'08' - SETUP/GO in progress
			X'04' - FMH3 send indicator
			As used by SNA message processor:
			X'20' - Component not available (WAMC)
			X'10' - End of RU reached (WAMR)

Figure 5.38. SNA Work Area (Part 1 of 3)

Bytes		Label of field	Description/function of field
Dec	Hex		
173	AD	WAST	Status byte
174	AE	WASS	Data stream state byte:
		WASF	Data stream state flags: X'80' - Between data stream (WASB) X'40' - In data stream (WASI) X'20' - End of data stream pending (WASE) X'10' - Data stream abort (WASA)
175	AF	WACS	Chain state byte:
		WACF	Chain state flags: X'80' - Between chain (WACB) X'40' - In chain (WACI) X'20' - End of chain pending (WACL) X'10' - End of file (WACE)
176	B0	WAPR	Processing options: X'80' - ASCII (WAAS) X'40' - Compression (WACM) X'20' - Transparency (WATR) X'10' - Spanning (WASP) X'08' - Inter-record separator (WARS) X'01' - Compaction support
177	B1	WACI	Compaction indicator X'40' - Compaction table found X'80' - Indicates use count increased
178-179	B2-B3		Reserved
180-183	B4-B7	WAPH	Save area for LUPH, used by interrupting processors
184-187	B8-BB	WASN	Error sense bytes
188	BC	WAMN	Error message number
189-191	BD-BF		Reserved
192-195	C0-C3	WAER	Error routine address
196-307	C4-133	WARP	RPL (plus 12 bytes for expansion)
308-379	134-17B	WASV	Register save area
380-491	17C-1EB	WADA	PL/S dynamic area
492-503	1EC-1F7		Reserved

Figure 5.38. SNA Work Area (Part 2 of 3)

Bytes		Label of field	Description/function of field
Dec	Hex		
504-511	1F8-1FF	WAFM	Function management header (plus 2 bytes for expansion)
504	1F8	FMHLN	FMH length byte
505	1F9	FMHTYP	FMH type byte X'80' - Concatenation B'00111111' - FMH type 1
506	1FA	FMHSEL	FMH select byte B'10000000' - Demand select B'01110000' - Device select B'00000000' - Console B'00100000' - Card reader B'00110000' - Printer B'00001111' - Logical address
507	1FB	FMHFLAG	FMH flag byte - FMH1 ADS sent on inbound data error X'80' - ADS send by receiver
508	1FC	FMHPROP	FMH properties byte B'11100000' - Data stream state B'00000000' - Data stream resume B'00100000' - Data stream end B'01000000' - Data stream begin B'01100000' - Data stream begin and end B'10000000' - Data stream interrupt B'10100000' - Data stream abort B'00011111' - Data stream characteristics B'00010000' - Basic exchange B'00000100' - Compression B'00000010' - Compaction
509	1FD	FMHERCL	FMH basic exchange length byte
512-1023	200-3FF	WABF	Two SNA buffers

Figure 5.38. SNA Work Area (Part 3 of 3)

How to Locate

Refer to Figure 6.1 in Section 6.

SNA MESSAGE CONTROL BLOCK (MSCB)

Definition macro: IPW$DMS

The SNA message control block controls all access to the remote message queue. The block is created by the DOS/VS POWER/VS initialization routine (IPW$$IR) if RJE processing (BSC and/or SNA) has been specified in the POWER/VS generation macros.

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F	MSSD	Storage descriptor (MSCB)
16-23	10-17	MSWW	Work area
24	18	MSFC	Free chain index
25	19	MSFI	Function indicator
26-27	1A-1B		Reserved
28-31	1C-1F	MSLW	Lockword
		MSSV	Register save area
32-35	20-23	MSRE	Register 14
36-39	24-27	MSRF	Register 15
40-43	28-2B	MSR0	Register 0
44-47	2C-2F	MSR1	Register 1
48-51	30-33	MSR2	Register 2
52-55	34-37	MSR3	Register 3
56-59	38-3B	MSR4	Register 4
60-63	3C-3F	MSR5	Register 5
64-67	40-43	MSR6	Register 6
68-71	44-47	MSR7	Register 7
72-75	48-4B	MSR8	Register 8
76-79	4C-4F	MSR9	Register 9
80-87	50-57		Reserved

Figure 5.39. SNA Message Control Block

How to Locate

Refer to Figure 6.1 in Section 6.

SNA MANAGER TCB FIELDS

These fields replace the General Task Work Area, Linkage Register Save Area, and the File Control Words of the TCB in case it belongs to an SNA manager task.

Bytes		Label of field	Description/function of field
Dec	Hex		
		TCEL	Wait ECB list
00-03	00-03	TCE1	Receive any ECB address
04-07	04-07	TCE2	Work ECB address
08	08	TCED	End of list (X'FF')
09-15	09-0F		Reserved
16-19	10-13	TCSU	Address of SNA unit control block (SUCB)
20-23	14-17	TCWA	Address of SNA work area (WACB)
24-27	18-1B	TC13	Save area for register 13
28-31	1C-1F		Reserved

Figure 5.40. SNA Manager TCB Fields

POWER/VS GENERATION TABLE (GNB)

Definition macro: IPW$DGN

The load routine required to load IPW$$I1 and a generation table with POWER/VS default options are supplied to the user cataloged on CIL together with all POWER/VS phases.

Should the user require other than default options, a new generation table must be assembled, and cataloged to CIL. (Refer to Appendix B.)

Bytes		Label of field	Description/function of fields
Dec	Hex		
00-15	00-0F	GNSD	Storage descriptor (GNB)
16-19	10-13	GNDB	DBLK value
20-21	14-15	GNTR	Track group value
22-23	16-17	GNTL	Table length plus BSC
24	18	GNSL	Sublibrary
25	19	GNJA	Account switch [1]
26	1A	GNPP	Pause punch switch
27	1B	GNLG	Log option [2]
28	1C	GNPY	Default priority
29	1D	GNNL	Number of BSC lines
30	1E	GNNR	Number of BSC remotes
31	1F	GNSP	Spool management specification
		Master List Values (referred to by label GNLV)	
32	20	GNOP	Option byte X'80' Clear printer at EOF (see POWER macro) X'40' Mark form option for separator pages (see POWER macro) X'20' No separator pages/cards between copies (see POWER macro) X'01' - Feed for 3540 (see POWER macro) X'02' - Channel 12 multiple (see POWER macro)
33	21	None	Reserved
35	23	GNJL	JSEP list
36-39	24-27	GNL1	STDLINE first
40-43	28-2B	GNL2	STDLINE second
44-47	2C-2F	GNRL	RBS list
		Master Punch Values (referred to by label GNPV)	
48-50	30-32	None	Reserved
51	33	GNJP	JSEP punch
52-55	34-37	GNC1	STDCARD first
56-59	38-3B	GNC2	STDCARD second
60-63	3C-3F	GNRP	RBS punch
64-71	40-47	GNRE	RDREXIT name
72-79	48-4F	GNIN	IPW$$I1 phase name
		During initialization the IPW$$I1 phase name is overlaid by the following fields:	
72-75	48-4B	GNRM	Address of remote control block (RMCB)
76-79	4C-4F	GNSS	Address of SNA unit control block (SUCB)

Figure 5.41. Generation Table (Part 1 of 2)

Bytes		Label of field	Description/function of fields
Dec	Hex		
80-92	50-5C	GNLT	LTAB
93-95	5D-5F	GNLU	Length of LU table. Accumulated length of LU= in PRMT macro.
		SNA Information	
96-97	60-61	GNTT	Table length plus BSC plus SNA
98	62	None	Reserved
99	63	GNAL	Length of ACB password
100-107	64-6B	GNAP	ACB password
108	6C	GNSU	Maximum number of SNA logical units [3]
109	6D	GNSR	Number of SNA remotes
110	6E	GNFR	First SNA remote ID
111	6F	GNHR	Last SNA remote ID
112	70	GNVA	Length of APPLID for VTAM
113-120	71-78		APPLID for VTAM
121-127	79-7F		Reserved
Variable (depending on number of lines (GNNL))			BSC line table entries of 12 bytes each
Variable (depending on number of remotes (GNNR))			BSC remote block entries of 8 bytes each
Variable (depending on number of SNA remotes (GNSR))			SNA remote block entries of 20 bytes each

[1] This byte contains a single alphabetic character; the character 'A' indicates that POWER/VS job accounting is required; a blank character indicates that POWER/VS accounting is not required.

[2] LOG Option. This byte contains a single alphabetic character; the character 'L' indicates that the JECL job statement is to be logged; a blank character indicates the opposite.

[3] GNSU will be overlaid by GNSR during initialization if the number of SNA remotes is smaller than the maximum number of SNA logical units.

Figure 5.41. Generation Table (Part 2 of 2)

COMMAND CONTROL BLOCK (CCB)

Definition macro: IPW$DCB

Bytes		Label of field	Description/function of field
Dec	Hex		
00-01	00-01	CBCT	Residual count
02	02	CBC1	First communication byte (see flags A)
03	03	CBC2	Second communication byte (see flags B)
04	04	CBSD	Device status byte (see flags C)
05	05	CBSC	Channel status byte
06	06	CBLC	LUB class (see flags D)
07	07	CBLN	LUB number within class
08	08	CBLI	LIOCS communication byte
09-11	09-0B	CBCA	CCW address
12	0C	CBPI	PIOCS communication byte
13-15	0D-0F	CBCS	CCW address in CSW
16	10	CBNX	First entry outside CCB

Flags A	WDE	X'04' = wait for device end
	AUIO	X'10' = accept unrecoverable I/O error
	UIO	X'20' = unrecoverable I/O error
	RODC	X'08' = return on data check
Flags B	CCR	X'01' = command chain retry option
	CHN9	X'02' = channel 9 overflow
Flag C	UE	X'01' = unit exception
	UNCK	X'02' = unit check
Flags D	EXR	X'80' = EXCP real
	PRU	X'01' = programmer unit

Figure 5.42. Command Control Block

CHANNEL COMMAND WORD (CCW)

Definition macro: IPW$DCW

Bytes		Label of field	Description/function of field
Dec	Hex		
00	00	CWCC	Command code
01-03	01-03	CWDA	Data address
04	04	CWFL	Chain byte (see flags below)
05	05	CWRE	Reserved
06-07	06-07	CWCT	Data length field

Flags	NOP	X'03' = NOP command
	CC	X'40' = command chaining
	SLI	X'20' = suppress incorrect length

Figure 5.43. Channel Command Word

SEPARATOR LINE AREA (SLA)

DSECTname: LWSL

The separator line area is used to produce separator pages and separator cards in list and punch writer output, by the logical writer IPW$$LW.

Bytes		Label of field	Description/function of fields
Dec	Hex		
00-119	00-77	LWPL	Print line
120-239	78-EF	LWSL	Separator line
120-142	78-8E	LWS0	Line delimiter. Contains all asterisks.
143-150	8F-96	LWS1	START/END separator
151-161	97-A1	LWS2	Job name
162-169	A2-A9	LWS3	Job number
170-177	AA-B1	LWS4	Job suffix number
178-196	B2-C4	LWS5	User information
197-208	C5-D0	LWS6	Date
209-219	D1-DB	LWS7	Time
220-239	DC-EF	LWS8	Line delimiter. Contains all asterisks.
240-247	F0-F7	LWCD	Double word for convert
248-255	F8-FF	LWSR	Save area for request word
256-259	100-103	LWSA	Start address
260-267	104-10B	LWJN	Work area
268	10C	LWFL	Filler
269-271	10D-10F	LWTC	Save area for restart information

Figure 5.44. Separator Line Area

How to Locate

This area is placed at the virtual data area address in the TCB.

OPEN 3540 DISKETTE WORK SPACE

DSECTname: OEWS

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F		Storage descriptor ('OEWS V7M0 cuu')
		OECB	3540 command control block
16-17	10-11	OECT	Residual count
18-19	12-13	OECM	Communications bytes
20-21	14-15	OEST	Device status
22-23	16-17	OELU	Device type and logical unit
24	18		Reserved for LIOCS
25-27	19-1B	OECA	First CCW
28	1C		Reserved for PIOCS
29-31	1D-1F	OECW	CCW address in CSW
32-87	20-57	OESV	Temporary register save area for the interface between functions
88-95	58-5F	OECV	Conversion work space
		OECP	3540 channel program
96-103	60-67	OEDO	Define operations or NOP
104-111	68-6F	OESK	Seek
112-119	70-77	OERD	Read label
120-123	78-7B	OESM	Mode setting argument
124-127	7C-7F	OESA	Seek argument (00CCHHRR)
128-207	80-CF	OELB	3540 input area and label test area (see Figures 5.46 and 5.47)
			Message buffers and work areas
208	D0	OML1	Message length of first line
		OMS1	First line of message output area
209-215	D1-D7	OMT1	Message identity
216-263	D8-107	OMT1	Message text of first line
264	108	OML2	Message length of second line
		OMS2	Second line of message output area
265	109	OMI2	Message identity
266-319	110-13F	OMT2	Message text of second line
320	140		Not used
321	141	OERL	Reply length
322-327	142-147	OERP	Reply input area
328-329	148	OECC	Cylinder number save area

Figure 5.45. 3540 Diskette Work Space (Part 1 of 2)

Bytes		Label of field	Description/function of field
Dec	Hex		
			Physical reader information indicators. The following indicators are copied from the physical work space to prevent them from being destroyed should the open be unsuccessful. On a successful open, the indicators in the physical work space are overwritten by these updated indicators. On an unsuccessful open, only the open indicator 'PEOC' will be updated with the stop code 'S'. (See Figure 5.15.)
330-331	14A-14B	WERL	Record length (copy of PERL)
		WESI	Sequence ID (copy of PESI)
332	14C	WEMI	Multivolume identification (copy of PEMI)
333	14D	WESN	Volume sequence number (copy of PESN)
334	14E	WEOD	Number of opened diskettes (copy of PEOD)
335	14F	WEND	Number of diskettes to be read (copy of PEND)
336-343	150-157		Not used

Figure 5.45. 3540 Diskette Work Space (Part 2 of 2)

Bytes		Label of field	Description/function of field
Dec	Hex		
		VOLL	Diskette volume 1 label
128-131	80-83	VLID	Volume label ID and number
132-137	84-89	VLSN	Volume serial number
138	8A	VLAI	Volume access indicator
139-164	8B-A4		Reserved
165-178	A5-B2	VLDI	Volume owner identity
179-202	B3-CA		Reserved
203	CB	VLPL	Physical record length
204-205	CC-CD	VLRS	Physical record sequence code
206	CE		Reserved
207	CF	VLST	Label standard version (W)

Figure 5.46. 3540 Volume 1 Label Layout in Label Test Area (OELB)

Bytes		Label of field	Description/function of field
Dec	Hex		
		HDRL	Diskette header 1 label
128-131	80-83	HDID	Header label ID and number
132	84		Reserved
133-140	85-8C	HDFI	File identifier
141-149	8D-95		Reserved
150-154	96-9A	HDBL	Block length of data record
155	9B		Reserved
156-160	9C-A0	HDLO	Begin of extent (CCHRR)
161	A1		Reserved
162-166	A2-A6	HDHI	End of extent (CCHRR)
167	A7		Reserved
168	A8	HDBI	Bypass indicator (B)
169	A9	HDFS	File security indicator (S)
170	AA	HDWP	File write protection indicator (P)
171	AB	HDEI	Basic exchange indicator (,E)
172	AC	HDMV	Multivolume indicator (,C,L)
173-174	AD-AE	HDSN	Volume sequence number
175-180	AF-B4	HDCR	Creation date
181-193	B5-C1		Reserved
194-199	C2-C7	HDEX	Expiration date
200	C8	HDVI	Verify indicator (,V)
201	C9		Reserved
202-206	CA-CE	HDED	End of data address (CCHRR)
207	CF		Reserved

Figure 5.47. 3540 Header 1 Label Layout in Label Test Area (OELB)

ASYNCHRONOUS SERVICE ANCHOR BLOCK

Definition Macro: IPW$DAB

The asynchronous service anchor block contains the queue pointers for all service requests to be performed by the service subtask.

Storage for the anchor block is acquired the first time a POWER/VS task issues an attach request. The anchor block exists as long as POWER/VS is active.

When a POWER/VS task uses asynchronous service, an IPW$IAS TYPE=ATTACH request must be issued first, before any other IPW$IAS TYPE=SERVICE request can be made.

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F		Storage descriptor (ASWS)
16-19	10-13	ABNEXT	Address of first service request in queue (zero if no entry)
20-23	14-17	ABLAST	Address of last request in queue
24-27	18-1B	ABECB	Subtask event control block
28-31	1C-1F	ABLCK	Lock word
32-33	20-21	ABTIK	Task identification key (TIK) of asynchronous service subtask.
34-35	22-23	ABUSCT	Use count
36-39	24-27	ABADR1	Address of SETPRT routine (IJVSRPDV) in SVA
40-43	28-2B	ABECB2	Event control block
44-51	2C-33	ABWFLD	General purpose work field
52-55	34-37		Unused
		ABLN	Length of anchor block

Figure 5.47.1. Asynchronous Service Anchor Block

SERVICE REQUEST BLOCK (SRB)

Definition Macro: IPW$DSR

A service request block is created whenever a service request is passed to asynchronous service for processing.

During the time asynchronous service is performing the service request, the SRBs are chained together.

Asynchronous service handles the request on a 'first-in first-out' basis.

Bytes		Label of field	Description/function of field
Dec	Hex		
00-03	00-03	SRBNEXT	Address of next SRB in chain
04	04	SRBREQ	Request type C'S' - SETPRT request (SRBSPR)
05	05	SRBTRC	Return code (low-order byte of register 15)
06-07	06-07		Reserved
08-11	08-0B	SRBECB	Event control block (POWER/VS task, issuing service request, waits on this ECB, which is posted by the service subtask).
12-23	OC-17	SRBPARM	Request parameter list: This field is broken down differently for each service request type: SETPRT request ------------------------------ Byte 00-03 Address of SETPRT parameter list 04-11 unused
-	-	SRBLN	Length of service request block

Figure 5.47.2. Service Request Block

ASSIGN/UNASSIGN WORK SPACE

The work space is primarily used as a register save area and to contain printer setup information when a 3800 printer is being unassigned and asynchronous service is invoked to set up the printer with the system/hardware defaults.

DSECT Name: LUWS

Bytes		Label of field	Description/function of field
Dec	Hex		
00-15	00-0F		Storage descriptor ('LUWS')
16-71	10-47	LUSV	Temporary register save area for the interface between functions.
72-79	48-4F	LUGR	Save area for registers 14-15. Used when another function is invoked.
80-147	50-93	LUSP	SETPRT parameter list
148-151	94-97		Not used
		LULN	Length of assign/unassign work space

Figure 5.47.3. Assign/Unassign Work Space

3800 TCB EXTENSION AREA

This control block is constructed:

1. At job execution time whenever a 3800 printer is being spooled.
2. At print time when the printer is a 3800. Like the TCB, the 3800 TCB extension area is built in real storage and exists as long as the task exists.

The control block contains device status information of the current or new printer setup. The TCB extension is pointed to by the TC3E field in the TCB. DSECT name is IPW$DTE.

Bytes		Lable of field	Description/function of field
Dec	Hex		
00-01	00-01	TE38LU	Logical unit number or physical device address
02	02	TE38RQB	Request byte
		TE38SRI	x'80' SETPRT required
		TE38DFLT	x'40' default setup required
		TE38DGMT	x'20' segmentation required
03	03	TE38FLG	Unused
04	04	TE38CGI	Current copy group index
05-07	05-07		Reserved for future use
			General work area
08-27	08-1B	TE38GW	This area may be broken down into fields in whatever way is required by the task.
			Overlay area for EXTRACT/MODCTB parameter list
08-27	08-1B	SVC98PL	Parameter list for EXTRACT/MODCTB
08	08	SVC98ID	Indicator for SVC 98 routine
		PUB2EXID	x'01' ID for extract PUB2 table
		PUB2MOID	x'F0' ID for modify PUB2 table
09	09	SVC98FL	Flag field
		PHYSUNIT	x'01' physical unit ID flag
10-11	0A-0B	SVC98RS	Reserved
12-13	0C-0D	SVC98LN	Length of user area
14-15	0E-0F	SVC98DI	Displacement in PUB2 table
16-19	10-13	SVC98AR	Address of user area
20-23	14-17	SVC98SE	Address of logical or physical unit
24-27	18-1B	SVC98PI	Address of user-specified PIK
28-31	1C-1F		Reserved for future use

Figure 5.47.4. 3800 TCB Extension Area

Bytes		Lable of field	Description/function of field
Dec	Hex		
			SETPRT parameter list
32	20	SPLLNGTH	**Length of SETPRT parameter list**
33	21	SPLFLAG1	Flag byte 1
		SPLLUADR	x'80' Field SPLLUNIT contains the address of the LUB
		SPLINIT	x'40' INIT=Y was specified
		SPLBURY	x'30' BURST=Y was specified
		SPLBURD	x'20' BURST=* was specified
		SPLBURN	x'10' BURST=N was specified
		SPLUDCHK	x'08' DCHK=U was specified
		SPLNOMSG	x'04' suppress setup message to operator
		SPLMARK	x'02' SEP=M was specified
		SPLOFFST	x'01' SEP=0 was specified
34	22	SPLFLAG2	Flag byte 2
		SPLFCBV	x'80' FCB verification was requested
		SPLQSPRT	x'40' Query SETPRT was requested
		SPLTRCY	x'30' TRC=Y was specified
		SPLTRCN	x'10' TRC=N was specified
		SPLDEBTR	x'0E' DEBUG=TRAC was requested
		SPLDEBDU	x'0A' DEBUG=DUMP was requested
		SPLDEBTE	x'06' DEBUG=TERM was requested
		SPLDEBNO	x'02' DEBUG=NORM was requested
		SPLRSVD3	x'01' Reserved

Figure 5.47.5. SETPRT Parameter List (Part 1 of 2)

Bytes		Lable of field	Description/function of field
Dec	Hex		
35	23	SPLRSVD1	Reserved
36-39	24-27	SPLFCB	Last 4 characters of the FCB phase name
40-55	28-37	SPLCHAR	Character arrangement tables
40-43	28-2B	SPLCHAR1	Last 4 characters of first translate table
44-47	2C-2F	SPLCHAR2	Last 4 characters of second translate table
48-51	30-33	SPLCHAR3	Last 4 characters of third translate table
52-55	34-37	SPLCHAR4	Last 4 characters of fourth translate table
56-59	38-3B	SPLCMCHR	Last 4 characters of the translate table phase name specified in the MODIFY keyword.
60-63	3C-3F	SPLCPMOD	Last 4 characters of the copy modification phase name
64-67	40-43	SPLFORMS	Paper forms ID
68-71	44-47	SPLFLASH	Forms overlay ID (Flash ID)
72-79	48-4F	SPLCOPYG	Copy groups
80	50	SPLCINDX	Copy group index
81	51	SPLFLSHC	Flash count
82-83	52-53		Reserved
84-87	54	SPLLUNIT	Address of LUB
84-85	54-55		Reserved
86-87	56-57	SPLLUSYS	Logical unit number
88-91	58-5B	SPLREG1	Register save area
92-95	5C-5F	SPLREG2	Register save area
96-99	60-63	SPLREG3	Register save area
100-119	64-77		Reserved for SETPRT parameter list extension

Figure 5.47.5. SETPRT Parameter List (Part 2 of 2)

SPOOL PARAMETER LIST (SPL)

The SPL is the means of cross-partition communication between POWER/VS and another program using the PUTSPOOL, GETSPOOL or CTLSPOOL interface. When POWER/VS receives control, the SPL address is located at the user's XECB+5, and spool management initializes the address in the TCB (TCPL) for use by POWER/VS. The external interface is described in the DOS/VS Supervisor and I/O Macros manual.

Bytes		Label of field	Description/function of field
Dec	Hex		
00-02	00-02	SPHD	SPL header ('SPL')
03-10	03-0A	SPJB	Job name
11	0B		Blank delimiter
12	0C	SPER	Error feedback
		SPIA	X'80' - Invalid address
		SPLA	X'88' - Invalid SPL address
		SPPA	X'84' - Invalid POWER buffer address
		SPBA	X'82' - Invalid data buffer chain
		SPPP	X'40' - Diagnostic logged by POWER/VS
		SPSP	X'48' - During PUTSPOOL processing
		SPLP	X'44' - During GETSPOOL processing
		SPCP	X'42' - During CTLSPOOL processing
		SPAP	X'41' - POWER/VS terminated
		SPUE	X'20' - Processing error
		SPLE	X'28' - Invalid CTLSPOOL request
		SPBE	X'24' - Loop in PUTSPOOL buffer chain; or, more than 4096 buffers used per request
		SPPE	X'22' - GETSPOOL was unable to locate output file by specified job name, job class, and dispatchable POWER/VS disposition; or, requested output file is in use
		SPSE	X'21' - Buffer area too small (88-byte minimum)
		SPPI	X'10' - Invalid parameter
		SPJI	X'18' - Invalid job name
		SPCI	X'14' - Invalid class
		SPDI	X'12' - Invalid disposition
		SPNR	X'00' - Normal return
		SPLR	X'08' - End-of-data on GETSPOOL
13	0D	SPR1	PUTSPOOL request type
		SPEJ	X'40' - The last data record for informal reader job is contained in this PUTSPOOL request
14	0E	SPR2	CTLSPOOL request type
		SPRP	X'01' - Route to new priority
		SPRD	X'02' - Route to new disposition
		SPRC	X'04' - Route to new class
		SPRJ	X'08' - Route to new remote ID
		SPCX	X'10' - Cancel from RDR queue
		SPSC	X'20' - Scratch from LST queue
		SPST	X'40' - Display job status
		SPPC	X'80' - User-supplied POWER command
15	0F	SPR3	GETSPOOL request type
		SPLD	X'01' - GETSPOOL request
		SPPO	X'02' - Position on Q-record
		SPBR	X'04' - Position on line number
		SPCO	X'08' - Return control characters
16-23	10-17	SPXR	PUTSPOOL user's XECB name
24-31	18-1F	SPXL	GETSPOOL/CTLSPOOL user's XECB name

Bytes		Label of field	Description/function of field
Dec	Hex		
32-35	20-23	SPCB	Address current PUTSPOOL buffer area
36-39	24-27	SPPB	Address user-supplied buffer area for POWER
40-43	28-2B	SPBL	Data buffer area length
44-47	2C-2F	SPRL	Data record length
		SPRS	Browse control
48	30	SPSN	Signed browse start control
49-51	31-33	SPCT	Browse start line number
52	34	SPCL	LST output class
53	35	SPDP	LST output disposition
54	36	SPCC	Print/POWER control character
55	37	SPSQ	Display job status return
			C'N' - Not on POWER/VS queues
			C'R' - On RDR queue
			C'L' - On LST queue
56	38	SPQD	Job disposition on RDR/LST queue
57	39		Unused
58	3A	SPNV	CTLSPOOL new value
			PRI=
			DISP=
			FLASS=
			REMOTE=
59	3B	SPLN	Spool parameter list length

This figure contains a set of examples which illustrate the interrelationships between the POWER/VS control blocks and tables. The examples are based on the assumed position of six separate jobs at an assumed point in time (time "t"). Note that the illustrations do not represent true situations.

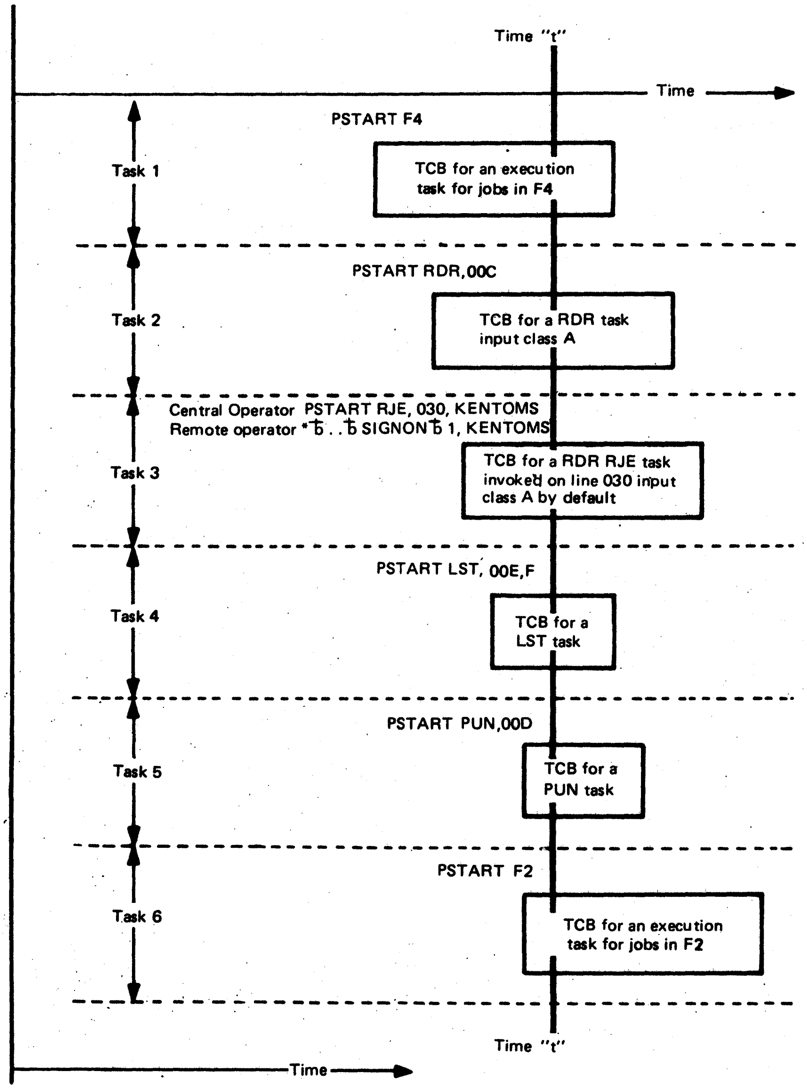

Time "t"
Time
PSTART F4
Task 1
TCB for an execution task for jobs in F4
PSTART RDR,00C
Task 2
TCB for a RDR task input class A
Central Operator PSTART RJE, 030, KENTOMS
Remote operator *ƀ . . ƀ SIGNONƀ 1, KENTOMS
Task 3
TCB for a RDR RJE task invoked on line 030 input class A by default
PSTART LST, 00E,F
Task 4
TCB for a LST task
PSTART PUN,00D
Task 5
TCB for a PUN task
PSTART F2
Task 6
TCB for an execution task for jobs in F2
Time "t"
Time

Figure 5.48. Part 1: Assumed position of six tasks at time "t"

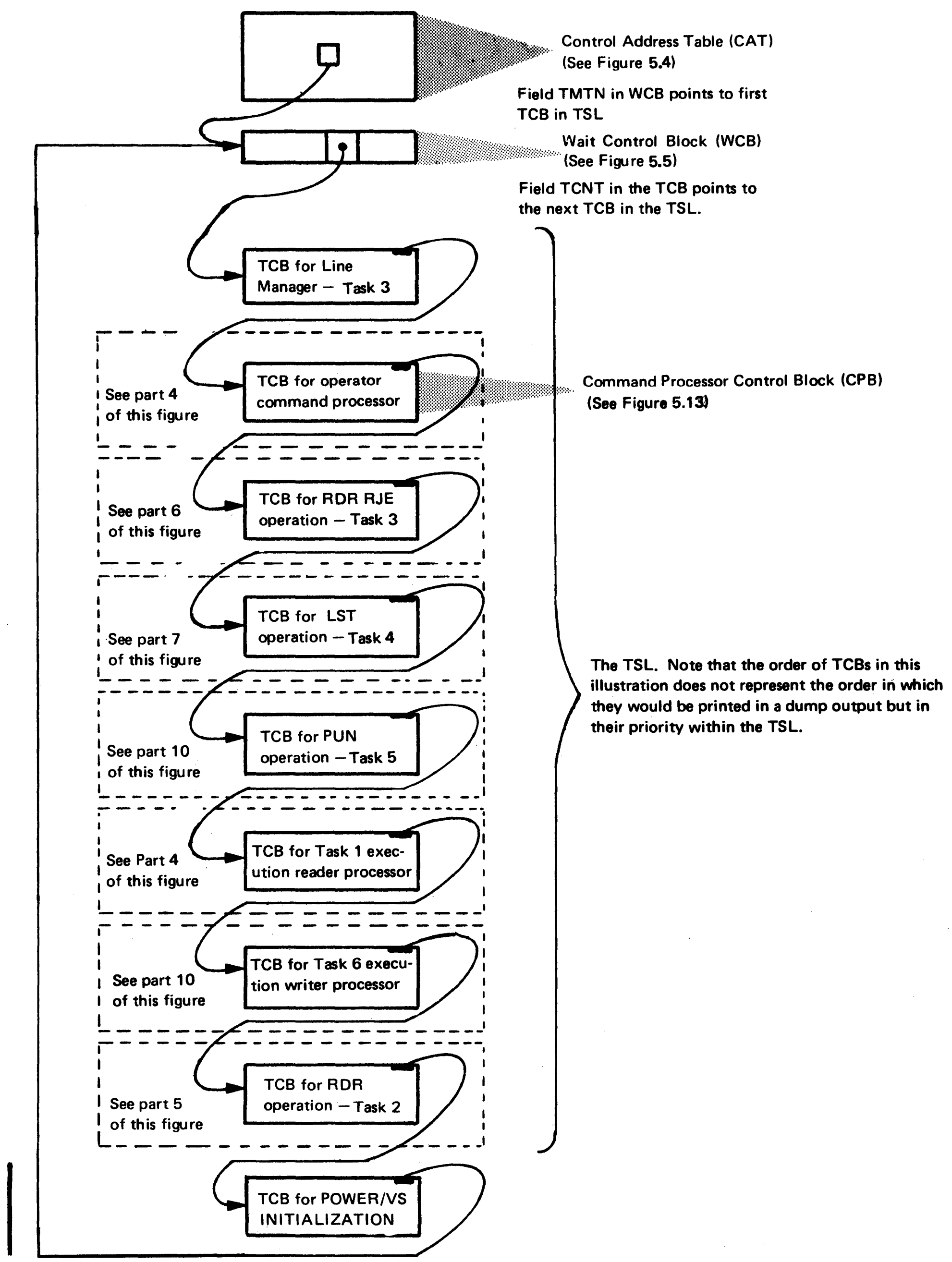

Control Address Table (CAT)
(See Figure 5.4)
Field TMTN in WCB points to first TCB in TSL
Wait Control Block (WCB)
(See Figure 5.5)
Field TCNT in the TCB points to the next TCB in the TSL.
TCB for Line Manager – Task 3
See part 4 of this figure
TCB for operator command processor
Command Processor Control Block (CPB)
(See Figure 5.13)
See part 6 of this figure
TCB for RDR RJE operation – Task 3
See part 7 of this figure
TCB for LST operation – Task 4
The TSL. Note that the order of TCBs in this illustration does not represent the order in which they would be printed in a dump output but in their priority within the TSL.
See part 10 of this figure
TCB for PUN operation – Task 5
See Part 4 of this figure
TCB for Task 1 execution reader processor
See part 10 of this figure
TCB for Task 6 execution writer processor
See part 5 of this figure
TCB for RDR operation – Task 2
TCB for POWER/VS INITIALIZATION

Figure 5.48. Part 2: The Task Control Blocks (TCB) linked into the Task Selection List (TSL)

Before proceeding it is necessary to understand the method of presentation used in the other parts of the figure.

Each table or block in the following diagrams is represented by a block divided horizontally into eight boxes. Each box represents a 4-byte field within a table which is 32 bytes across, thus matching the number of bytes printed in each line of a standard dump as shown below.

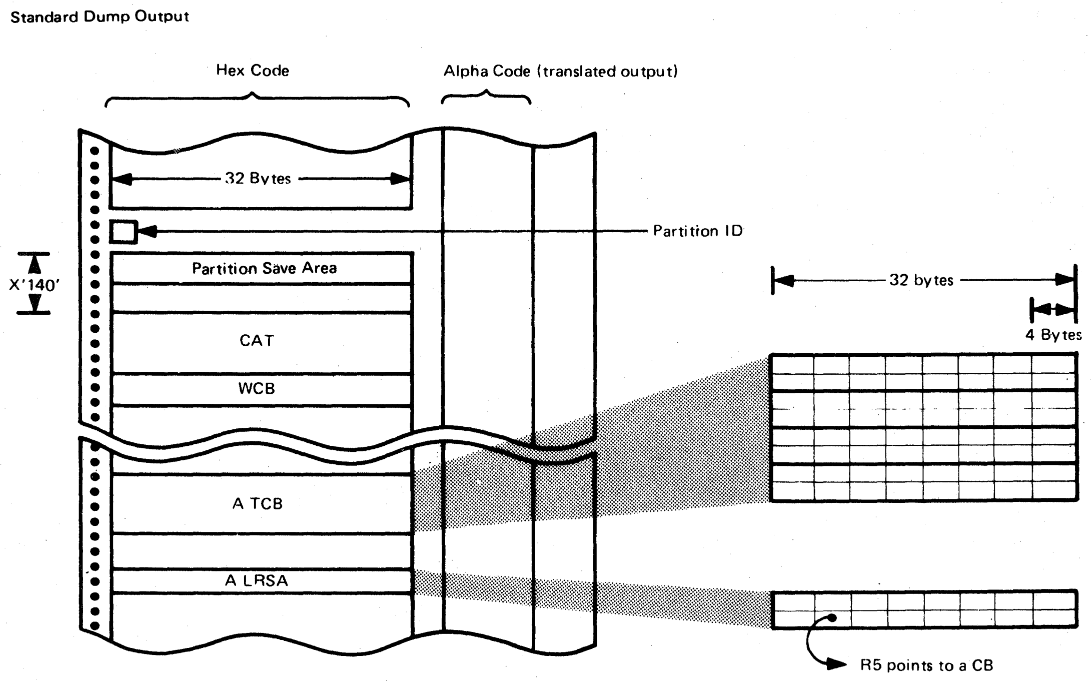


Figure 5.48. Part 3: Method of presentation and task conditions

These tables in the permanent area are illustrated in the order in which they are printed in the dump. The symbol ↑ indicates a pointer. For example, R5↑ queue record indicates that register 5 contains the address of a queue record.

Furthermore, the assumed status of each task at time "t" is described in the figure captions for each task represented in the following parts of this figure.

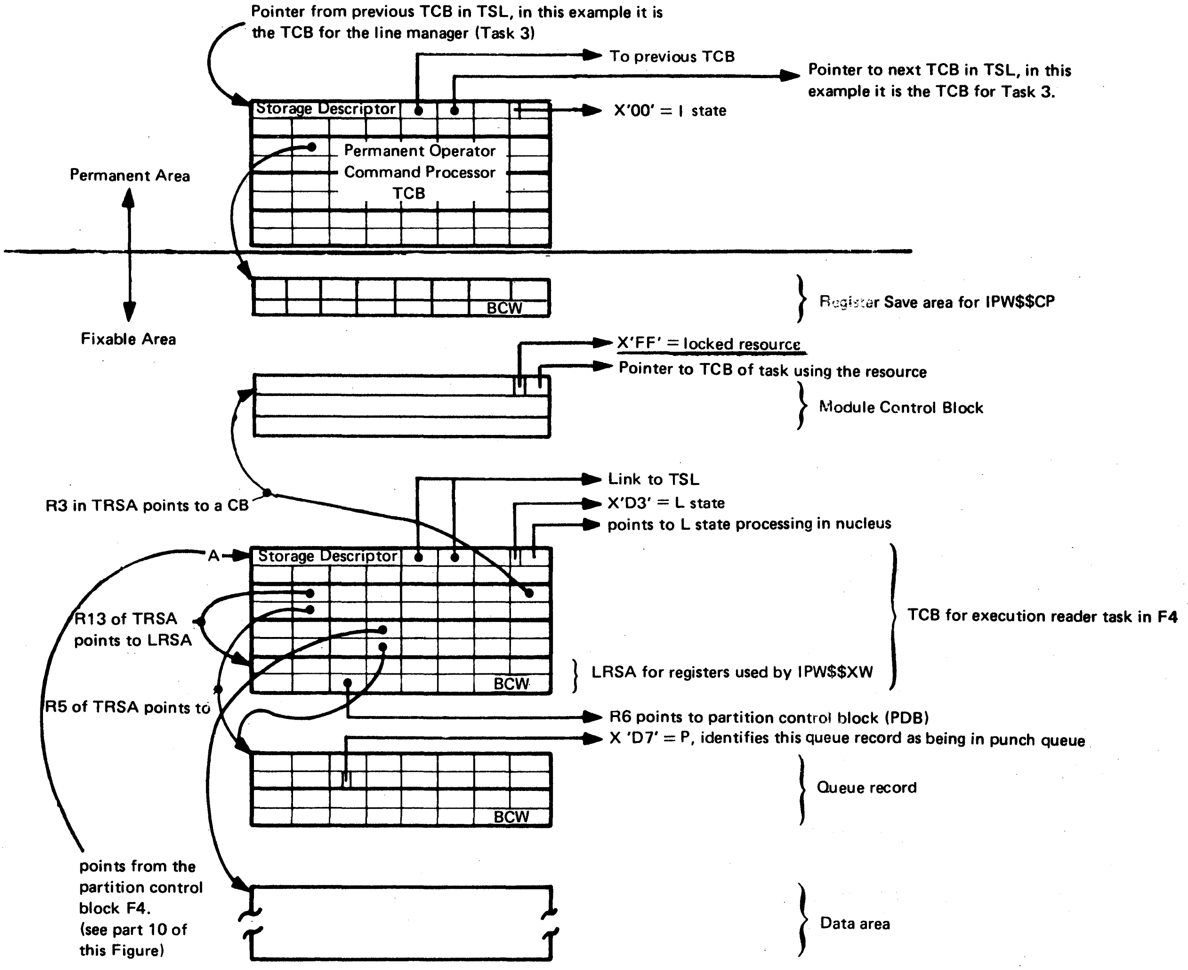

Pointer from previous TCB in TSL, in this example it is the TCB for the line manager (Task 3)
To previous TCB
Pointer to next TCB in TSL, in this example it is the TCB for Task 3.
Storage Descriptor
X'00' = I state
Permanent Operator Command Processor TCB
Permanent Area
Fixable Area
BCW
Register Save area for IPW$$CP
X'FF' = locked resource
Pointer to TCB of task using the resource
Module Control Block
Link to TSL
X'D3' = L state
points to L state processing in nucleus
R3 in TRSA points to a CB
A
Storage Descriptor
TCB for execution reader task in F4
R13 of TRSA points to LRSA
LRSA for registers used by IPW$$XW
BCW
R5 of TRSA points to
R6 points to partition control block (PDB)
X 'D7' = P, identifies this queue record as being in punch queue
Queue record
BCW
points from the partition control block F4. (see part 10 of this Figure)
Data area

Figure 5.48. Part 4: Relations between data areas in use by task 1

Task 1:

- The permanent operator command processor TCB in the permanent area is in I state waiting for the operator to enter a command.

- Execution reader processor TCB in L state - task waiting for a locked resource. (The Module Control Block of Data File 2 is being used by another task.)

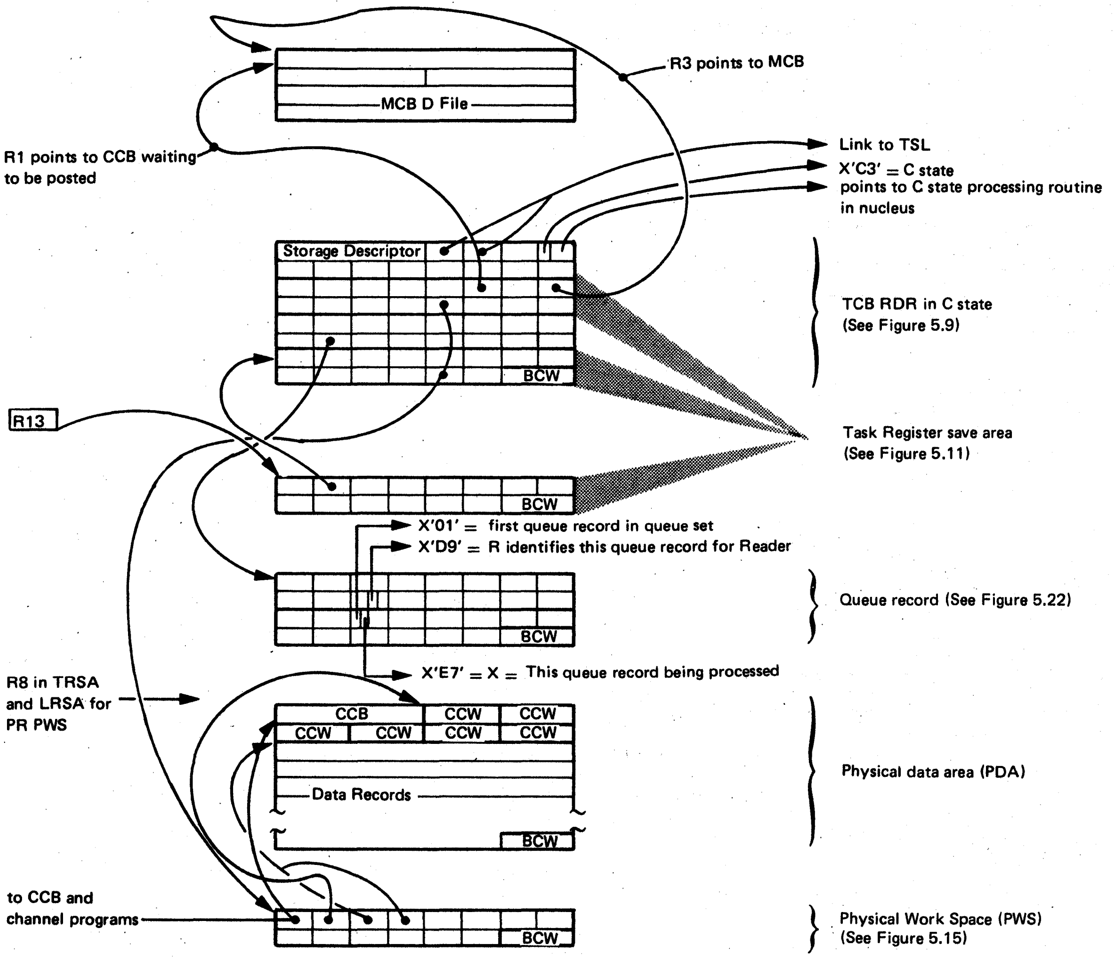

R3 points to MCB
MCB D File
R1 points to CCB waiting to be posted
Link to TSL
X'C3' = C state
points to C state processing routine in nucleus
Storage Descriptor
TCB RDR in C state (See Figure 5.9)
BCW
R13
Task Register save area (See Figure 5.11)
BCW
X'01' = first queue record in queue set
X'D9' = R identifies this queue record for Reader
Queue record (See Figure 5.22)
BCW
X'E7' = X = This queue record being processed
R8 in TRSA and LRSA for PR PWS
CCB
CCW
CCW
CCW
CCW
CCW
CCW
Data Records
BCW
Physical data area (PDA)
to CCB and channel programs
BCW
Physical Work Space (PWS) (See Figure 5.15)

Figure 5.48. Part 5: Relations between data areas in use by task 2, showing the PWS and PDA

Task 2:

- TCB for Reader task, class A, in C state waiting for disk output (CCB posting).

- Task was executing code in a logical routine.

- CCW chain for input consists of six CCWs.

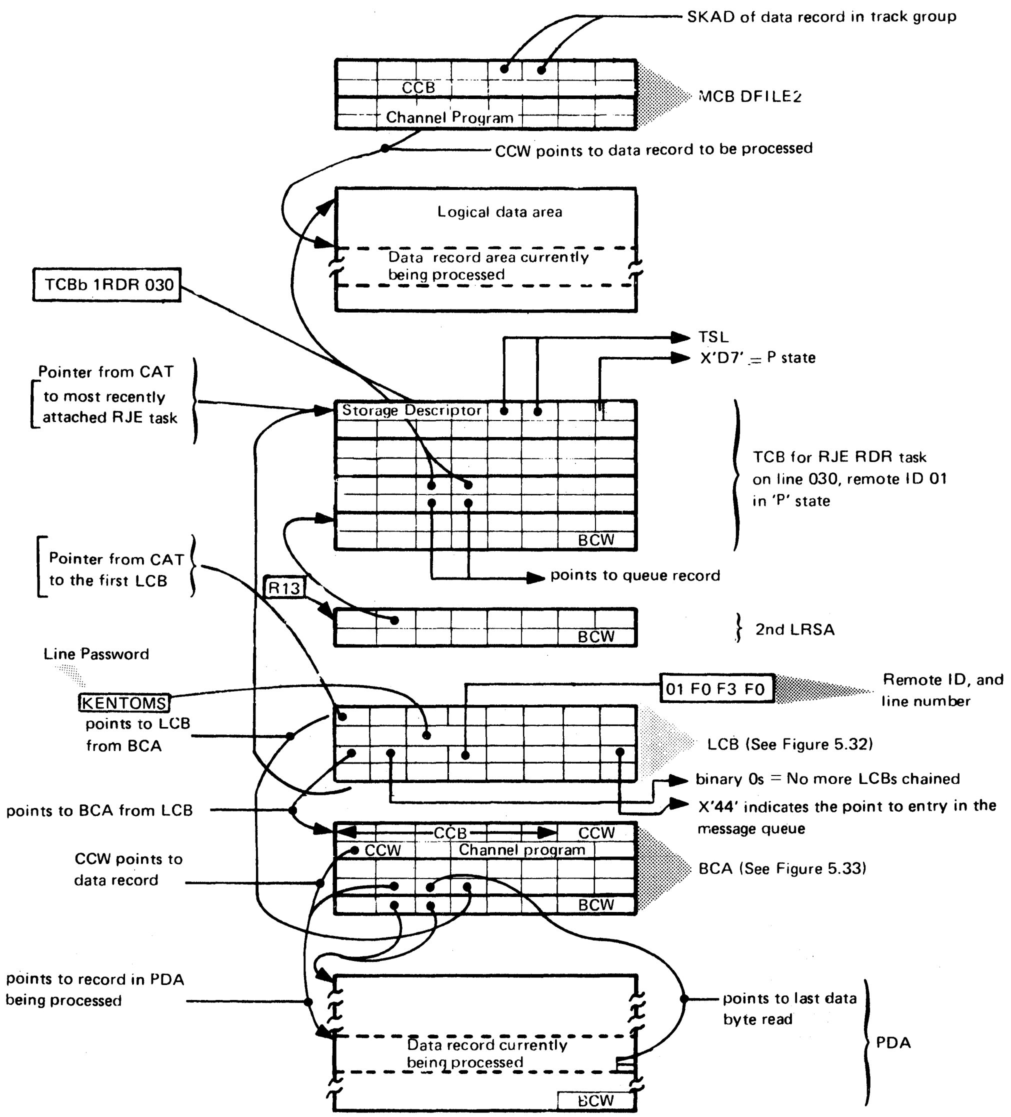

SKAD of data record in track group
CCB
Channel Program
MCB DFILE2
CCW points to data record to be processed
Logical data area
Data record area currently being processed
TCBb 1RDR 030
TSL
X'D7' = P state
Pointer from CAT to most recently attached RJE task
Storage Descriptor
TCB for RJE RDR task on line 030, remote ID 01 in 'P' state
BCW
Pointer from CAT to the first LCB
R13
points to queue record
BCW
2nd LRSA
Line Password
KENTOMS
points to LCB from BCA
01 F0 F3 F0
Remote ID, and line number
LCB (See Figure 5.32)
binary 0s = No more LCBs chained
X'44' indicates the point to entry in the message queue
points to BCA from LCB
CCB
CCW
CCW
Channel program
BCA (See Figure 5.33)
CCW points to data record
BCW
points to record in PDA being processed
points to last data byte read
PDA
Data record currently being processed
BCW

Figure 5.48. Part 6: Relations between data areas in use by task 3

Task 3:

- RDR task on RJE line 30 from remote terminal 01.
- Task in function PD transferring a data record from the PDA to the LDA.
- The function was called by the logical routine IPW$$LR.
- During execution of the code in PUTDA a page fault occurred.
- No more LCBs are chained.

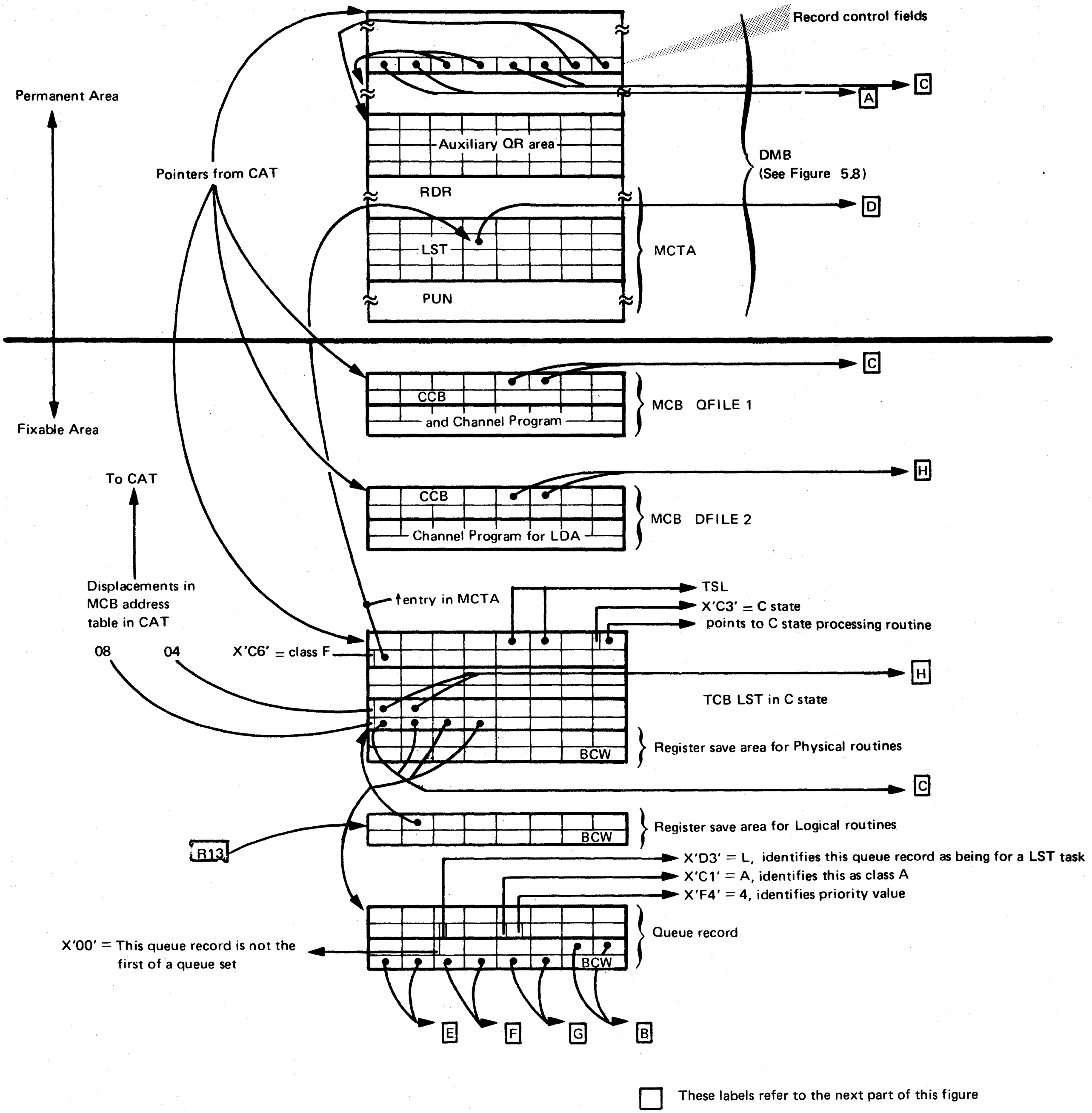
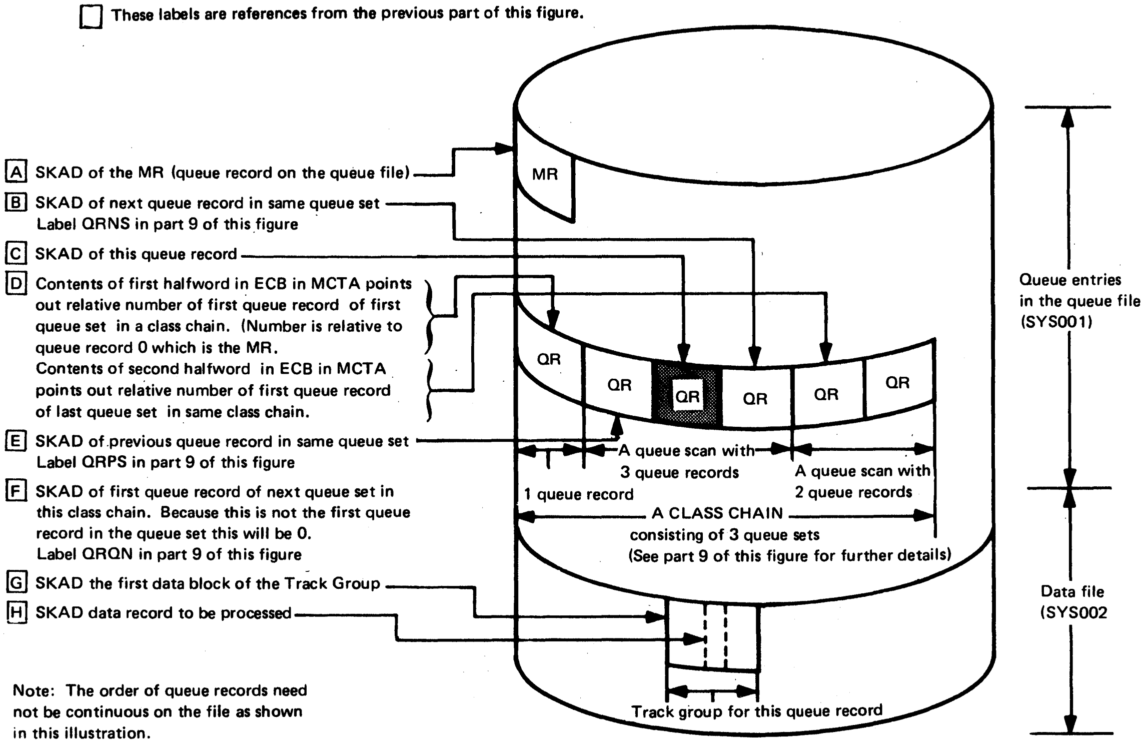


Figure 5.48. Part 7: Relations between data areas in use by task 4 showing pointers to the Queue and Data files

Task 4:

- LST task is in C state, output class F, that has read a queue record but not been dispatched.
- The task is executing code in IPW$$NQ being called from the IPW$$LW routine.
- The queue record read is the second queue record of the second queue set in class chain F in the LST queue on the queue file.

□ These labels are references from the previous part of this figure.



A SKAD of the MR (queue record on the queue file)

B SKAD of next queue record in same queue set Label QRNS in part 9 of this figure

C SKAD of this queue record

D Contents of first halfword in ECB in MCTA points out relative number of first queue record of first queue set in a class chain. (Number is relative to queue record 0 which is the MR. Contents of second halfword in ECB in MCTA points out relative number of first queue record of last queue set in same class chain.

E SKAD of previous queue record in same queue set Label QRPS in part 9 of this figure

F SKAD of first queue record of next queue set in this class chain. Because this is not the first queue record in the queue set this will be 0. Label QRQN in part 9 of this figure

G SKAD the first data block of the Track Group

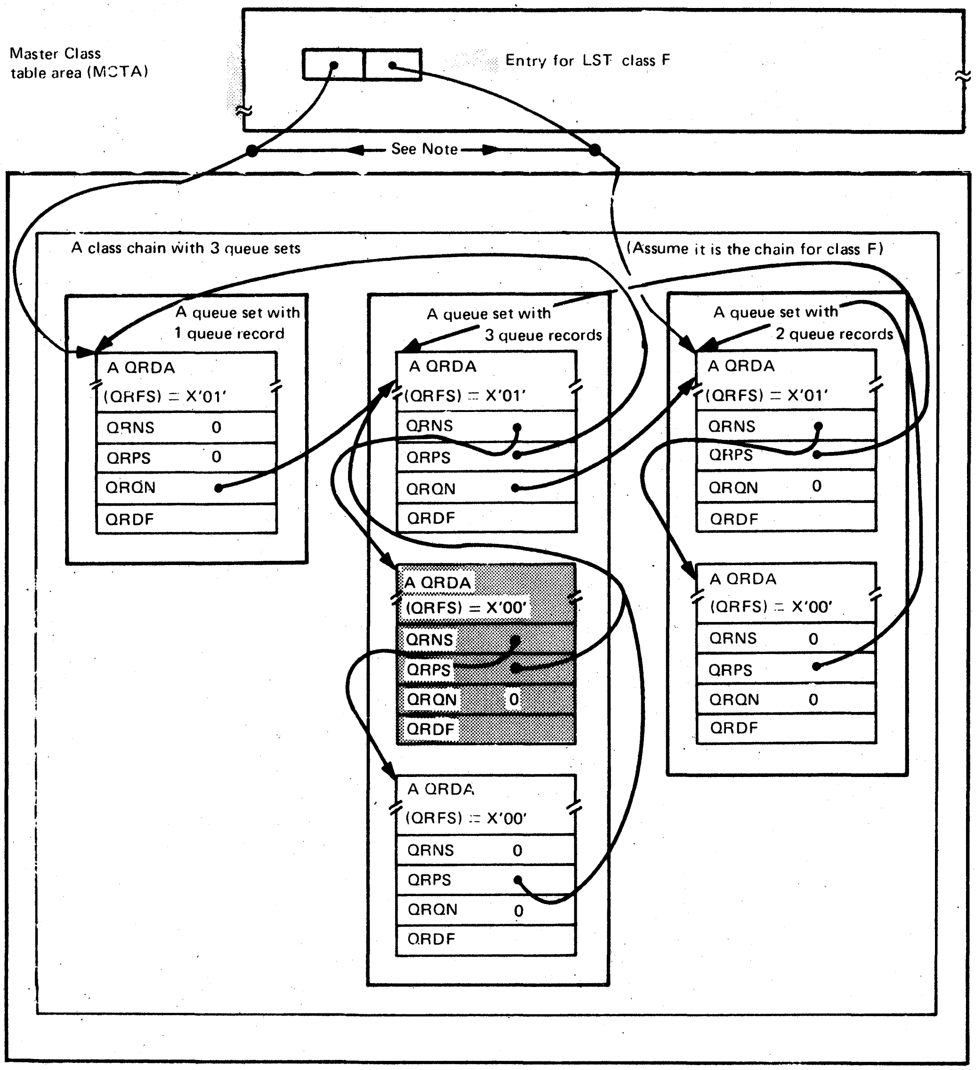
H SKAD data record to be processed

Note: The order of queue records need not be continuous on the file as shown in this illustration.

MR
QR
QR
QR
QR
QR
QR
1 queue record
A queue scan with 3 queue records
A queue scan with 2 queue records
A CLASS CHAIN consisting of 3 queue sets (See part 9 of this figure for further details)
Track group for this queue record
Queue entries in the queue file (SYS001)
Data file (SYS002

Figure 5.48. Part 8: The queue and data file of task 4.

The queue record being processed (shaded) is the second queue record of the second queue set in a class chain (F).



Note: All pointer addresses are seek addresses, except those in the MCTA entries, which point to queue records on the queue file relative to the MR which is record 1. (The shaded queue record corresponds to the queue record being processed as shown in part 7 of this figure).

Figure 5.48. Part 9: The Queue Set of task 4

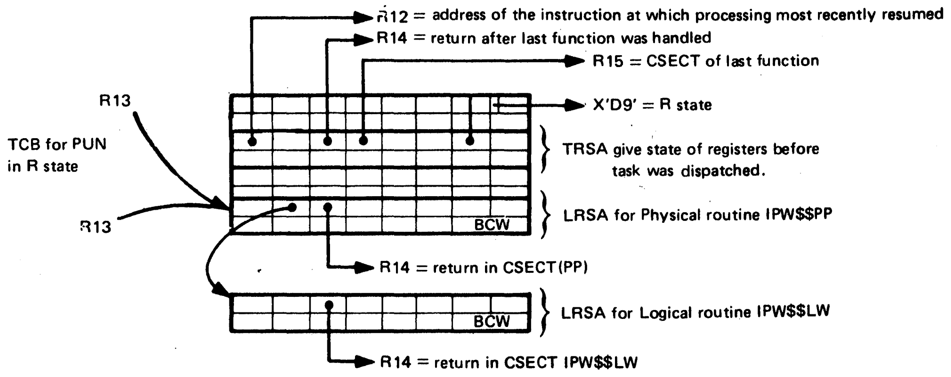

R12 = address of the instruction at which processing most recently resumed
R14 = return after last function was handled
R15 = CSECT of last function
R13
TCB for PUN in R state
X'D9' = R state
TRSA give state of registers before task was dispatched.
R13
BCW
LRSA for Physical routine IPW$$PP
R14 = return in CSECT(PP)
BCW
LRSA for Logical routine IPW$$LW
R14 = return in CSECT IPW$$LW

Figure 5.48. Part 10: Relations between data areas in use by task 5

Task 5:

- Punch task in R state executing code in PP routine after the IPW$OLI macro has been issued to open the interface with LW routine.

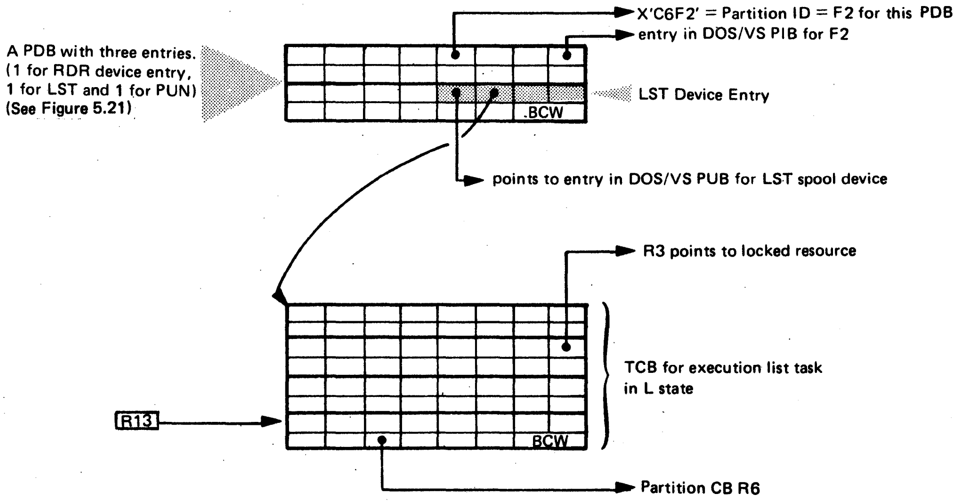

X'C6F2' = Partition ID = F2 for this PDB
entry in DOS/VS PIB for F2
A PDB with three entries. (1 for RDR device entry, 1 for LST and 1 for PUN) (See Figure 5.21)
LST Device Entry
BCW
points to entry in DOS/VS PUB for LST spool device
R3 points to locked resource
TCB for execution list task in L state
R13
BCW
Partition CB R6

Figure 5.48. Part 11: Relations between data areas in use by task 6, showing the PDB for this execution list task

Task 6:

- Execution List processor task for F2. Task is in L state waiting for a locked resource.

Section 6: Diagnostic Aids

This section consists of error diagnostic flowcharts that indicate a method of dump analysis to isolate the cause of a software error (a bug) in the POWER/VS SCP.

The flowcharts do not represent the only method of analysis but are hints and suggestions about "where and what" to look for in a dump containing the POWER/VS partition.

The level of analysis becomes more detailed if the analysis progresses to a lower level.

Recommendations on the use of debugging aids as described in DOS/VS Serviceability Aids and Debugging Procedures, GC33-5380, are given, together with suggestions regarding program reruns in order to eliminate components.

The section begins with a list of general debugging hints.

REFERENCE LIST OF GENERAL DEBUGGING HINTS

1. The standalone dump and SYSVIS (PDS) dump
2. Identifying the POWER/VS partition (the partition in which POWER/VS is initialized)
3. Identifying pages belonging to the fixable area
4. Identifying the start of the pageable area
5. Locating and identifying control blocks, tables and areas
6. Identifying the start of a CSECT
7. Establishing the "level" of a CSECT
8. Determining the active routine and analyzing the register save areas.
9. Analyzing event control blocks
10. Using the buffer control words
11. Analyzing TCBs
12. Establishing queue records in queue sets and class chains
13. RJE, BSC I/O track wraparound buffer
14. POWER/VS file dump program
15. Establishing the last command issued
16. An aid to eliminate components
17. Problems related to VTAM.

General Debugging Hints

1. STANDALONE DUMP

It is recommended that the user generate a formatted standalone dump with translation (DUMPGEN parameter, FORMAT=YES. See DOS/VS Serviceability Aids and Debugging Procedure for detailed description.)

This dump should always be used when a standalone dump is required. Formatted page tables will save time in converting virtual to real addresses.

After executing a standalone dump, it is advisable to use the SYSVIS dump, as recommended and described in Section 2 of DOS/VS Serviceability Aids and Debugging Procedures, to dump, or copy, the POWER/VS virtual partition. This dump will be required to complete an offline analysis of the error.

2. IDENTIFYING THE POWER/VS PARTITION

The start of the POWER/VS partition can be easily identified in the translated portion of any dump by the name given to the POWER macro. The characters CAT, ten lines under that name, identify the control address table.

3. IDENTIFYING FIXED PAGES

The address of the first page in the fixable area is contained in bytes 18-1B of the CAT. Since each page is 2K bytes, the start of other pages in the fixable area can be calculated. Also, by examining the contents of the Page Control Table in the SCB the amount of pages and usage of each page can be established.

4. IDENTIFYING THE START OF THE PAGEABLE AREA

The address of the pageable area is contained in bytes 1C-1F of the CAT.

5. LOCATING AND IDENTIFYING CONTROL BLOCKS, TABLES, AND AREAS

In the Permanent Area

Abbreviated* Mnemonic of Table or Area	Pointer to or Address of the Table/Area	Identification in the translated dump output or remarks
AQRA		Execution date of POWER/VS
CAT	Start of POWER/VS partition save area (PSA***) plus X'140'	CAT and version/modification level
MCTA starts at X'280' of .DMB		No special identity
MLT	X'148' - X'157' of DMB	No special identity
MMB	X'74' - X'77' of CAT	MMB
MRA		Execution date of POWER/VS
POWER/VS PSA***	X'5C' - X'5F' of SYSCOM	Name of POWER/VS generated
DMB	X'68' - X'6B' of CAT	DMB
SCB	X'70' - X'73' of CAT	SCB
SYSCOM	X'80' - X'83' of LAS **	
Start of POWER/VS part	X'5C' - X'5F' of SYSCOM also X'14' - X'17' of CAT	Name of POWER/VS generated
WCB	X'D8' - X'DB' of CAT	WCB

* Refer to "Abbreviation List" in Appendix A.
** LAS = Low address storage as described in DOS/VS Serviceability and Debugging Aids.
***PSA = Partition save area as described in DOS/VS Serviceability Aids.

Figure 6.1. Part 1: Locating and Identifying Control Blocks, Tables and Areas in the Permanent Area

In the Fixable Area (Actual tables present depend on task requirements)

Abbreviated * Mnemonic of Table or Area	Pointer to or Address of the Table/Area	Identification in the translated dump output
Account file seek address	X'180' - X'187' of DMB	
ACB	X'6C' - X'6F' of CAT	
AQRA	X'F8' - X'FB' of DMB	This is a real address (Virtual is in X'5C' - X'5F' of DMB)
BCA	X'40' - X'43' of the LCB	
CPB	X'E0' - X'E3' of the CP TCB	
CP TCB	X'DC' - X'DF' of CAT	TCBbObCP
DLRSA (or second LRSA)	R13 in TRSA for task executing code in logical routines X'4' - X'7' in LRSA (in TCB) for task executing code in physical routines.	
End address ot POWER/VS partition	Subtract 1 from contents of X'20' - X'23' of CAT	
Execution account record	See AARA	
First fixed page	X'14' - X'17' of SCB	
INIT/TERM TCB	X'E4' - X'E7' of CAT	TCBbIbIT
Last permanent page	X'18' - X'1B' of SCB	
Line account record	First 56 bytes of LCB	
List account record	See AARA	
LCB	X'21C' - X'21F' of CAT	
LDA	X'88' - X'8B' of a TCB (not a CP TCB or line manager TCB)	This is a real address (Virtual is in X'C' - X'F' of LWS)

Figure 6.1. Part 2: Locating and Identifying Control Blocks, Tables and Areas in the Fixable Area (Part 1 of 3)

Abbreviated * MNEMONIC OF Table or Area	Pointer to or Address of the Table/Area	Identification in the translated dump output
LMF (Line Manager Fields)	X'80' - X'F7' of a TCB for an RJE Line Manager task.	
LRSA	X'C0' - X'E7' of a TCB for an RDR, LST, PUN, or XP task	This LRSA is always the first, and is used to save registers 14-9 used by the physical routines
MCB for Q file	X'8C' - X'8F' of CAT	MCB QFILE 1 and physical address of device
MCB data file 1	X'90' - X'93' of CAT	MCB DFILE 2 and physical address of device
MCB data file 2	X'94' - X'97' of CAT	MCB DFILE 3
MCB data file 3	X'98' - X'9B' of CAT	MCB DFILE 4
MCB data file 4	X'9C' - X'9F' of CAT	MCB DFILE 5
MCB data file 5	X'A0' - X'A3' of CAT	MCB DFILE 6
MCB private SSL	X'A4' - X'A7' of CAT	
MCB system SSL	X'A8' - X'AB' of CAT	
MRA (starts at X'100' of DMB	X'E8' - X'EB' of DMB	This is a real address. (Virtual is in X'EC' - X'EF' of DMB)
MSCB	X'5C - X'5F' of CAT	MSCB
PDB (partition CB)	X'A0' - X'A3' of the partition COMREG and R6 in TRSA in a TCB.	PDB
PCB	First 24 bytes of each page in fixable storage	
PDA	X'04' - X'07' of a PWS	This is a real address. (Virtual is in X'08' - X'08' of a PWS.) Note: For an RJE task the address of the PDA is X'61' - X'63' of the BCA.
PWS	R8 in a TCB for a task in a physical routine	
QRA (Queue Record Area)	X'110' - X'113' of a TCB (Not a CP TCB or a Line or SNA Manager TCB)	This is a real address (Virtual is in X'AC' - X'AF' of a TCB)
QR identity	X'2A' of a queue record	R or L, or P
Size of data buffer	X'20C' - X'20F' of CAT	
Size of data block	X'210' - X'213' of CAT	
SKAD of account file	X'180' - X'187' of DMB	
SKAD of MR	X'E0' - X'E7' of DMB	
SKAD of next QR	X'E8' - X'EF' of a DMB	See Table 1 and 2 in Figure 5.12
SKAD of previous QR	X'260' - X'267' of a DMB	See Table 1 and 2 in Figure 5.12
SKAD of current QR	X'110' - X'113' of TCB and X'F0' - X'F7' of DMB	
SLI work space (SLW)	X'30' - X'33' of PDB	
SNCB	X'7C' - X'7F' of CAT	SNCB
Start of fixable area	X'18' - X'1B' of CAT	Each page in fixable area starts with a PDB.
Start of pageable area	X'1C' - X'1F' of CAT	
Start of TSL	X'14' - X'17' of WCB	
Tape control block (data)	X'F0' - X'F3' of TCB *	TBB
Tape control block (Q file)	X'110' - X'113' of TCB * * Not CP TCB or Line or SNA Manager TCB	

Figure 6.1. Part 2: Locating and Identifying Control Blocks, Tables and Areas in the Fixable Area (Part 2 of 3)

Abbreviated * MNEMONIC OF Table or Area	Pointer to or Address of the Table/Area	Identification in the translated dump output
TCB of most recently attached auxiliary command processor	X'F8' - X'FB' of CAT	If no auxiliary command processor exists, this location contains address of TCB for the permanent command processor (CP TCB).
TCB for most recently attached execution processor task	X'104' - X'107' of CAT	
TCB of highest priority in TSL	X'14' - X'17' of WCB	
TCB for line or SNA manager (if present)	X'F0' - X'F3' of CAT	If line manager not present, this location contains address of WCB.
TCB for most recently attached Writer task	X'100' - X'103' of CAT	
TCB for most recently attached RJE task	X'FC' - X'FF' of CAT	
TCB for most recently attached reader task	X'108 - X'10B of CAT	
TCB for line manager	X'E8' - X'EB' of CAT	
TRSA	X'40' - X'77' in any TCB	
TSL	X'14' - X'17' of WCB; then X'14' - X'17' of each TCB until return to WCB	Recognize each TCB by its storage descriptor. (See debugging hint number 11.)
* Refer to "List of Abbreviations" in Appendix A.		

Figure 6.1. Part 2: Locating and Identifying Control Blocks, Tables and Areas in the Fixable Area (Part 3 of 3)

In the DOS/VS GETVIS Area

Abbreviated * MNEMONIC of Table or Area	Pointer to or Address of the Table/Area	Identification in the Translated Dump Output
COCB	X'8C' - X'8F' of SNCB	COCB
LRCB	X'84' - X'87' of SNCB	LRCB
LUCB	X'29' - X'2B' of SUCB	LUCB
	X'39' - X'3B' of SUCB	
	X'49' - X'4B' of SUCB	
	X'59' - X'5B' of SUCB	
	X'69' - X'6B' of SUCB	
	X'79' - X'7B' of SUCB	
RMCB	X'20' - X'23' of SNCB	RMCB
SUCB	X'14' - X'17' of SNCB	SUCB
	X'10' - X'13' of TCB	
WACB	X'14' - X'17' of TCB	WACB
* Refer to "List of Abbreviations" in Appendix A.		

Figure 6.1. Part 3: Locating and Identifying Control Blocks, Tables and Areas in the DOS/VS GETVIS Area

6. IDENTIFYING THE START OF A CSECT

Each control section within the POWER/VS code is identified by a 16-byte control section descriptor in the following format.

Bytes	Name	Description
00-07		Control section name This 8-byte field contains the alphameric name assigned to the control section. Since control section names are of four characters only, bytes 4-7 will contain character blanks.
08-		Supported changes This 6-byte field identifies SDD changes and will in general contain the version number and modification level of the control section in the form 'vnMnbb'. The initial contents of the field will be 'V5M0bb'.
-		Local changes [1] This field will be used to reflect changes made by other than SDD personnel. Its initial contents will be character blanks.

[1] Check this field for non-SDD changes.

7. ESTABLISHING THE LEVEL OF A CSECT

The level of a routine (Physical, Logical, Execution, Function, Service, see Figure 2.1 in Section 2) can be established by the first two characters of its CSECT name identified in a dump. For example, if the contents of register 12 in a TCB points to an address within CSECT name AQCS, the calling routine (AQCS) is at FUNCTION level.

8. DETERMINING THE ACTIVE ROUTINE AND ANALYZING REGISTER SAVE AREAS

It is important to know the routine in which a task is executing code in order to be able to analyze the meaning of the contents of the registers saved.

The contents of R12 in the TRSA in a TCB belonging to a task that is not in R state will address the instruction that will be next executed when the task is given control. The routine or CSECT in which this instruction is located can be identified in a dump by means of the storage descriptor.

Figure 5.14 shows the relationship between the LRSA in TCB and the DLRSA or second LRSA, which depends on the calling sequence of POWER/VS routines. (See Figure 2.1 in Section 2.)

Note: Register conventions are described in Section 3.

9. ANALYZING ECBS (EVENT CONTROL BLOCKS)

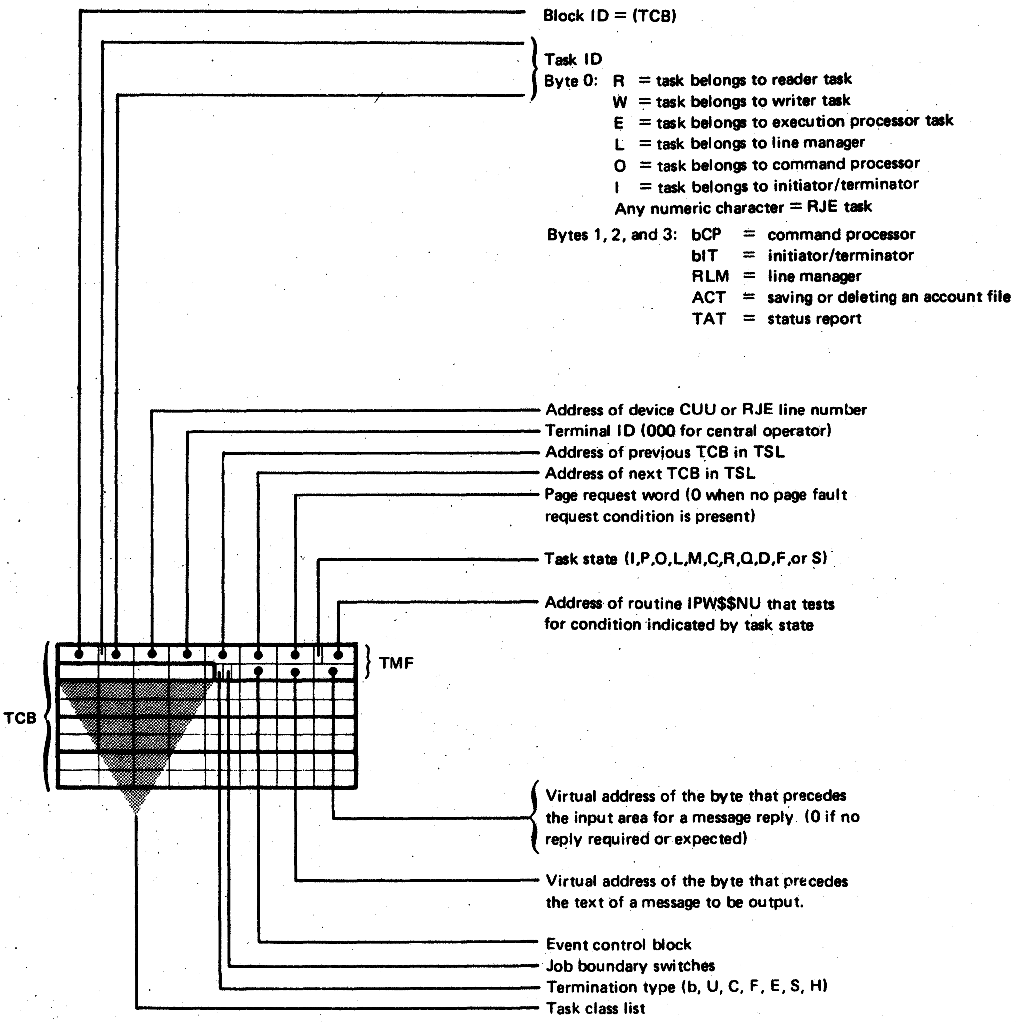
Several control blocks are equipped with ECBs, the condition of which (posted [bit 16 on (1)] or unposted [bit 16 off (0)]) may be important to problem analysis. See Appendix D.

10. USING BUFFER CONTROL WORDS

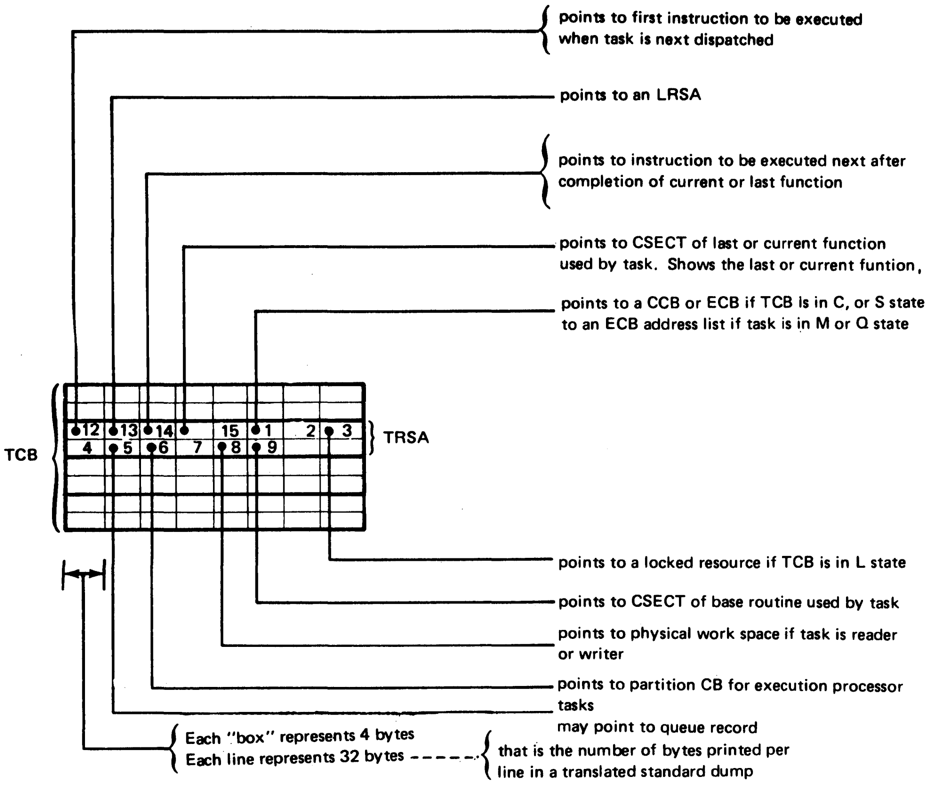
The four bytes immediately in front of any area contain the address of the task control block of the task which reserved the area. (See also Figure 5.20.)

11. ANALYZING TCBS (QUICK REFERENCE ONLY)

General meaning of the task management fields (TMF). (See Figure 5.10 for details.)


Block ID = (TCB)
Task ID
Byte 0: R = task belongs to reader task
W = task belongs to writer task
E = task belongs to execution processor task
L = task belongs to line manager
O = task belongs to command processor
I = task belongs to initiator/terminator
Any numeric character = RJE task
Bytes 1, 2, and 3: bCP = command processor
bIT = initiator/terminator
RLM = line manager
ACT = saving or deleting an account file
TAT = status report
Address of device CUU or RJE line number
Terminal ID (000 for central operator)
Address of previous TCB in TSL
Address of next TCB in TSL
Page request word (0 when no page fault request condition is present)
Task state (I,P,O,L,M,C,R,Q,D,F,or S)
Address of routine IPW$$NU that tests for condition indicated by task state
TMF
TCB
Virtual address of the byte that precedes the input area for a message reply (0 if no reply required or expected)
Virtual address of the byte that precedes the text of a message to be output.
Event control block
Job boundary switches
Termination type (b, U, C, F, E, S, H)
Task class list

General meaning of fields in the TRSA. (See Figure 5.11 for details.)



points to first instruction to be executed when task is next dispatched

points to an LRSA

points to instruction to be executed next after completion of current or last function

points to CSECT of last or current function used by task. Shows the last or current funtion,

points to a CCB or ECB if TCB Is in C, or S state to an ECB address list if task is in M or Q state

points to a locked resource if TCB is in L state

points to CSECT of base routine used by task

points to physical work space if task is reader or writer

points to partition CB for execution processor tasks

may point to queue record

Each "box" represents 4 bytes
Each line represents 32 bytes ------ that is the number of bytes printed per line in a translated standard dump

12. ESTABLISHING QUEUE RECORDS IN QUEUE SETS IN CLASS CHAINS

The contents of X'89' in a queue record can be X'00' or X'01'. X'01' indicates that the queue record is the first in the queue set. X'00' indicates that the queue record is not the first in the queue set.

This determines the meaning of the fields QRNS, QRQP, and QRQN as follows:

QRFS = X'00' (This queue record is not first in queue set)

Displacement in Queue Record	Contents is Zero	Contents not Zero
X'90'-X'97'	This queue record is the last in this queue set.	It is seek address of next queue record in this queue set.
X'98'-X'9F'	Cannot be zero.	It is seek address of the first queue record in this queue set.
X'A0'-X'A7'	Must be zero.	Must be zero.

QRFS = X'01' (This queue record is first in queue set)

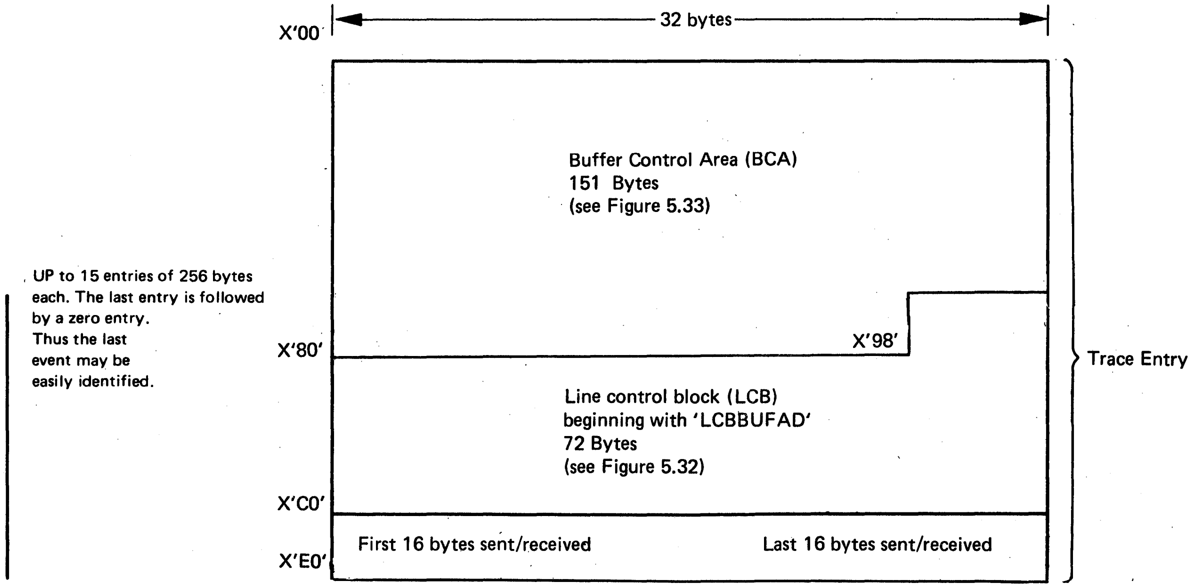
Displacement in Queue Record	Contents is Zero	Contents not Zero
X'90'-X'97'	This queue record is last in same queue set.	It is the seek address of the next queue record in this queue set.
X'98'-X'9F'	This queue record is the first in the first queue set in class chain.	It is the seek address of the first queue record of the previous queue set.
X'A0'-X'A7'	This queue record is the first queue record of the last queue set in this class chain.	It is the seek address of the next queue set in this class chain.

See also Figure 5.48, Part 9.

13. RJE,BSC I/O TRACE

An I/O trace for an RJE,BSC line after SIGNON can be initiated by specifying YES to TRACE= in the PRMT macro.

Entries are made in a wraparound buffer in the phase IPW$$TM. The following information is recorded at every I/O interrupt from this terminal.



32 bytes
X'00'
Buffer Control Area (BCA)
151 Bytes
(see Figure 5.33)
UP to 15 entries of 256 bytes each. The last entry is followed by a zero entry. Thus the last event may be easily identified.
X'80'
X'98'
Trace Entry
Line control block (LCB)
beginning with 'LCBBUFAD'
72 Bytes
(see Figure 5.32)
X'C0'
X'E0'
First 16 bytes sent/received
Last 16 bytes sent/received

The trace is to be used when RJE,BSC line errors occur or incorrect output is encountered which can be caused by the I/O operation.

A dynamic, continuous trace of BSC RJE activity may be taken using the following steps. The user should acquaint himself with SDAIDS before referring to these steps.

- Obtain a dump of the POWER/VS partition.
- Locate the POWER/VS line manager in the dump ("LMCS VxMy").
- Within the line manager locate the constant "SDAID 'IF' --->".
- The instruction immediately following this constant is fetched when the trace area is full and wrap-around will occur.
- Use SDAIDS instruction fetch at this location with OUTCLASS=PDUMP,X'AAAAAA',X'BBBBBB', where "AAAAAA" is the virtual address of the start of trace area (beginning on the second DOS/VS page from the above instruction), and "BBBBBB" equals "AAAAAA" + X'1000'.

This will provide a continuous trace of activity on any or all lines generated with TRACE=YES in the PRMT macro.

Note: Trace activity begins with SIGNON.

14. POWER/VS FILE DUMP PROGRAM

This program enables any of the POWER/VS files (account, queue, data) to be dumped on a line printer assigned to SYSLST. An option is also provided to enable queue records and their associated track groups belonging to specific jobs to be dumped.

How to Execute

The program is requested by JCL commands entered either via SYSLOG or SYSIN, where SYSIN is assigned to a card reader. Before requesting ensure relevant assignments are made for the file to be dumped.

Example Job Stream

```
// JOB name
// ASSGN (SYS000 for Account file)
         (SYS001 for Queue file)
         (SYS002-6 for Data files)
// EXEC IPW$$DD
```

When the program is loaded successfully, the following message will be issued to SYSLOG:

```
   DUMP FUNCTION=
```

At this point one of the following options can be entered via SYSLOG:

- A (to specify the Account file)
- Q (to specify the Queue file)
- D (to specify the Data file) [1]
- jobname[,jobnumber] [,queue] [2]
- EOJ (to enable cancelation of the program or selection of a new option)

[1] The complete data file will be dumped.

[2] This enables (a) queue record(s) belonging to a specific job in the RDR, LST, or PUN queue plus its associated track group(s) to be dumped. Job name may be 8 characters, job number may be 6 characters. For the 'queue' option one of the following three entries can be specified:

L, for LST queue (default)
P, for PUN queue
R, for RDR queue.

After the dump is completed, the message

```
   DUMP FUNCTION=
```

is issued to SYSLOG again to enable either a new option to be specified or the program to be terminated by the option EOJ.

Format of Output

For every 100 bytes, a block of four lines is printed. Line 1 contains the printable characters in those bytes; line 2 contains the zone-part of each byte; line 3 contains the numeric part of each byte; line 4 contains a scale indicating the position of the bytes in the string.

```
line 1:    CHAR  // JOB POWJOB01                  DATE 08/19/74,
line 2:    ZONE  664DDC4DDEDDCFF44444444444    4444CCEC4FF6FF6FF6
line 3:    NUMR  1101620766162010000000000000    0000413500811917 4B
line 4:            01...5...10...15...20...25.   .85...90...95.....
```

A. Analyzing Information in a Dump of the Data File

Establish correct contents according to data record layout (see Figure 5.16).

B. Analyzing Information in a Dump of the Q File

Establish correct contents and chaining of the 184-byte Q records according to Q record layout (see Figure 5.22).

C. Analyzing Information in a Dump of the Account File

Establish correct contents of the variable length records according to Account record layouts (five types, see Figures 5.26-5.30).

15. ESTABLISHING THE LAST COMMAND ISSUED

The last command issued by the central operator can be seen printed in the translated part of a dump within the permanent command processor control block, recognized by the storage descriptor CPB.

16. AN AID TO ELIMINATE COMPONENTS

It may be useful to have several different generation tables cataloged to CIL with at least a default version as originally supplied to the user. The various versions act as a debugging aid to eliminate components.

17. PROBLEMS RELATED TO VTAM

If a problem occurs where VTAM is involved, it is recommended to consult section "VTAM Serviceability Aids" (VTAM traces) in the DOS/VS VTAM Debugging Guide, GC27-0021.

A System Dump Containing the POWER/VS Partition

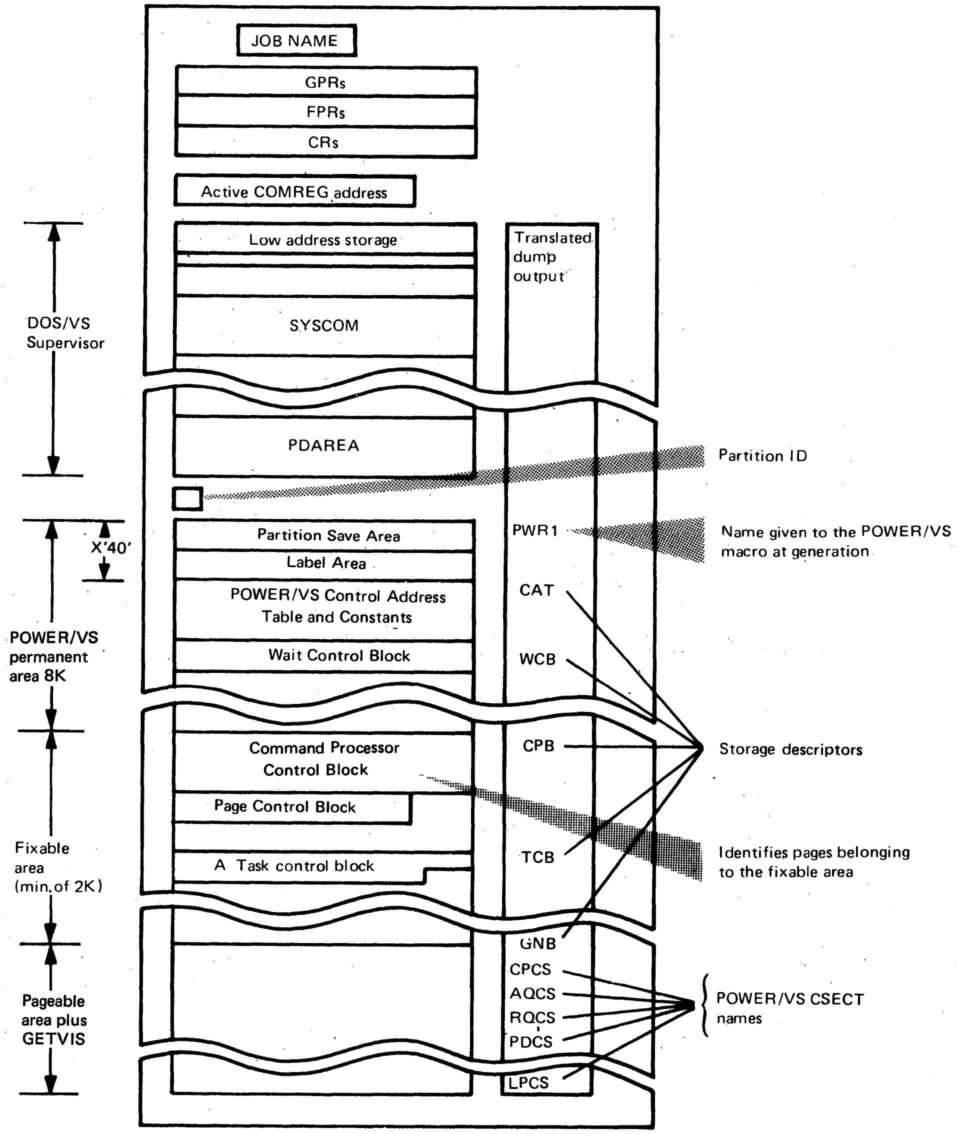


Figure 6.2. Pictorial representation of a system dump containing the POWER/VS partition

For a full description of a system dump refer to DOS/VS Serviceability Aids and Debugging Procedures.

Diagnostic Charts

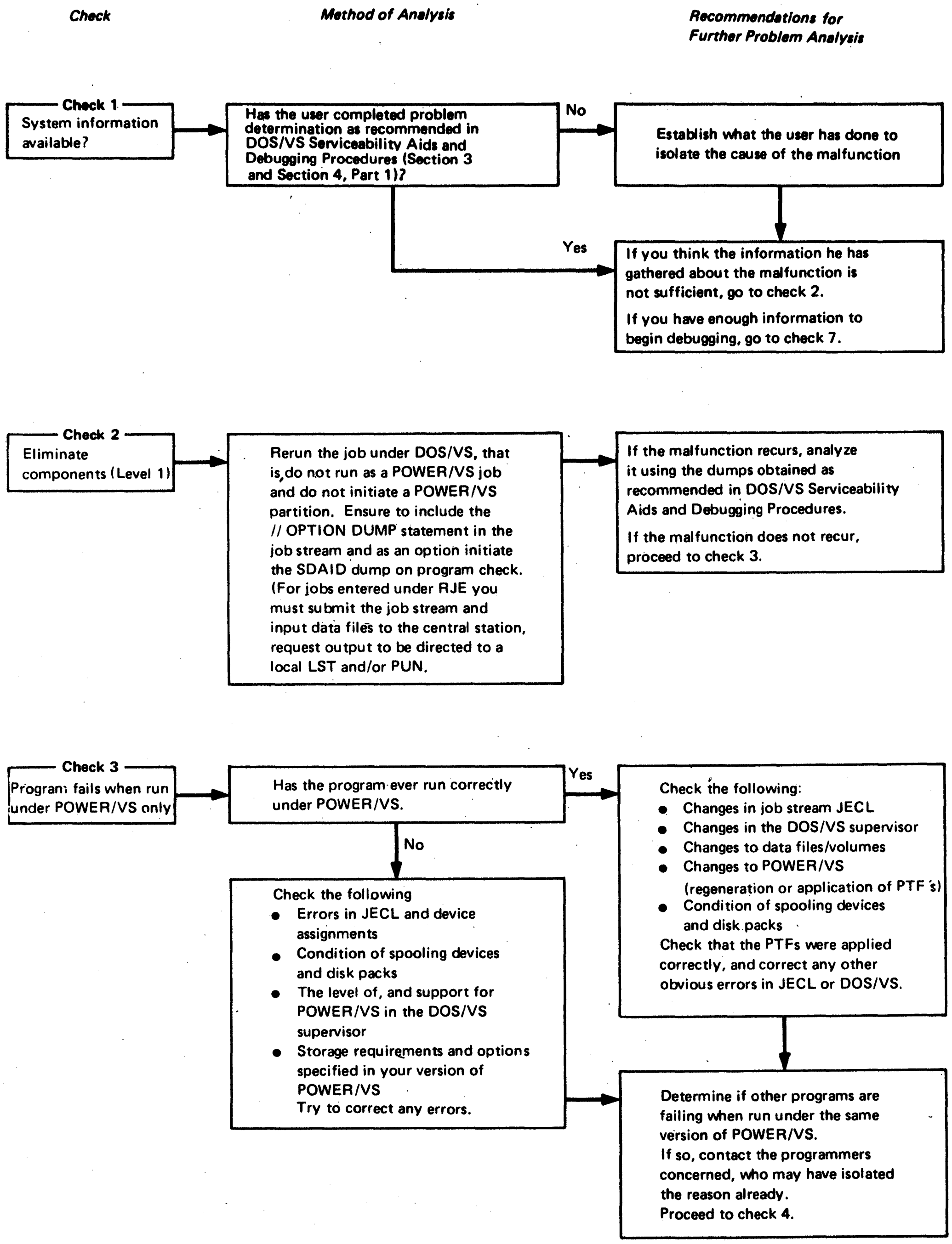
The following charts are designed to guide IBM service personnel.

They describe a method of software error analysis but are not intended to be foolproof.

However, these charts indicate simple checks on system environment and enable information to be gathered from a dump of the POWER/VS partition for given malfunctions.

Contents

Chart 01. Initial Environmental Checks (Part 1 of 3)



Check 1

System information available?

Has the user completed problem determination as recommended in DOS/VS Serviceability Aids and Debugging Procedures (Section 3 and Section 4, Part 1)?

Establish what the user has done to isolate the cause of the malfunction

If you think the information he has gathered about the malfunction is not sufficient, go to check 2.

If you have enough information to begin debugging, go to check 7.

Check 2

Eliminate components (Level 1)

Rerun the job under DOS/VS, that is, do not run as a POWER/VS job and do not initiate a POWER/VS partition. Ensure to include the // OPTION DUMP statement in the job stream and as an option initiate the SDAID dump on program check. (For jobs entered under RJE you must submit the job stream and input data files to the central station, request output to be directed to a local LST and/or PUN.

If the malfunction recurs, analyze it using the dumps obtained as recommended in DOS/VS Serviceability Aids and Debugging Procedures.

If the malfunction does not recur, proceed to check 3.

Check 3

Program fails when run under POWER/VS only

Has the program ever run correctly under POWER/VS.

Check the following:

- Changes in job stream JECL
- Changes in the DOS/VS supervisor
- Changes to data files/volumes
- Changes to POWER/VS (regeneration or application of PTF's)
- Condition of spooling devices and disk packs

Check that the PTFs were applied correctly, and correct any other obvious errors in JECL or DOS/VS.

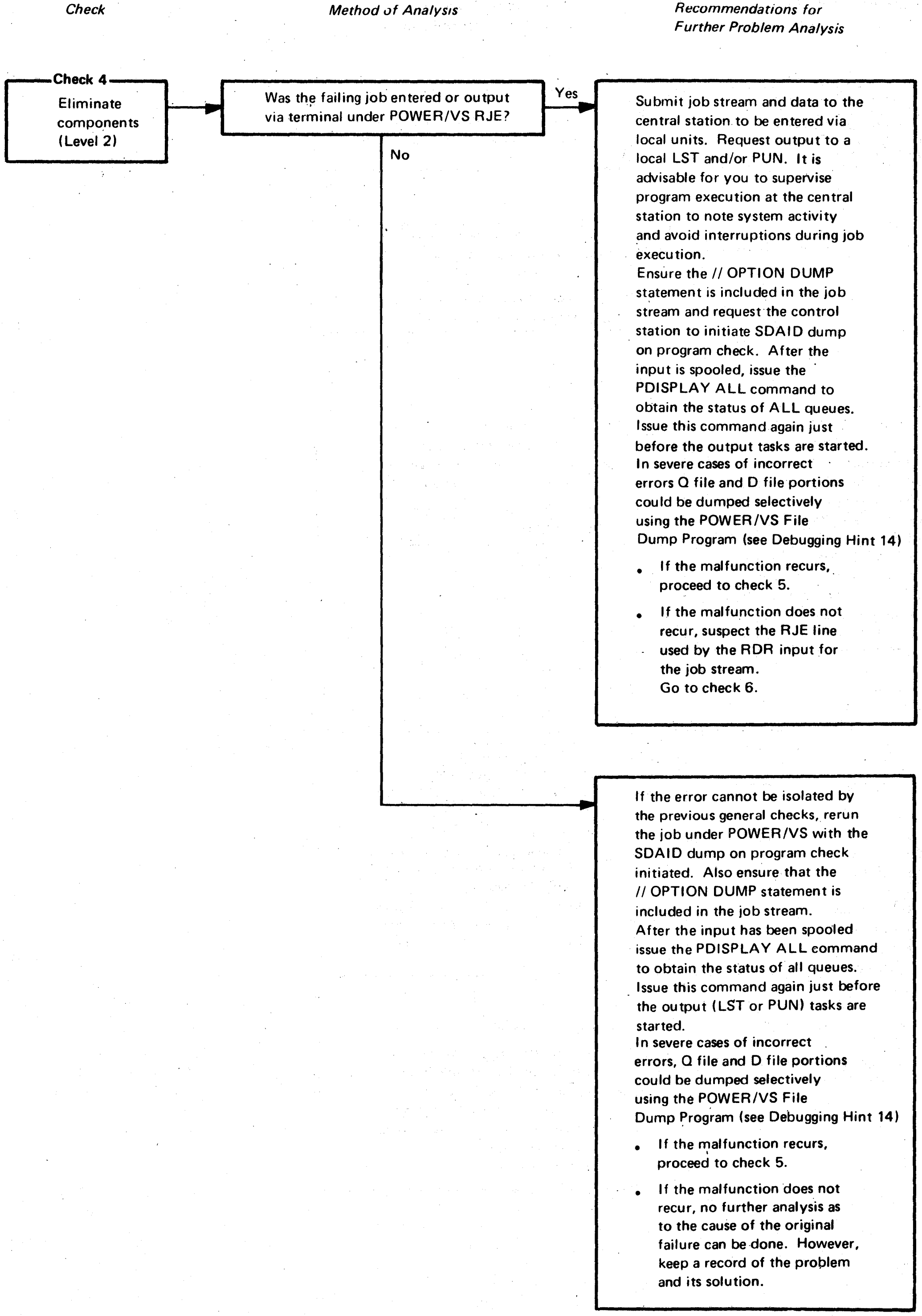
Check the following

- Errors in JECL and device assignments
- Condition of spooling devices and disk packs
- The level of, and support for POWER/VS in the DOS/VS supervisor
- Storage requirements and options specified in your version of POWER/VS

Try to correct any errors.

Determine if other programs are failing when run under the same version of POWER/VS.
If so, contact the programmers concerned, who may have isolated the reason already.
Proceed to check 4.

Chart 01. Initial Environmental Checks (Part 2 of 3)



Check 4

Eliminate components (Level 2)

Was the failing job entered or output via terminal under POWER/VS RJE?

Submit job stream and data to the central station to be entered via local units. Request output to a local LST and/or PUN. It is advisable for you to supervise program execution at the central station to note system activity and avoid interruptions during job execution.
Ensure the // OPTION DUMP statement is included in the job stream and request the control station to initiate SDAID dump on program check. After the input is spooled, issue the PDISPLAY ALL command to obtain the status of ALL queues. Issue this command again just before the output tasks are started.
In severe cases of incorrect errors Q file and D file portions could be dumped selectively using the POWER/VS File Dump Program (see Debugging Hint 14)

- If the malfunction recurs, proceed to check 5.
- If the malfunction does not recur, suspect the RJE line used by the RDR input for the job stream. Go to check 6.

If the error cannot be isolated by the previous general checks, rerun the job under POWER/VS with the SDAID dump on program check initiated. Also ensure that the // OPTION DUMP statement is included in the job stream.
After the input has been spooled issue the PDISPLAY ALL command to obtain the status of all queues. Issue this command again just before the output (LST or PUN) tasks are started.
In severe cases of incorrect errors, Q file and D file portions could be dumped selectively using the POWER/VS File Dump Program (see Debugging Hint 14)

- If the malfunction recurs, proceed to check 5.
- If the malfunction does not recur, no further analysis as to the cause of the original failure can be done. However, keep a record of the problem and its solution.

Chart 01. Initial Environmental Checks (Part 3 of 3)

Check	*Method of Analysis*	*Recommendation for further problem analysis*
Check 5 POWER/VS SCP error	You may suspect the POWER/VS SCP of causing the error.	Go to check 7
Check 6 RJE line or processing in error	You may suspect the RJE line used for job input or the processing of RJE under POWER/VS.	To verify the line, re-run the job using a different RJE line(s) and where possible, different terminal devices on the line. If the malfunction recurs suspect POWER/VS RJE processing Proceed to check 7
Check 7 Catagorize the type of malfunction	WAIT state	Turn to chart 2
	LOOP	Turn to chart 3
	Incorrect output	Turn to chart 4
	Job/program canceled	Turn to chart 5

Chart 02. Wait State, Part 1 of 9

Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 1 Check the information available	Use the list below to check the minimum amount of system information required: ● SYSLOG log sheet ● Note of PSW and low address storage bytes X'00' - X'17' ● Standalone dump (formatted)* ● Line printer output from job(s) ● User's POWER/VS generation listing	If possible try to obtain any notes made by the user about the job(s) action and system activity prior to the malfunction. A SYSVIS dump may be necessary. Proceed to check 2.
Check 2 Check messages and responses	Examine the SYSLOG log sheet scanning for messages and operator responses. Check responses given with those recommended in DOS/VS messages.	Look for unusual messages (those that do not obviously apply to jobs running may give you a clue). Any incorrect responses will require a user rerun of the job(s). Proceed to check 3.
Check 3 POWER/VS initialized	Check that POWER/VS has been correctly initialized by checking for message 1Q12I	Check DLBL, EXTENT, and LABEL information. If message 1Q12I has not been issued, go to Chart 06; otherwise, proceed to check 4.

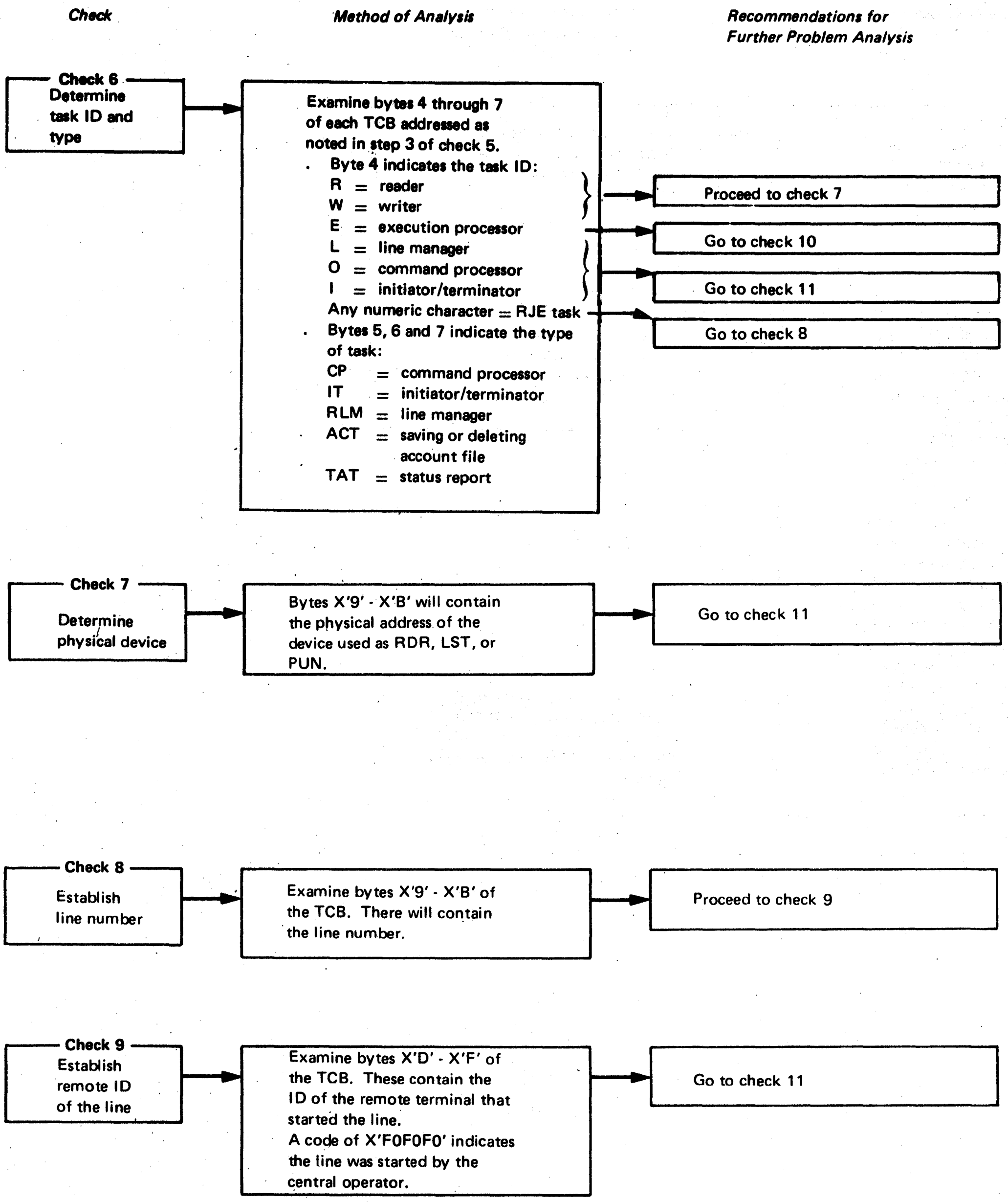
* An example and description of a formatted standalone dump is given in Appendix G of DOS/VS Serviceability Aids and Debugging Procedures.

Chart 02. Wait State, Part 2 of 9

Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 4 Establish tasks started	Use SYSLOG log sheet to determine partitions and tasks started under POWER/VS (this may help in further analysis).	Make a note of the tasks and classes specified, as well as partitions and priorities. Proceed to check 5.
Check 5 Locate all TCBs in the TSL	Locate the POWER/VS wait control block in the standalone dump. (Refer to Table 6.1 in this section, if you do not know how to do this) 1. Make a note of the address contained in bytes X'14' - X'17' in the WCB. (This will be a virtual address. Convert it to real by using the formatted list of page tables printed at the end of the standalone dump)* 2. Locate this address in the dump. This is the first byte of a TCB in the TSL. 3. Make a note of the address contained in bytes X'14' - X'17' in that TCB, and convert it to a real address. 4. Repeat steps 2 and 3 until the address is that of the WCB.	Proceed to check 6 to analyze each TCB whose address was noted in step 3 of this check

* An example and description of a formatted standalone dump is given in Appendix G of DOS/VS Serviceability Aids and Debugging Procedures.

Chart 02. Wait State, Part 3 of 9



Check

Method of Analysis

Recommendations for Further Problem Analysis

Check 6
Determine task ID and type

Examine bytes 4 through 7 of each TCB addressed as noted in step 3 of check 5.

- Byte 4 indicates the task ID:
  R = reader
  W = writer
  E = execution processor
  L = line manager
  O = command processor
  I = initiator/terminator
  Any numeric character = RJE task
- Bytes 5, 6 and 7 indicate the type of task:
  CP = command processor
  IT = initiator/terminator
  RLM = line manager
  ACT = saving or deleting account file
  TAT = status report

Proceed to check 7

Go to check 10

Go to check 11

Go to check 8

Check 7
Determine physical device

Bytes X'9' - X'B' will contain the physical address of the device used as RDR, LST, or PUN.

Go to check 11

Check 8
Establish line number

Examine bytes X'9' - X'B' of the TCB. There will contain the line number.

Proceed to check 9

Check 9
Establish remote ID of the line

Examine bytes X'D' - X'F' of the TCB. These contain the ID of the remote terminal that started the line.
A code of X'F0F0F0' indicates the line was started by the central operator.

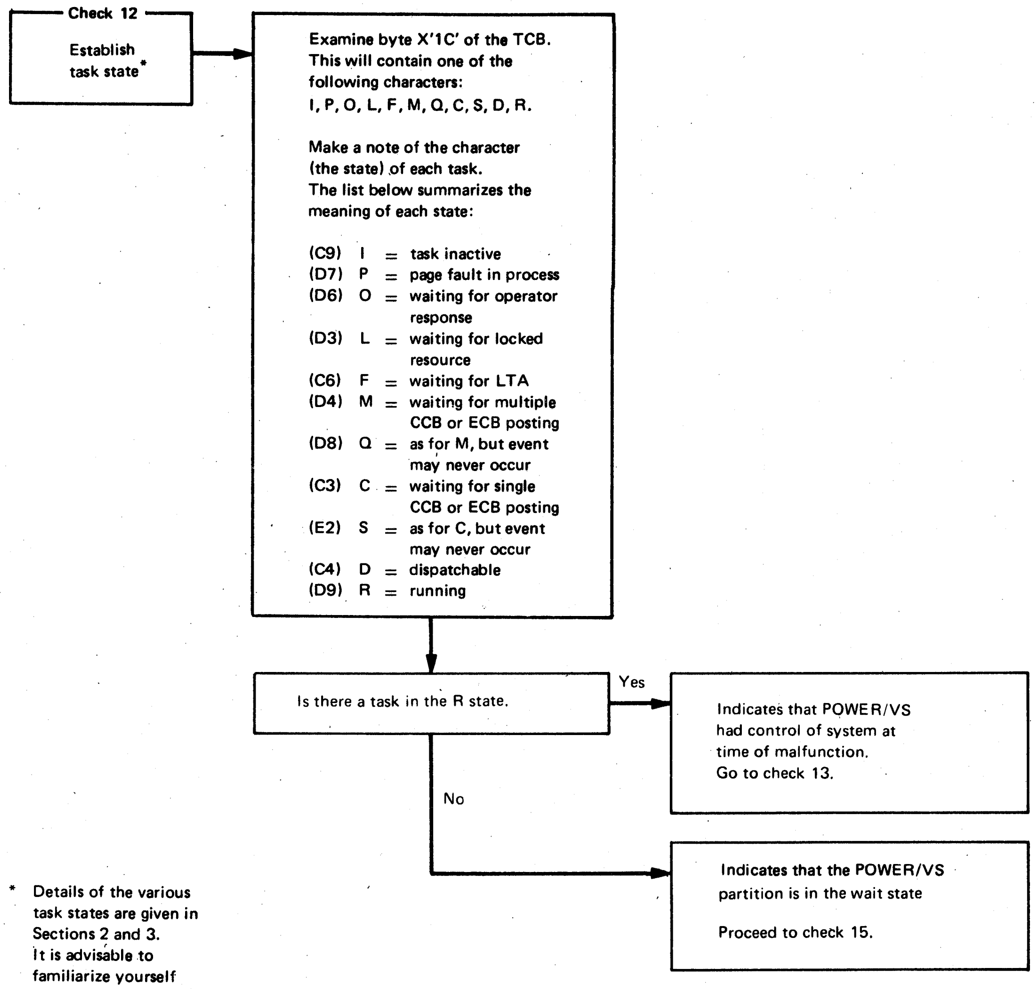
Go to check 11

Chart 02. Wait State, Part 4 of 9

Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 10 Establish partition	The contents of bytes 6 and 7 of the TCB will contain one of the following hex codes: ● C2C7 = BG ● C6F1 = F1 ● C6F2 = F2 ● C6F3 = F3 ● C6F4 = F4	Locate R6 in TRSA of execution task TCB(s). This should address the associated partition control block(s). Proceed to check 10a.
Check 10a Analyze information in the PDB	Examine the RDRdeviceentry, the LSTdeviceentries and the PUNdeviceentries. (see Figure 5.23) From these obtain the addresses of the TCBs associated and the CCB addresses. Also note the partition COMREG address, the PIB address, device type codes (in the spool entries), and requestor IDs.	Check CCB, TCB, PIB and COMREG addresses for validity and examine device type codes and requestor IDs. Make a note of any irregularities (this may be the cause of the error) and proceed to check 11.
Check 11 Summary of reason for the wait.	The wait state implies that a condition has occurred that has either not been satisfied or not been detected. When a POWER/VS event occurs it must be reported by: 1. The posting of the task ECB in the TCB, or the posting of any other ECB or CCB or the posting of an LK in a CB. 2. The posting of the POWER/VS MECB in the CAT. 3. The posting of the PIB entry belonging to the POWER/VS partition. Any code that is handling a condition must account for all three postings, or prove them not to be required.	The following checks guide you through a method of analysis that will help you to locate the reason for the loss of one of these postings. Proceed to check 12.

Chart 02. Wait State, Part 5 of 9

Check | *Method of Analysis* | *Recommendations for Further Problem Analysis*



Check 12

Establish task state*

Examine byte X'1C' of the TCB. This will contain one of the following characters: I, P, O, L, F, M, Q, C, S, D, R.

Make a note of the character (the state) of each task. The list below summarizes the meaning of each state:

(C9)	I	=	task inactive
(D7)	P	=	page fault in process
(D6)	O	=	waiting for operator response
(D3)	L	=	waiting for locked resource
(C6)	F	=	waiting for LTA
(D4)	M	=	waiting for multiple CCB or ECB posting
(D8)	Q	=	as for M, but event may never occur
(C3)	C	=	waiting for single CCB or ECB posting
(E2)	S	=	as for C, but event may never occur
(C4)	D	=	dispatchable
(D9)	R	=	running

Is there a task in the R state.

Yes: Indicates that POWER/VS had control of system at time of malfunction. Go to check 13.

No: Indicates that the POWER/VS partition is in the wait state

Proceed to check 15.

* Details of the various task states are given in Sections 2 and 3. It is advisable to familiarize yourself with 'task state processing'.

Chart 02. Wait State, Part 6 of 9

Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 13 Task in R state	Examine the contents of the registers in the TRSA in the TCB.	**The values in the registers record their contents when the task was given control. Register conventions are described in Section 3. Examine the register saved in the POWER/VS partition save area and the return PSW.*** **The address in this PSW will point to an instruction in the routine which will get control next.** **Proceed to check 14.**
Check 14 Analyze task in R state	Examine the PIB entry for the POWER/VS partition. Byte 0 should be X'83'. Examine partition save area of POWER/VS partition; R1 may point to CCB or ECB.	This is probably a DOS/VS error not caused by POWER/VS. However, check that the POWER/VS code has not been changed by the user, since this error indicates that an SVC 7 was issued by the POWER/VS task rather than by task management. If the cause of the wait state does not seem to be this task proceed to check 15.

* Refer to DOS/VS Serviceability Aids and Debugging Procedures for details about the partition save area.

Chart 02. Wait State, Part 7 of 9

Check

Method of Analysis

Recommendations for Further Problem Analysis

Check 15

Task(s) not in 'R' state.

This indicates that the POWER/VS partition is in the wait state. Locate the instruction before the instruction address contained in the return PSW in the partition save area. If this instruction is 0A07 it means that an SVC 7 has been issued to DOS/VS. Examine byte 0 of each PIB entry to confirm the wait state. The reason for this must be isolated by first analyzing information contained in each TCB.

Examine other fields in the TMS in each TCB and the registers in the TRSA in each TCB.

. R12 points to the instruction which would have been executed after the IPW$WFx macro was handled. Cross check this with the contents in R9 and/or 15 which will point to the start of that CSECT. From the SYSVIS dump you can determine the routine or the CSECT of the caller. (see debugging hint number 8) (If no SYSVIS dump is available, the address in R12 will have to be converted to real (see Appendix E) to enable the standalone dump to be used.)

. R13 may point to an LRSA. Depending on the task, the registers saved may give a clue to the error.

. R1 points to a CCB or ECB if the task was in 'C' or 'S' state, or to a list of several addresses of CCBs or ECBs if the task was in 'M' or 'Q' state.
Go to check 16.

. R3 points to a CB using a resource if the task was in 'L' state.
Go to check 17.

. Examine R13 in the TRSA. If it addresses the LRSA in the TCB and there is a second LRSA outside the TCB, the task was executing code in a physical routine. If it addresses an LRSA outside the TCB, the task was executing code in a logical routine. This is not true for an execution processor task which has only one LRSA in the TCB. (Refer to Figure 5.13)

. If the task was executing code in a physical routine (determined from step 1), a field (X'80' - X'FB') in the TCB may have been initiated (depending on how much of the physical routine had been executed prior to the wait), which will enable a check to be made on addresses for the QR and LDA and data records. (See Figure 5.17)

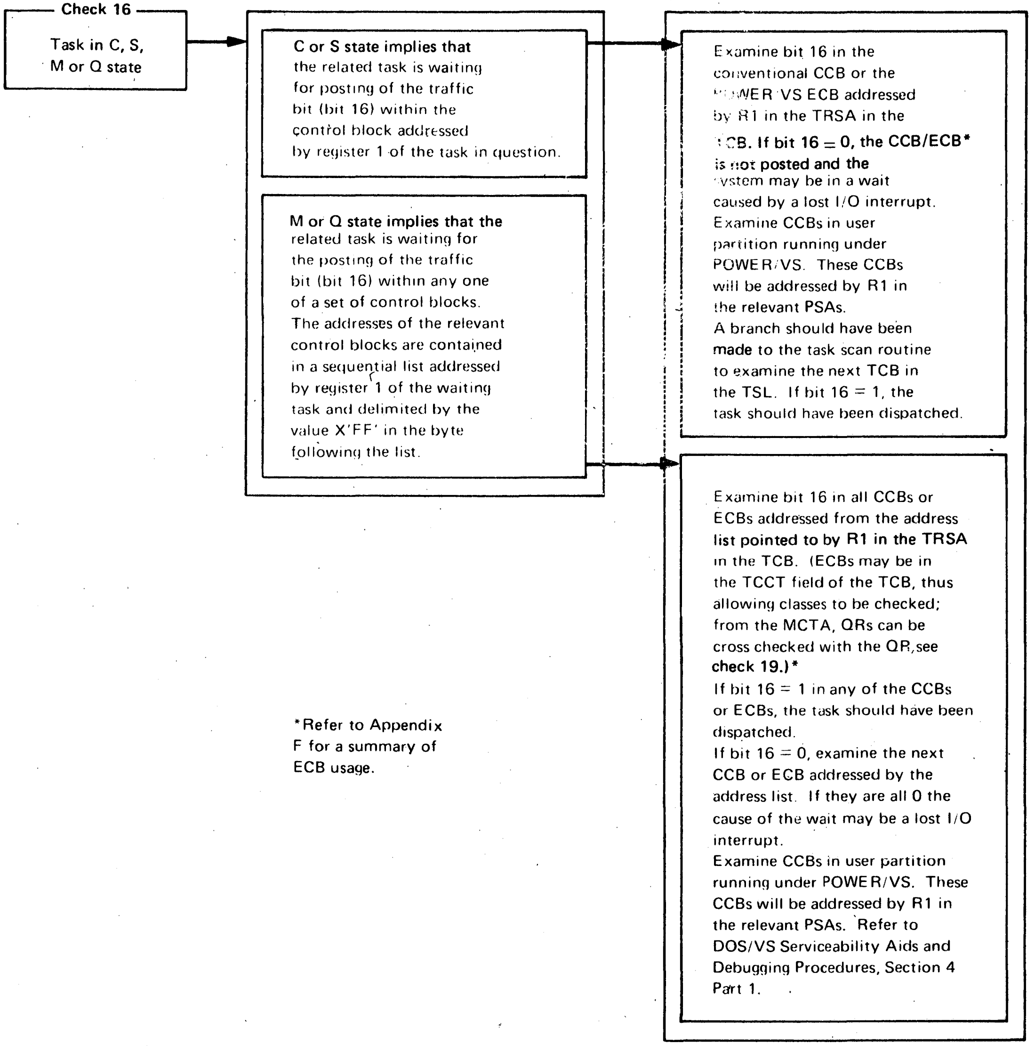
. For tasks in 'I', 'F', 'P', 'O' or 'D' state, go to check 18.

Chart 02. Wait State, Part 8 of 9

Check

Method of Analysis

Recommendations for Further Problem Analysis



Check 16

Task in C, S, M or Q state

C or S state implies that the related task is waiting for posting of the traffic bit (bit 16) within the control block addressed by register 1 of the task in question.

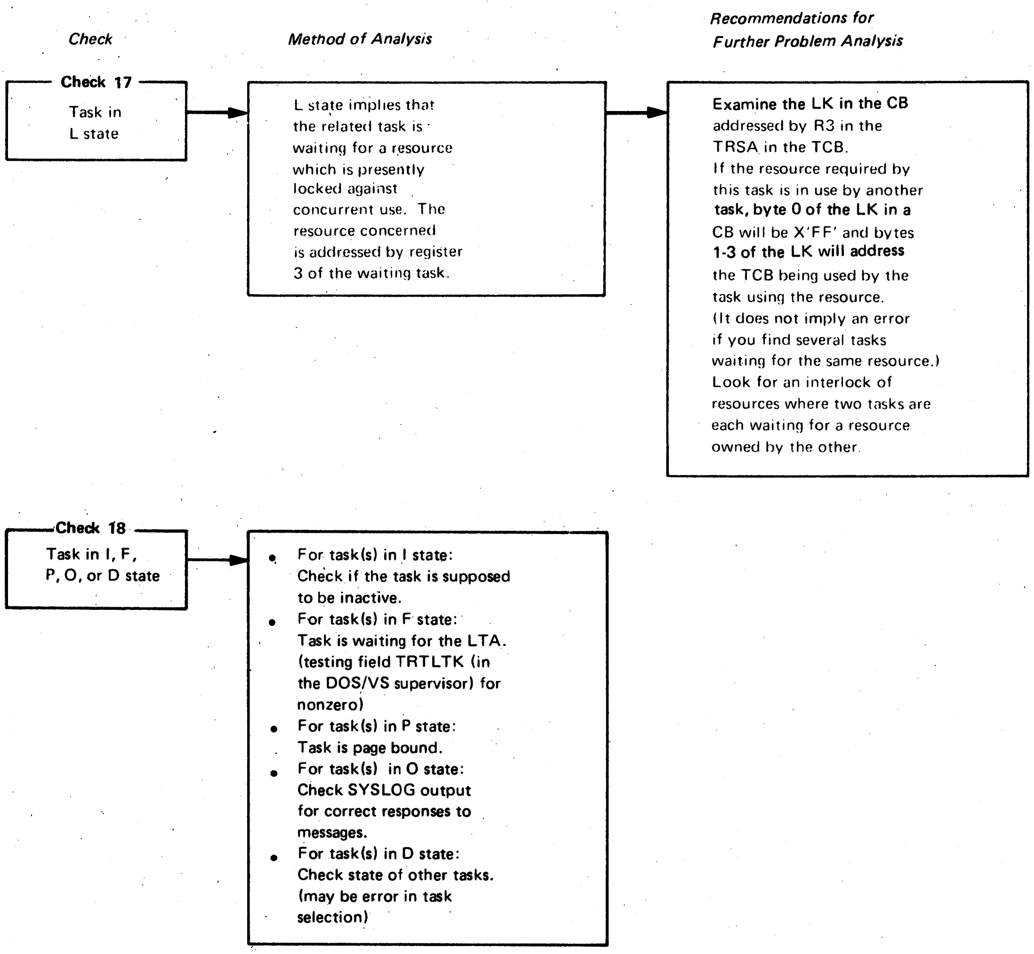
M or Q state implies that the related task is waiting for the posting of the traffic bit (bit 16) within any one of a set of control blocks. The addresses of the relevant control blocks are contained in a sequential list addressed by register 1 of the waiting task and delimited by the value X'FF' in the byte following the list.

Examine bit 16 in the conventional CCB or the POWER VS ECB addressed by R1 in the TRSA in the TCB. **If bit 16 = 0, the CCB/ECB* is not posted and the** system may be in a wait caused by a lost I/O interrupt. Examine CCBs in user partition running under POWER/VS. These CCBs will be addressed by R1 in the relevant PSAs. A branch should have been made to the task scan routine to examine the next TCB in the TSL. If bit 16 = 1, the task should have been dispatched.

Examine bit 16 in all CCBs or ECBs addressed from the address **list pointed to by R1 in the TRSA** in the TCB. (ECBs may be in the TCCT field of the TCB, thus allowing classes to be checked; from the MCTA, QRs can be cross checked with the QR,see **check 19.)*** If bit 16 = 1 in any of the CCBs or ECBs, the task should have been dispatched. If bit 16 = 0, examine the next CCB or ECB addressed by the address list. If they are all 0 the cause of the wait may be a lost I/O interrupt. Examine CCBs in user partition running under POWER/VS. These CCBs will be addressed by R1 in the relevant PSAs. Refer to DOS/VS Serviceability Aids and Debugging Procedures, Section 4 Part 1.

*Refer to Appendix F for a summary of ECB usage.

Chart 02. Wait State, Part 9 of 9



Check

Method of Analysis

Recommendations for Further Problem Analysis

Check 17

Task in L state

L state implies that the related task is waiting for a resource which is presently locked against concurrent use. The resource concerned is addressed by register 3 of the waiting task.

Examine the LK in the CB addressed by R3 in the TRSA in the TCB. If the resource required by this task is in use by another task, byte 0 of the LK in a CB will be X'FF' and bytes 1-3 of the LK will address the TCB being used by the task using the resource. (It does not imply an error if you find several tasks waiting for the same resource.) Look for an interlock of resources where two tasks are each waiting for a resource owned by the other.

Check 18

Task in I, F, P, O, or D state

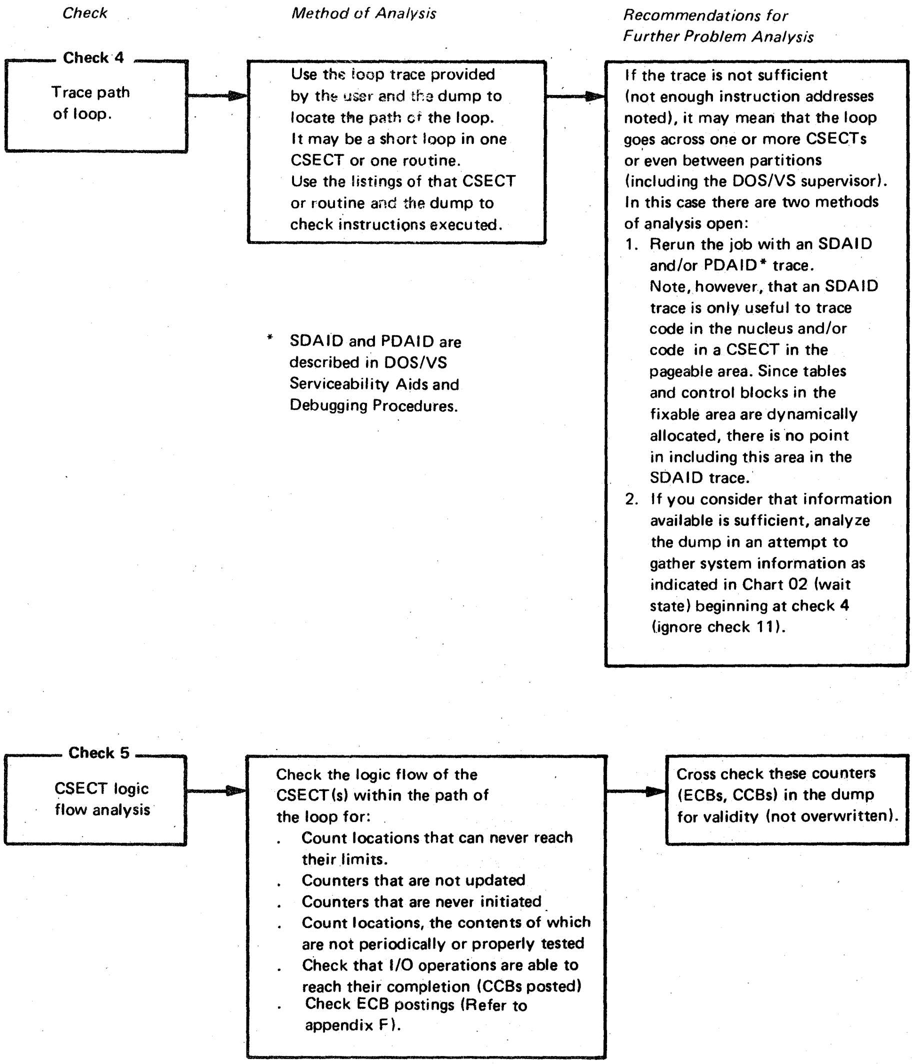
- For task(s) in I state: Check if the task is supposed to be inactive.
- For task(s) in F state: Task is waiting for the LTA. (testing field TRTLTK (in the DOS/VS supervisor) for nonzero)
- For task(s) in P state: Task is page bound.
- For task(s) in O state: Check SYSLOG output for correct responses to messages.
- For task(s) in D state: Check state of other tasks. (may be error in task selection)

Chart 03. LOOP, Part 1 of 2

Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 1 Check the information available	Use the list below to check the minimum amount of system information required: ● SYSLOG log sheet ● Note of PSW and low address storage bytes X'00' - X'17' ● Standalone dump (formatted)* ● Line printer output from job(s) ● User's POWER/VS generation listing ● Trace of the loop.	If possible try to obtain any notes made by the user about the job(s) action and system activity prior to the malfunction. A SYSVIS dump may be necessary. Proceed to check 2.
Check 2 Check messages and responses	Examine the SYSLOG log sheet scanning for messages and operator responses. Check responses given with those recommended in DOS/VS messages.	Look for unusual messages (those that do not obviously apply to jobs running may give you a clue). Any incorrect responses will require a user rerun of the job(s). Proceed to check 3.
Check 3 POWER/VS initialized	Check that POWER/VS has been correctly initialized by checking for message 1Q12I.	Check DLBL, EXTENT, and LABEL information. If message 1Q12I has not been issued, go to Chart 06; otherwise, proceed to check 4.

*An example and description of a formatted standalone dump is given in Appendix G of DOS/VS Serviceability Aids and Debugging Procedures.

Chart 03. LOOP, Part 2 of 2



Check

Method of Analysis

Recommendations for Further Problem Analysis

Check 4

Trace path of loop.

Use the loop trace provided by the user and the dump to locate the path of the loop. It may be a short loop in one CSECT or one routine. Use the listings of that CSECT or routine and the dump to check instructions executed.

If the trace is not sufficient (not enough instruction addresses noted), it may mean that the loop goes across one or more CSECTs or even between partitions (including the DOS/VS supervisor). In this case there are two methods of analysis open:

1. Rerun the job with an SDAID and/or PDAID* trace. Note, however, that an SDAID trace is only useful to trace code in the nucleus and/or code in a CSECT in the pageable area. Since tables and control blocks in the fixable area are dynamically allocated, there is no point in including this area in the SDAID trace.
2. If you consider that information available is sufficient, analyze the dump in an attempt to gather system information as indicated in Chart 02 (wait state) beginning at check 4 (ignore check 11).

* SDAID and PDAID are described in DOS/VS Serviceability Aids and Debugging Procedures.

Check 5

CSECT logic flow analysis

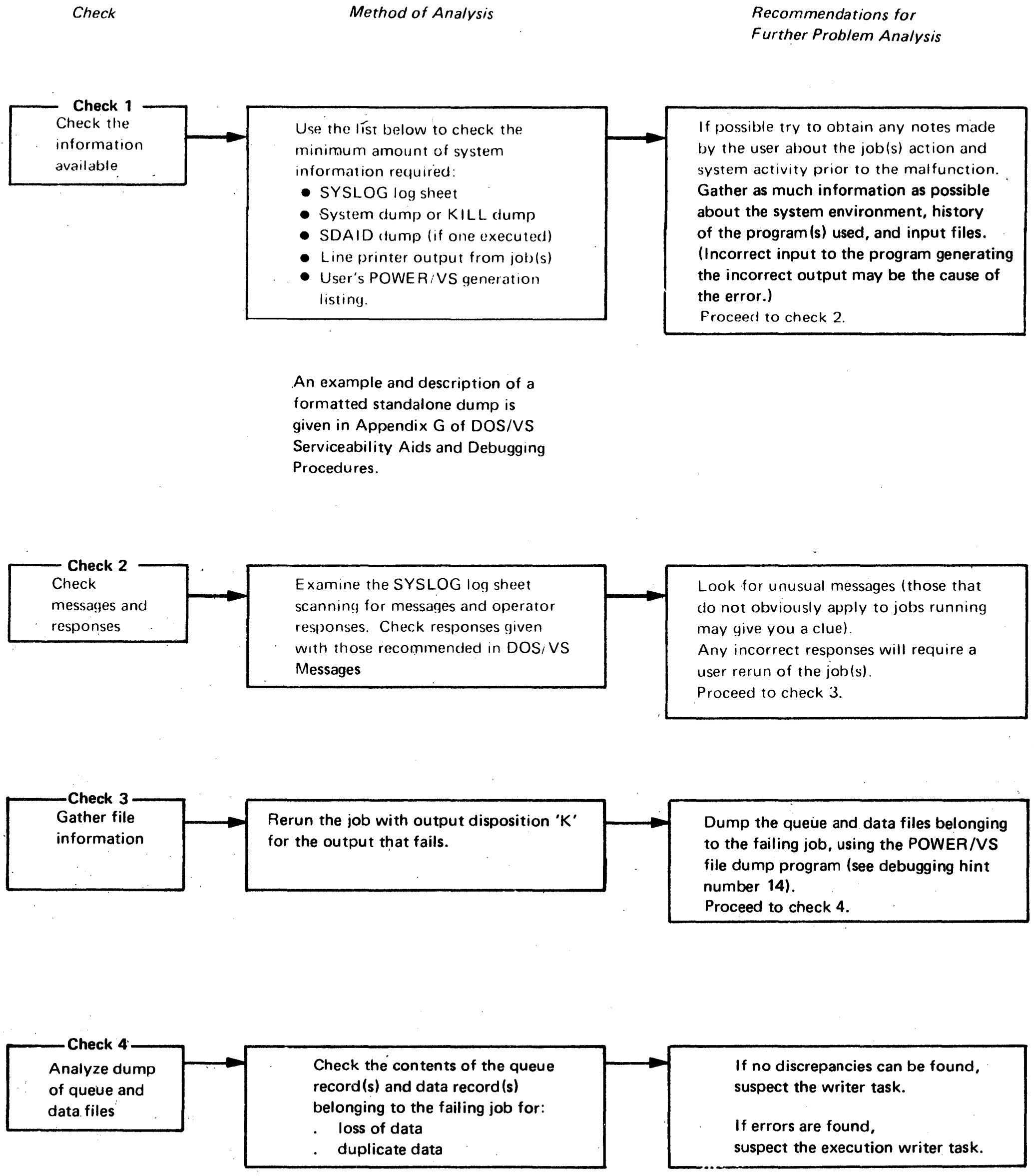
Check the logic flow of the CSECT(s) within the path of the loop for:

. Count locations that can never reach their limits.
. Counters that are not updated
. Counters that are never initiated
. Count locations, the contents of which are not periodically or properly tested
. Check that I/O operations are able to reach their completion (CCBs posted)
. Check ECB postings (Refer to appendix F).

Cross check these counters (ECBs, CCBs) in the dump for validity (not overwritten).

Chart 04. Incorrect Output

Check | *Method of Analysis* | *Recommendations for Further Problem Analysis*



Check 1
Check the information available

Use the list below to check the minimum amount of system information required:
- SYSLOG log sheet
- System dump or KILL dump
- SDAID dump (if one executed)
- Line printer output from job(s)
- User's POWER/VS generation listing.

If possible try to obtain any notes made by the user about the job(s) action and system activity prior to the malfunction. **Gather as much information as possible about the system environment, history of the program(s) used, and input files. (Incorrect input to the program generating the incorrect output may be the cause of the error.)**
Proceed to check 2.

An example and description of a formatted standalone dump is given in Appendix G of DOS/VS Serviceability Aids and Debugging Procedures.

Check 2
Check messages and responses

Examine the SYSLOG log sheet scanning for messages and operator responses. Check responses given with those recommended in DOS/VS Messages

Look for unusual messages (those that do not obviously apply to jobs running may give you a clue).
Any incorrect responses will require a user rerun of the job(s).
Proceed to check 3.

Check 3
Gather file information

Rerun the job with output disposition 'K' for the output that fails.

Dump the queue and data files belonging to the failing job, using the POWER/VS file dump program (see debugging hint number 14).
Proceed to check 4.

Check 4
Analyze dump of queue and data files

Check the contents of the queue record(s) and data record(s) belonging to the failing job for:
. loss of data
. duplicate data

If no discrepancies can be found, suspect the writer task.

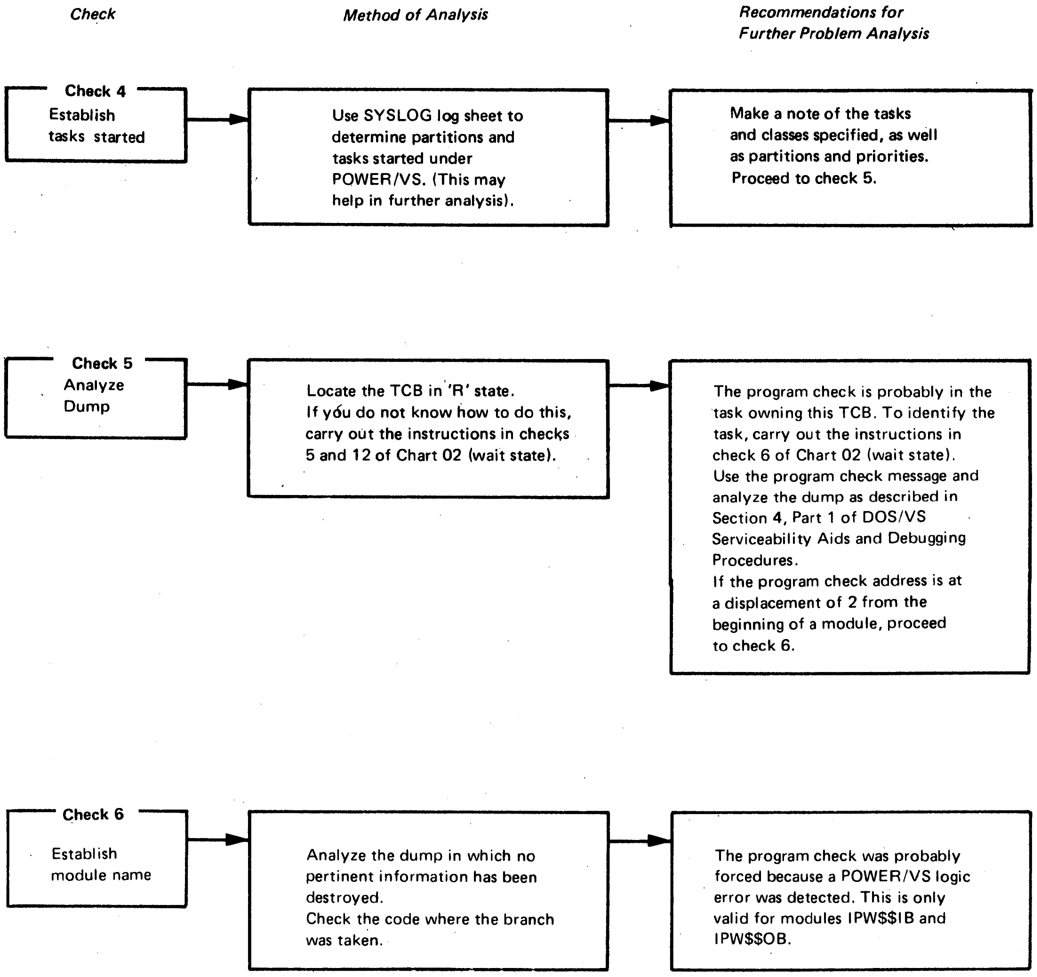
If errors are found, suspect the execution writer task.

Chart 05. Program Check, Part 1 of 2

Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 1 Check the information available	Use the list below to check the minimum amount of system information required: . SYSLOG log sheet . System dump or KILL dump . SDAID dump (if one executed) . Line printer output from job(s) . User's POWER/VS generation listing.	If possible try to obtain any notes made by the user about the job(s) action and system activity prior to the malfunction. Proceed to check 2.
Check 2 Check messages and responses	Examine the SYSLOG log sheet scanning for messages and operator responses. Check responses given with those recommended in DOS/VS messages.	Look for unusual messages (those that do not obviously apply to jobs running may give you a clue). Any incorrect responses will require a user rerun of the job(s). Proceed to check 3.
Check 3 POWER/VS initialized	Check that POWER/VS has been correctly initialized by checking for message 1Q12I.	Check DLBL, EXTENT, and LABEL information. If message 1Q12I has not been issued go to Chart 06, otherwise proceed to check 4.

* An example and description of a formatted standalone dump is given in Appendix G of DOS/VS Serviceability Aids and Debugging Procedures.

Chart 05. Program Check, Part 2 of 2



Check

Method of Analysis

Recommendations for Further Problem Analysis

Check 4
Establish tasks started

Use SYSLOG log sheet to determine partitions and tasks started under POWER/VS. (This may help in further analysis).

Make a note of the tasks and classes specified, as well as partitions and priorities. Proceed to check 5.

Check 5
Analyze Dump

Locate the TCB in 'R' state.
If you do not know how to do this, carry out the instructions in checks 5 and 12 of Chart 02 (wait state).

The program check is probably in the task owning this TCB. To identify the task, carry out the instructions in check 6 of Chart 02 (wait state).
Use the program check message and analyze the dump as described in Section 4, Part 1 of DOS/VS Serviceability Aids and Debugging Procedures.
If the program check address is at a displacement of 2 from the beginning of a module, proceed to check 6.

Check 6
Establish module name

Analyze the dump in which no pertinent information has been destroyed.
Check the code where the branch was taken.

The program check was probably forced because a POWER/VS logic error was detected. This is only valid for modules IPW$$IB and IPW$$OB.

Chart 06. POWER/VS not Initialized, Part 1 of 2

Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 1 Is the WCB present in the dump.	Locate the POWER/VS wait control block in the standalone dump. (Refer to Table 6.1 in this section if you do not know how to do this.)	If neither the WCB nor CAT appear to be in the dump, go to check 8; otherwise, proceed to check 2.
Check 2 Is initiator TCB attached.	1. Make a note of the address contained in bytes X'14' - X'17' in the WCB. (This will be a virtual address. Convert it to real by using the formatted list of page tables printed at the end of the standalone dump.*) 2. Locate this address in the dump. This is the first byte of the initiator TCB. * An example and description of a formatted standalone dump is given in Appendix G of DOS/VS Serviceability Aids and Debugging Procedures.	Examine byte X'1C' of TCB. It should contain X'C9' (I). If not, it may contain one of the following hex codes: ● C3(C), go to check 5 ● D9(R), proceed to check 3 ● D7(P), suspect the DOS/VS page manager routines.
Check 3 Task in R state	Examine the contents of the registers in the TRSA in the TCB.	**The values in the registers record their contents when the task was given control. Register conventions are described in Section 3. Examine the register saved in the POWER/VS partition save area and the return PSW.* The address in this PSW will point to the next instruction to be executed. Proceed to check 4.** * Refer to DOS/VS Serviceability Aids and Debugging Procedures for details about the partition save area.
Check 4 Analyze task in R state	Examine the PIB entry for the POWER/VS partition. Byte 0 should be X'83'. Examine partition save area of POWER/VS partition; R1 may point to CCB or ECB.	This is probably a DOS/VS error not caused by POWER/VS. However, check that the POWER/VS code has not been changed by the user, since this error indicates that an SVC 7 was issued by the POWER/VS task rather than by task management. If the cause of the wait state cannot be isolated, proceed to check 6.

Chart 06. POWER/VS not Initialized, Part 2 of 2

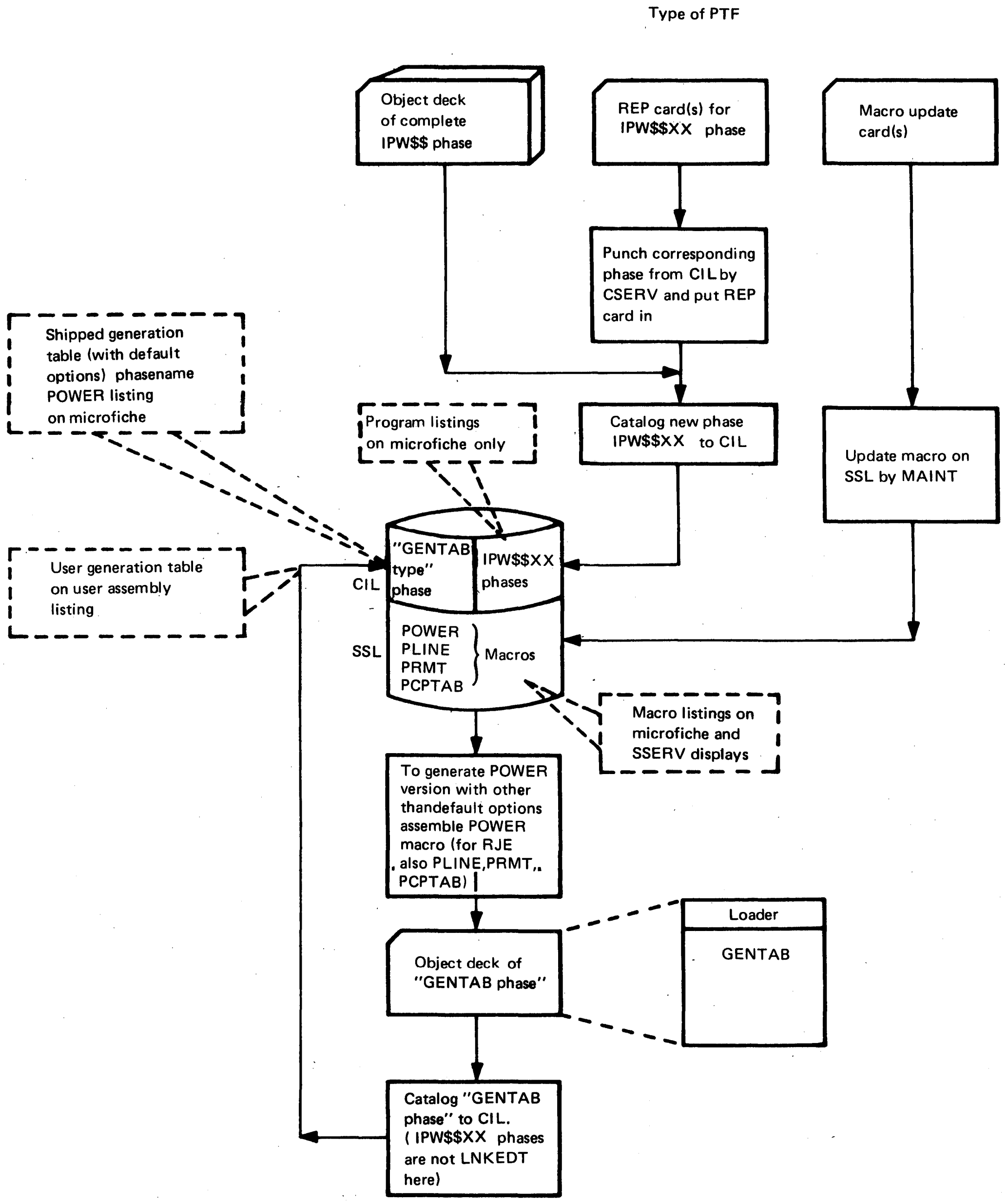
Check	*Method of Analysis*	*Recommendations for Further Problem Analysis*
Check 5 Task in C state	C state implies that the related task is waiting for posting of the traffic bit (bit 16) within the control block addressed by register 1 of the task in question.	Examine bit 16 in the conventional CCB or the POWER/VS ECB addressed by R1 in the TRSA in the TCB. If bit 16 = 0, the CCB is not posted and the system may be in a wait caused by a lost I/O interrupt. Examine CCBs in user partitions running under POWER/VS. These CCBs will be addressed by R1 in the relevant PSAs. A branch should have been made to the task scan routine to examine the next TCB in the TSL. If bit 16 = 1, the task should have been dispatched.
Check 6 Check tables setup by I2	1. Examine addresses of phases **in the module load address part** of the CAT (see Figure **5.4**) Check them and all relocatable constants for correct initiation.	If fields are not initialized correctly, proceed to check 7.
	2. Examine the SCB, which should be completely initialized. (See Figure **5.6**)	If not initialized, cause of error may be PFIX routine.
	3. **Examine the MMB and** check initialization of the channel program. (CCWs should point to an area within the CB.)	
	4. Check if there is a temporary operator command processor TCB chained in the TSL.	If there is a temporary OCPR in the fixable area, the dump was executed **during the IPW$ICP macro routine.**
Check 7 I2 loaded correctly	Examine registers 8 and 9 in the PSA of the POWER/VS partition.	If they are initialized as base registers for I2, compare the listings of I2 with the code in the dump. If all appear correct, check if the listing is at correct level and has not been changed by the user. If there appear to be discrepancies, proceed to check 8.
Check 8 I1 loaded.	**Look for the phase I1 and the GNB. If I1 has been loaded correctly, the GNB will immediately follow the label area in the POWER/VS partition at X'140'. The phase I1 should be loaded immediately after the GNB.**	**Check data in the GNB against the** POWER/VS generation listing supplied by the user. Check the listing to see that I1 is at correct level and has not been changed by user.

Appendix A. Abbreviations

Abbreviation	Meaning
AARA	Auxiliary account record area
ACB	Account control block
ACB	Access method control block
AQ	Add to queue
AQRA	Auxiliary queue record area
AS	Asynchronous service
ASAB	Asynchronous service anchor block
AWS	Account work space
BCA	Buffer control area
BCW	Buffer control word
CAT	Control address table
CB	Control block
CCB	Command control block
CP	Command processor
CPB	Command processor control block
DLRSA	Double linkage register save area
DMB	Disk management block
DQ	Delete from queue
DRW	Disk request word
EAR	Execution account record
ECB	Event control block
ER	3540 Diskette reader
ETX	End of text
FCB	Forms control buffer
FQ	Free queue
GD	Get data record
GNB	Generation table
IB	SNA inbound processor
IC	Pass command routine
ICR	Internal reader
INIT/TERM	Initiator/Terminator
JECL	Job entry control language
LCB	Line control block
LDA	Logical data record area
LF	SNA logoff processor
LK	Lockword
LL	Logical list
LMF	Line manager field
LMGR	Line manager
LN	SNA logon processor
LP	Logical punch
LR	Logical reader
LRSA	Linkage register save area
LST	List
LTA	Logical transient area in the DOS/VS supervisor
LU	LUB/PUB update
LW	Logical writer
MCB	Module control block
MCTA	Master class table area
MECB	Master (main) event control block
MLT	Master line table
MMB	Message control block
MP	SNA message processor
MR	Master record
MRA	Master record area
MS	Message handler
MS	Message service in nucleus
MSCB	SNA message control block

Abbreviation	Meaning
NQ	Get next from queue
OB	SNA outbound processor
OE	Open 3540 diskette reader
OT	Open tape routine
PA	Put account record
PD	Put data record
PDB	Partition control block
PCB	Page control block
PDA	Physical data record area
PL	Physical list
PP	Physical punch
PR	Physical reader
PS	Print status report
PSA	Partition save area
PUN	Punch
PWS	Physical work space
QRA	Queue record area
QS	Queue set
RCF	Record control field
RDR	Reader
RE	User reader exit routine
RJE	Remote job entry
RM	Resource management routine in nucleus
RMCB	SNA remote control block
RPL	Request parameter list
RQ	Reserve queue record
RTAM	Remote terminal access method
SA	Save account
SAM	Sequential access method
SC	Scan reader JECL statement
SCB	Storage control block
SDA	Single data adapter
SKAD	Seek address
SL	Get SSL record
SLA	Separator line area
SLW	SLI work space
SM	Storage management service in nucleus
SN	SNA manager
SNCB	SNA control block
SPL	Spool parameter list
SPM	Spool management
SRB	Service request block
SUCB	SNA unit control block
TBB	Tape control block
TCB	Task control block
TM	Remote job entry routines
TMF	Task management field
TMS	Task management service
TR	Task terminator
TRSA	Task register save area
TSL	Task selection list
VS	Validation service in nucleus
VTAM	Virtual telecommunications access method
WACB	SNA work space
WCB	Wait control block
WTR	Writer
XJ	Scan execution JECL statement
XL	Execution list
XP	Execution punch
XR	Execution reader
XW	Execution writer

Appendix B. POWER/VS Organization of Libraries and PTF Application


Type of PTF
Object deck of complete IPW$$ phase
REP card(s) for IPW$$XX phase
Macro update card(s)
Punch corresponding phase from CIL by CSERV and put REP card in
Shipped generation table (with default options) phasename POWER listing on microfiche
Program listings on microfiche only
Catalog new phase IPW$$XX to CIL
Update macro on SSL by MAINT
User generation table on user assembly listing
CIL
"GENTAB type" phase
IPW$$XX phases
SSL
POWER
PLINE
PRMT
PCPTAB
Macros
Macro listings on microfiche and SSERV displays
To generate POWER version with other thandefault options assemble POWER macro (for RJE, also PLINE,PRMT,, PCPTAB)
Object deck of "GENTAB phase"
Loader
GENTAB
Catalog "GENTAB phase" to CIL. (IPW$$XX phases are not LNKEDT here)

Appendix C. POWER/VS Status Bytes in the DOS/VS Supervisor

SYSCOM*

Location X'5C' - X'5F' contains the address of the POWER/VS partition (label IPW$PDA in phase IPW$$NU), if POWER/VS is initiated.

Location X'42' contains a flag byte:

X'08' = POWER/VS supported
X'04' = POWER/VS initialized

Location X'B8' contains the address of resource table (PIB translate masks) in DOS/VS supervisor.

PARTITION COMREGS*

Location X'A0' - X'A3' contains the address of the partition control block (0 if no CB exists for this partition).

POWER/VS Flag Bytes

Location X'A4' contains POWER/VS flags:

X'80' = POWER/VS accounting support
X'40' = This partition under control of POWER/VS
X'20' = This partition is the POWER/VS partition

* Refer to DOS/VS Serviceability Aids and Debugging Procedures for a full description and locations of SYSCOM and the partition COMREGs.

Location X'A5'

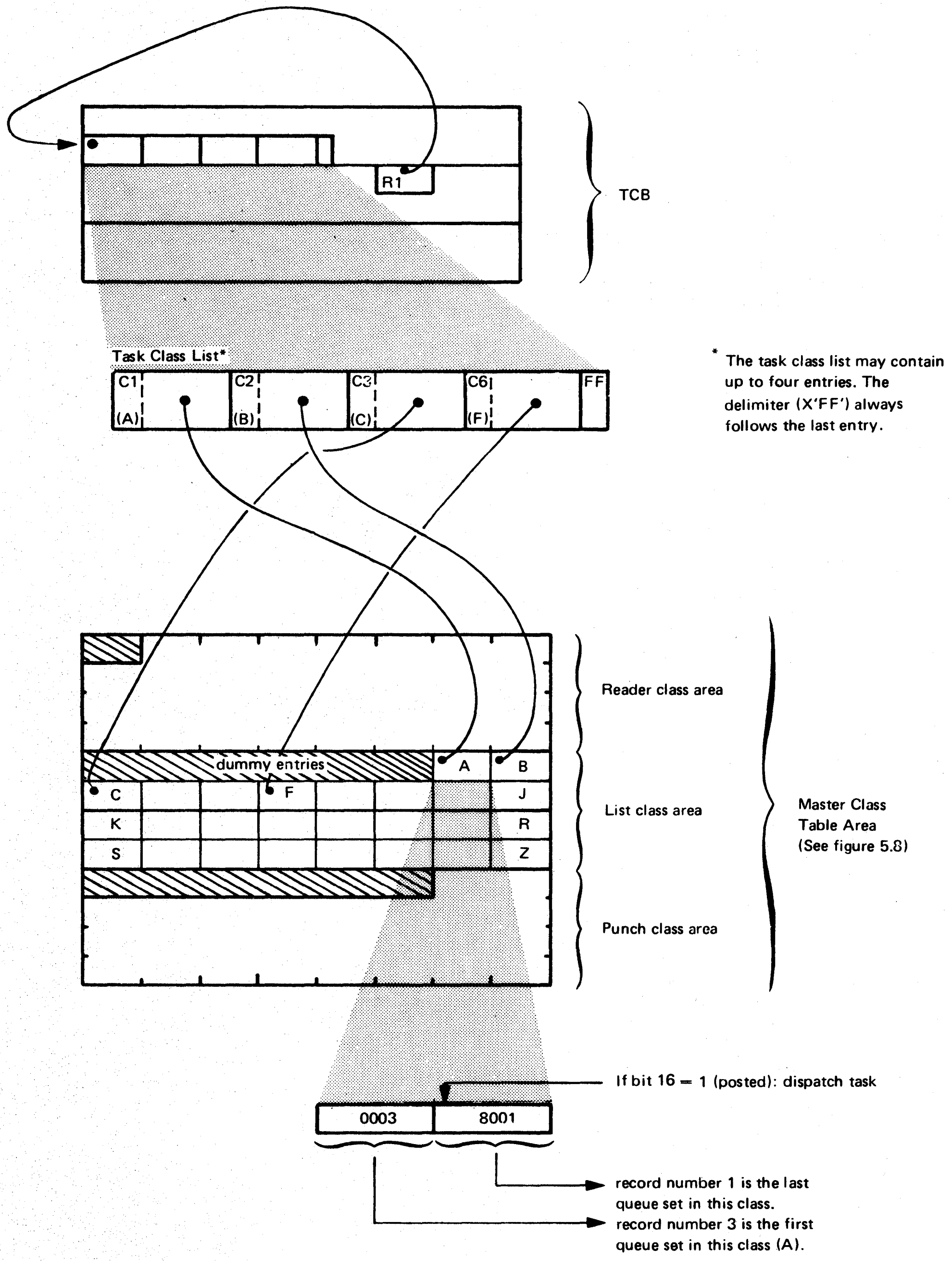
Not used.

Appendix D. Summary of ECB Usage

ECB in:	Posted by: (Phase)	Unposted by: (Phase)	Use when posted:
ACB	IPW$$GA	IPW$$PA	Indicates account file is emptied.
CAT	Appendage	Task selection mechanism	Indicates work to be done for POWER/VS.
SCB	Release work space	Reserve work space	Indicates that work space is available.
DMB	IPW$$FQ	IPW$$RQ and IPW$$PD	Indicates that queue space is available.
TCB (CP)	IPW$$I2	IPW$$CP	Indicates that IPW$$I2 has sent information to IPW$$CP.
TCB (LMGR)	• Channel end appendage • When line is started • When line is stopped	Line manager every time it is selected	Indicates work to be done for line manager.
TCB (LST)			
TCB (OB)	IPW$$IB, IPW$$MP, IPW$$SN	IPW$$OB	Indicates that RJE,SNA transmission to SNA terminal previously suspended is to continue.
TCB (PUN)			Not used.
TCB (RDR)			
TCB (SN)	VTAM (at completion of a RECEIVE ANY)	IPW$$SN	(VTAM ECB) indicates that the SNA manager (IPW$$SN) has to attach the SNA inbound processor (IPW$$IB).
TCB (SN)	IPW$$SN, IPW$$LN, IPW$$IB, IPW$$OB, IPW$$MP, IPW$$LF IPW$$VE, IPW$$LH, IPW$$MS	IPW$$SN	(Work ECB) indicates work to be done for the SNA manager (IP$$SN).
TCB (XP)	SVC appendage	IPW$$XP	Indicates work to be done for the execution processor.
TCP (XW)	Double buffering posting IPW$$I2	IPW$$PD	Disk control block freed and (if write operation) data area freed.
TCB INIT/TERM	IPW$$CP	IPW$$I2	Normally indicates that IPW$$CP has finished a task. In case of AUTOSTART indicates that information from IPW$$I2 has been received and processed by IPW$$CP.
SRB	IPW$$AS (SUBTASK)	Task offer posting	Indicates that the service request has been processed.
ASAB	IPW$$AS	IPW$$AS (SUBTASK)	Indicates that one or more service requests are waiting to be processed.

Note: An entry in the master class table area can be used as an ECB. In that case the address of the entry is contained in the task class list (ECB list) in the TCB. When the ECBs in the RDR, LST, or PUN class are posted (by IPW$$AQ), they indicate that an active entry exists in the class chain represented by this class table entry. These ECBs are unposted by IPW$$NQ.

For example, assume a TCB for a LST task in the queue state:



* The task class list may contain up to four entries. The delimiter (X'FF') always follows the last entry.

If bit 16 = 1 (posted): dispatch task

record number 1 is the last queue set in this class.

record number 3 is the first queue set in this class (A).

Appendix E. POWER/VS Status Report

STATUS REPORT FORMAT

This report is printed on two occasions:

1. When POWER/VS is initiated, when SYSLST is assigned to a line printer on the POWER/VS partition.

2. When POWER/VS is terminated with a PEND command that includes a physical address of a line printer.

The format is shown below:

```
                                                   POWER/VS STATUS REPORT  date  time


QUEUE FILE   IJQFILE

Total number of queue entries                                                    xxxx
Number of free queue entries                                                     xxxx
Maximum number of queue entries used in last session                             xxxx


DATA FILE   IJDFILE

Total number of tracks                                                           xxxx
Track group size                                                                   xx
Data block size                                                                    xx


ACCOUNT FILE   IJAFILE

Total number of tracks                                                           xxxx
Percentage of file that is filled                                                xx %


Real storage allocated to POWER/VS partition                                     xx pages
Number of times tasks were waiting for real storage                              xxx
Maximum number of real pages used                                                xx pages
Maximum number of tasks active at one point in time                              xx
```

If POWER/VS is initiated via a warm start a PDISPLAY ALL,cuu command is issued by the initiator task. (In addition to this status report) on the printer assigned to SYSLST (cuu).

A PDISPLAY ALL,cuu is also executed when a PEND cuu command is issued by the terminator task.

Glossary

account file: The POWER/VS account file is a direct access file maintained by POWER/VS to hold the accounting information generated by POWER/VS and the programs which it controls.

appendage routine: a set of code physically located in the POWER/VS nucleus but logically an extension of a DOS/VS supervisor routine.

AUTOSTART: A facility used to specify that a POWER/VS system is to be started automatically when it is initialized.

class: A means of grouping jobs that require the same set of resources for their execution. There are two types of class: input class and output class.

cold start: The initialization of input and output work queues. All information present in the queues before the cold start is lost.

data file: The POWER/VS data file is a direct access file maintained by the POWER/VS system to hold the input and output program data records required and generated by DOS/VS programs under POWER/VS control. The file may occupy from one to five extents of direct access storage according to the user's requirements. The total space provided by the user is divided into units called track groups.

disk request word: see I/O request word.

external routine: a set of code physically located in the POWER/VS pageable area, which provides task support external to the POWER/VS nucleus. Physical readers and physical writers are examples of external routines.

flushing: Flushing is used to discontinue a job that is being processed by a POWER/VS task. Processing continues with the next entry in the read queue.

function routine: a set of code within POWER/VS which performs high-level operations on POWER/VS files.

intermediate storage: A storage device (disk or tape) used in POWER/VS to which the information from the card reader is spooled before execution, or from which the information is spooled to the printer or punch after execution.

internal routine: a set of code physically located in the POWER/VS pageable area, which provides task support internal to the POWER/VS nucleus. Logical readers and logical writers are examples of internal routines.

I/O request word: An I/O request word is sixteen bytes of fixed storage (TCB) used to define a disk or tape input or output operation to the disk or tape service routines.

job entry control language: A control language which allows the programmer to specify how POWER/VS is to handle a particular job entry. Abbreviated JECL.

lockword: A lockword is a fullword of fixed storage associated with a specific resource and used by the Resource Management mechanisms to control task access to the resource.

master record: The first record within the POWER/VS queue file. It describes the free queue set within the POWER/VS system and hence provides the system with a warm start capability.

multitasking: concurrent execution of one or more subtasks attached to a main task within one partition.

page fault: A program interruption that occurs when a page that is marked "not in real storage" is referred to by an inactive page. Synonymous with page translation exception.

POWER/VS operator command language: An operator language used by POWER/VS to present operator commands. Abbreviated POCL.

priority: A rank assigned to each job within its class that determines its precedence in receiving system resources.

queue: A waiting line or list formed by items in a system waiting for service.

queue entry: A queue entry is a single direct access record describing one particular job element to the POWER/VS system; it is used to record the progress of that job element through the system. A queue entry contains a queue set together with its associated track groups.

queue file: The POWER/VS queue file is a direct access file maintained by the POWER/VS system to record the passage of user jobs through the system. The size of the queue file determines the number of jobs which POWER/VS can handle concurrently. A maximum of 1024 jobs is presently imposed on the system by programming.

queue record: A record on the queue file containing the address of a track group on the data file. The queue records which contain the addresses of the track groups for one input job stream form a queue set.

queue set: One or more direct access records (queue records) describing one particular job element to POWER/VS. It is used to record the progress of that job element through POWER/VS.

reenterable code: a routine or other set of instructions of which the same copy can be used concurrently by more than one task.

remote job entry: Submission of job control statements and data from a remote terminal, causing the jobs described to be scheduled and executed as though encountered in the input stream. Abbreviated RJE.

remote terminal: An input/output control unit and one or more input/output devices attached to a system through a transmission control unit.

resource: A resource (more properly "serial resource") is any code, control block, table, record, or file used by a task.

resource management: Resource management consists of the mechanisms that protect serial resources from concurrent access by competing tasks.

segmentation: A facility to break bulky list or punch output into segments so that printing or punching can be started before execution of the generating user program has been completed.

service routine: a set of code whithin POWER/VS which performs low-level operations (task scheduling, storage control, etc.).

spooling: The reading of input data streams and the writing of output data streams on auxiliary storage devices, concurrently with job execution, in a format convenient for later processing or output operations.

staging: The moving of data from an offline or low-priority device back to an online or high-priority device, usually on demand of the system or on request of the user.

storage descriptor: Storage Descriptors are sixteen-byte alphameric character strings which serve to identify important areas in a storage dump. Where appropriate, storage descriptors may also be used for programming purposes.

storage line: A storage line is a unit of program address space of 32 bytes and is aligned on a 32-byte boundary.

storage page: A storage page is a unit of program address space of 2048 bytes and is aligned on a 2048-bytes boundary.

tape request word: see I/O request word.

task: A task is the basic unit of synchronous program execution within the POWER/VS system. A task consists of instructions operating synchronously upon program data. Though a task is executed synchronously with respect to its own instructions, these are executed asynchronously with respect to all other tasks existing in the system.

task management: Task management consists of the mechanisms that allocate control of the central processor to the POWER/VS tasks that are competing for its use.

track group: The track group is the basic organizational unit of the data file. Each track group consists of an integral number of tracks; the track group size is calculated by POWER/VS at initialization time on the basis of the amount of direct access storage space provided by the user. This value may be overridden by the user should he wish to do so.

warm start: A restart that allows reuse of previously initialized input and output work queues.

SY33-8576-1

DOS/VS POWER/VS Logic Part 1 Printed in U.S.A. SY33-8576-1

IBM

IBM World Trade Corporation
821 United Nations Plaza
New York, New York 10017
U.S.A.

READER'S COMMENT FORM

DOS/VS POWER/VS
Logic Part 1

SY33-8576-1

Please comment on the usefulness and readability of this publication, suggest additions and deletions, and list specific errors and omissions (give page numbers). All comments and suggestions become the property of IBM. If you wish a reply, be sure to include your name and address.

COMMENTS

THANK YOU FOR YOUR COOPERATION
PLEASE FOLD ON TWO LINES, STAPLE AND MAIL

YOUR COMMENTS, PLEASE

Your comments on the other side of this form will help us improve future editions of this publication. Each reply will be carefully reviewed by the persons and department responsible for writing and publishing this material.

Please note that requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or the IBM branch office serving your locality.

IBM Germany
Publications Department
Schwertstrasse 58
D-7032 Sindelfingen
Germany

IBM

IBM World Trade Corporation
821 United Nations Plaza
New York, New York 10017
U.S.A.