LY12-5027-3
File No. S370/4300-36

VSE/POWER

Program Product

Program Logic Manual Part 1

VSE/POWER
Version 2 Release 1
Modification Level 0
Program Number 5666-273

VSE/POWER Shared Spooling
Version 2 Release 1
Modification Level 0
Feature Order Numbers 6016 and 6017

Fourth Edition (June, 1983)

This is a major revision of, and obsoletes, LY12-5027-2 and Technical Newsletter LN33-9328.

This edition applies to Version 2, Release 1, Modification level 0 of VSE/POWER, Program Number 5666-273, and to the VSE/POWER Shared Spooling feature, Version 2, Release 1, Feature Order Numbers 6016 and 6017. This edition also applies to all subsequent release and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

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

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

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

IBM Corporation
Dept 812BP
1133 Westchester Avenue
White Plains, NY 10604, USA

or to:

IBM Deutschland GmbH
Dept 3248
Schoenaicher Strasse 220
D-7030 Boeblingen, Federal Republic of Germany

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

PREFACE

The internal logic of VSE/POWER is described in three separate volumes:

VSE/POWER Program Logic Manual...

Part 1, LY12-5027 (this manual)

Part 2, LY12-5028

Part 3, LY12-5034

Part 1, this manual, contains:

- **Section 1**: Gives an overview of VSE/POWER, states requirements for operation, and lists the devices supported by VSE/POWER.
- **Section 2**: Refers you to Parts 2 and 3 for information on the "method of operation":
  - Part 2 describes the "method" for VSE/POWER.
  - Part 2 describes the "method" for the feature VSE/POWER Shared Spooling.
- **Section 3**: Outlines the logical structure of VSE/POWER; it explains the internal operations and shows the relationships between tasks and routines.
- **Section 4**: Lists program identifiers. This allows you to establish the relationship between phases, modules, and control sections. This part also lists the VSE/POWER messages and the relating modules.
- **Section 5**: Describes the layout of the VSE/POWER partition, account records, control blocks, and the work areas required by VSE/POWER.
- **Section 6**: Gives debugging hints, and shows how you can get information from a dump of the VSE/POWER partition.

At the back of this manual, you find:

- **Appendixes**: They expand on the information given in the above sections.
- **List of Abbreviations**.
- **Bibliography**: Lists manuals you may want to consult.
- **Glossary**: Explains some of the terminology used in this manual.
- **Index**.

CONTENTS

FIGURES

1. INTRODUCTION

This section contains an overview of the Virtual Storage Extended/Priority Output Writers, Execution Processors and Input Readers (VSE/POWER) Program Product. It is organized as follows:

- Purposes of VSE/POWER. A general description of VSE/POWER and the way its major functions are performed under VSE.
- Communication with VSE/POWER. A summary of the VSE/POWER Operator Commands and the Job Entry Control Language, which allow the user to control VSE/POWER operations. The format of the messages issued by VSE/POWER is also explained.
- Environmental Requirements. The programming requirements for the various functions of VSE/POWER, and the basic organization of the VSE/POWER partition with its storage requirements. The machines and devices which are supported by VSE/POWER are listed under "Hardware Support".

PURPOSES OF VSE/POWER

VSE/POWER performs automatic spooling and priority scheduling under the control of the VSE/Advanced Functions supervisor. VSE/POWER occupies a virtual partition in which it is initiated and can service from one to eleven partitions (other than the VSE/POWER partition) 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 VSE/POWER partition. Output requests to list and punch devices are also intercepted, with the output being stored in output data areas of the VSE/POWER partition and later transferred to disk or tape. Printing and punching of the output from disk or tape is carried out when requested by the operator. Under the control of VSE/POWER, programs may be executed in either real or virtual mode.
The optional Shared Spooling feature permits the sharing of the VSE/POWER files that contain the spooled input and output among two or more VSE/Advanced Functions systems running in the same processor or different processors.

The use of the optional networking function, referred to in other parts of this manual as PNET, allows VSE/POWER to fully participate within networks consisting of other VSE/POWER nodes, JES2 NJE nodes or RSCS nodes.
Jobs, Output (list, punch), operator commands and messages can be transmitted from one computer system to another.
The methods of communication used are binary synchronous communication (BSC) lines and synchronous data link control (SDLC) lines.

Three major operations are performed under VSE/POWER control:

COMMAND TASK. Handles system operator commands and initiates other VSE/POWER tasks, or allows stopping of tasks.

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

RJE,BSC LINE MANAGER TASK. Controls line activities with remote terminals. The task is alive as long as VSE/POWER is active.

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 VSE/Advanced Functions subtask in which the interface with VTAM 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 VSE/POWER and a remote SNA work station.

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

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

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 VSE/POWER initialization when SPOOL=YES is specified in POWER macro and detached at VSE/POWER 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.

TAPE READ TASKS.[3] Performs the first part of a read operation and transfers information from a tape device to intermediate direct access storage. The operator can call for the concurrent execution of as many tape read tasks as he has physical tape units available.

RJE,BSC READ TASKS.[2] Perform the read operation for a remote station. Each RJE,BSC read task has the standard name '1RDR' 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 VSE/POWER cross-partition communication macro interface.

STATUS TASK. Scans the Queue file or network definition table and prints the status report or node information respectively on SYSLOG, a line printer, a terminal printer, or sends the status report/ node information back to the originating node.

OFFLOADING TASK. Performs one of two functions:

- Saves queue sets on tape, or
- Restores saved queue sets from tape to VSE/POWER queues.

The operator can call for the concurrent execution of as many offloading tasks as he has physical tape units available.

TIMER TASK. Supports the time-sharing approach used by the Shared Spooling feature. Interfaces with a VSE/Advanced Functions subtask that handles the timer intervals.

NOTIFY TASK. Controls the transmission to VSE/ICCF of any message destined for a VSE/ICCF user. The task is attached during VSE/POWER initialization when NTFYMSG=nnn is specified in the POWER macro; the task is detached at VSE/POWER termination.

PNET DRIVER TASK. Controls all activities on a PNET BSC or SDLC communication line. Processing is performed on a demand basis. The task is attached when the first PSTART PNET,nodeid is entered and detached when the last node is disconnected (signed off). The Network Driver also attaches a VSE/Advanced Functions subtask in which the interface with ACF/VTAM is opened, when the first PSTART for a node is given using a SDLC communication line.

TRANSMITTER TASK. Transmits job or output to another node in the network. Up to eight transmitters can be active at a time for any node currently connected. There can be a mixture of job and output transmitters active concurrently. The task is active as long as there are jobs or output eligible for transmission.

RECEIVER TASK. Receives either job or output from another node in the network. Up to eight receivers, which may be a mixture of job or output receivers, can be active at a time for any node which is currently connected. The receiver task is attached only for the duration of the transmission of one job or output.

CONSOLE TRANSMITTER TASK. Sends messages and commands to another node in the network. The task exists as long there are messages or commands to send.

CONSOLE RECEIVER TASK. Receives messages and commands from another node in the network. The task is associated with a specific node and is attached when the node is started and exists as long as the connection to that node exists.

CONNECT TASK. Establishes a SNA session between VSE/POWER and another node in the network. The task is attached either when a PSTART nodeid is entered by the central operator, in which case the task acts as primary application, or when a BIND is received, in which case the task acts as secondary application.

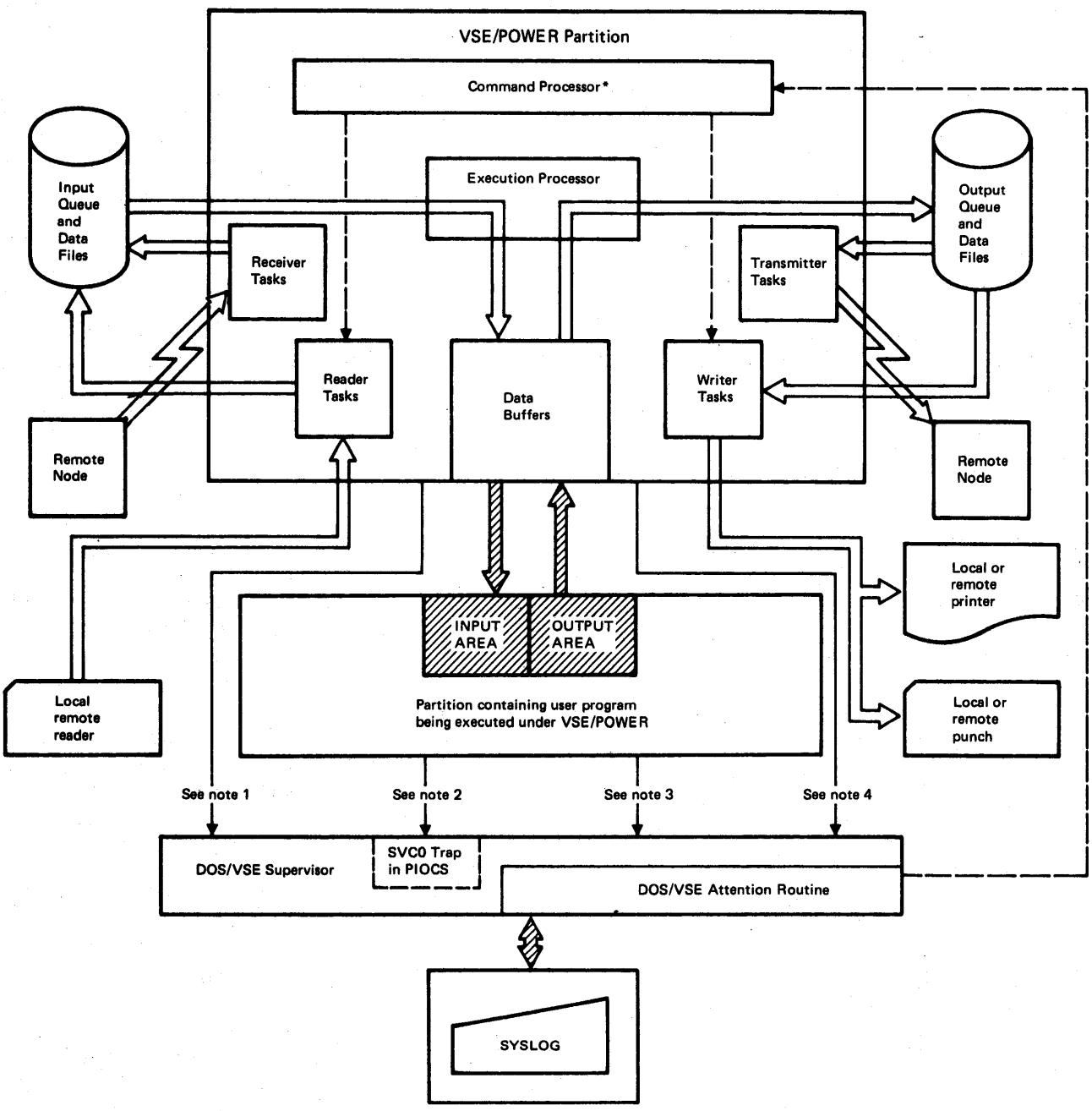


Figure 1-1. Relationship Between VSE/POWER, VSE/Advanced Functions, and the Program Running Under the Control of VSE/POWER

Job Entry Control Language

VSE/POWER provides a job entry control language (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 VSE/Advanced Functions itself. The JCL statements required for normal VSE/Advanced Functions system operation are also required when operating under VSE/POWER.

For a detailed description of the VSE/POWER JECL statements, refer to VSE/POWER Installation and Operations Guide.

Format of VSE/POWER Operator Messages

Messages sent by VSE/POWER 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 VSE/POWER general-message indicator.

R VSE/POWER message indicator for messages issued by RJE,BSC, the command-processor tasks, the Shared Spooling feature, and networking.

V VSE/POWER message indicator for messages issued by RJE,SNA tasks.

nn message-identification number. (May also include alpha characters).

I-type messages are for the operator's information only; no response is required. Processing continues normally.

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, if the Asynchronous Operator Support of VSE/Advanced Functions is not generated.

A-type messages require some action from the operator, such as mounting a tape. A-type messages for remote terminals are directly displayed on the remote printer. The VSE/POWER task issuing the message is put in the wait state.

Messages issued by VSE/POWER are listed in "Message-to-Module Cross-Reference List" on page 4-10 and are further described in the manual VSE/POWER Messages.

IPW$$BM
IPW$$BR
IPW$$BW
IPW$$CI
IPW$$LM

- The following macros must a cataloged in a source library.
  PLINE
  PRMT

3. For VSE/POWER RJE/SNA (optional)

- The following phases must be cataloged in a library.

IPW$$CI	IPW$$LN	IPW$$OC
IPW$$IB	IPW$$MD	IPW$$SN
IPW$$LF	IPW$$MP	IPW$$VE
IPW$$LH	IPW$$OB	

- The following macros must be cataloged in a source library.
  PRMT
  PCPTAB

4. For VSE/POWER Networking (optional)

- The following phases must be cataloged in a library.

IPW$$BS	IPW$$CX	IPW$$NR
IPW$$CAC	IPW$$IN	IPW$$NT
IPW$$CI	IPW$$LD	IPW$$SE
IPW$$CLD	IPW$$NC	IPW$$SR
IPW$$CN	IPW$$NK	IPW$$S1
IPW$$CPF	IPW$$NM	IPW$$S2
IPW$$CPS	IPW$$NP	IPW$$S3

- The following macros must be cataloged in a source library.
  PNODE
  PLINE (BSC only)

5. For VSE/POWER Accounting (optional)

- The following phases must be cataloged in a library.
  IPW$$GA
  IPW$$PA
  IPW$$SA
- If the account file is on an FBA device, the following phases must be cataloged in a library.
  IPW$$GF
  IPW$$PF
  IPW$$SF
- The following macros must be cataloged in a source library.
  PACCNT
  PUTACCT
- The required job accounting specifications must be given at VSE/Advanced Functions IPL time.
- An account file must be assigned to SYS000 on a spooling device. Account file space must be defined for IJAFILE on the label cylinder.

6. For VSE/POWER Cross Partition Communication Support (optional)

- The permanent area, consisting of five pages (10K), is permanently fixed in the real partition during initialization of VSE/POWER; if RJE,BSC support is used, the permanent area consists of six pages (12K). These pages are freed only when VSE/POWER is terminated (see Figure 1-3).

- 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 VSE/Advanced Functions requires additional real storage.

- The GETVIS area contains control blocks and work areas used by RJE SNA, PNET, VSE/Advanced Functions service routines, SYSIN tape support, printer setup processing (3800 or 3200), Notify message queue and I/O buffers for accounting if the account file is on an FBA device.

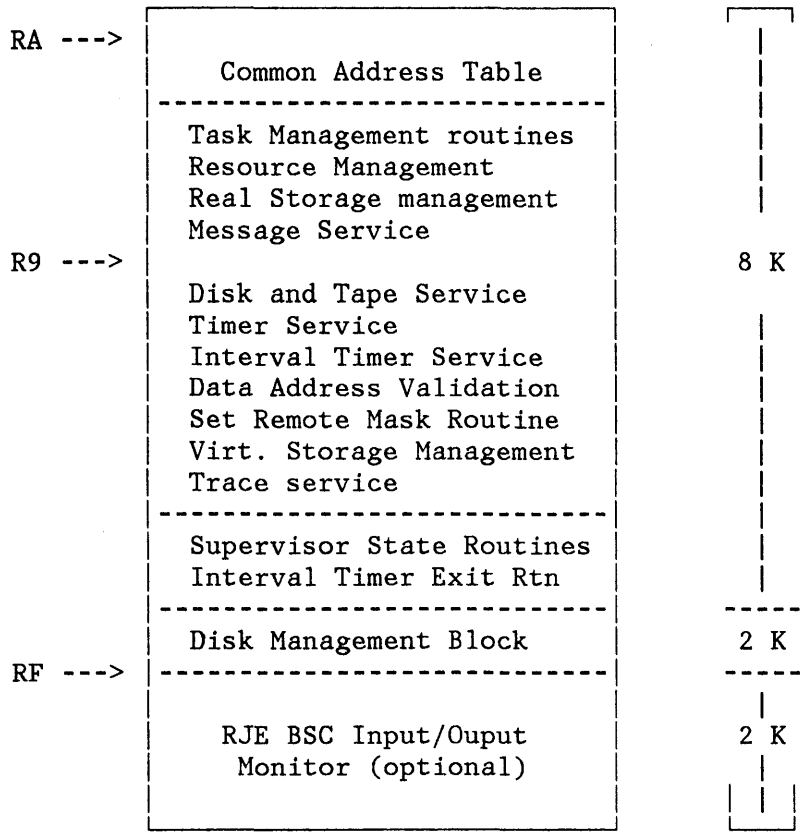


Figure 1-3. VSE/POWER Permanent Area

Real Storage

The minimum real-address space must be equal to the size of the permanent area (10K) plus the fixable area. The size of the fixable area (minimum = 4K)

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
Integrated Communications Adapter in the various models

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 VSE/POWER. 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. VSE/POWER 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:

   - VSE/POWER 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:

2. METHOD OF OPERATION

For the description of the method of operation, refer to:

- VSE/POWER Program Logic Manual, Part 2, LY12-5028.
  Part 2 describes the method of operation of VSE/POWER.

- VSE/POWER Program Logic Manual, Part 3, LY12-5034.
  Part 3 describes the method of operation of the Remote Job Entry (RJE) and networking functions of VSE/POWER.

3. PROGRAM ORGANIZATION

This section describes the program organization of VSE/POWER. It outlines the logical structure of the VSE/POWER Program Product, 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 VSE/POWER 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 VSE/POWER processing.

VSE/POWER 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.

The Spooling Process describes spooling to and from the queue file and the data file.

Services describes the routines of the nucleus phase.

Miscellaneous tasks and functions describes various tasks and functions that are not readily associated with the above areas.

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

VSE/POWER 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 VSE/Advanced Functions system control programs.

VSE/POWER Shared Spooling gives an overview of how the Shared Spooling feature works.

VSE/POWER Networking gives an overview of how the Networking function works.

- The second group of page frames are withdrawn from the VSE/Advanced Functions page pool and contain, firstly, the pages of the VSE/POWER permanent area and, secondly, those pages of the VSE/POWER fixable area which have been fixed in real storage by the VSE/POWER real storage management service.

Note:

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

2. For RJE,SNA operations, for 3800 Printer setup processing, for SYSIN tape support, for networking transmission buffers, for message queues, for PNET, for VSE/Advanced Functions service routines or for accounting (account file on FBA device), the GETVIS area in the VSE/POWER partition is used to allocate work areas and control blocks.
   Refer to Figure 3-72 for a description of the RJE,SNA control blocks and work areas and to 5, "Data Areas" on page 5-1. The SNA work areas within their pools are allocated by the SNA processors and freed when no longer needed. GETVIS storage pools for SNA control blocks and work areas for SYSIN tape and account support (if required) are acquired when required by VSE/POWER and freed when no longer needed. Transmission buffers (SNA only), control blocks and workareas for networking support are acquired only when the transmitters and/or receivers are active and are given back to the GETVIS pool when these tasks terminate. For 3800 Printer setup processing, the work area is allocated by the SETPRT logic module and freed when no longer needed.
   Some of the control blocks, such as the network definition table (NDT), are allocated at VSE/POWER initialization time and exist as long as VSE/POWER is active.

Code Structure

The code of VSE/POWER consists of External Routines, Internal Routines, Functions, Services, and Appendages, (see Figure 3-1).

External Routines

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 VSE/POWER pageable area.

The following external routines are provided:

IPW$$BR	RJE,BSC Reader
IPW$$BW	RJE,BSC Writer
IPW$$CA	PALTER Command Processor
IPW$$CAC	PACT Command Processor
IPW$$CB	PBRDCST Command Processor
IPW$$CC	PCANCEL Command Processor
IPW$$CD	PDISPLAY Command Processor
IPW$$CE	PEND Command Processor
IPW$$CF	PFLUSH Command Processor
IPW$$CG	PGO Command Processor
IPW$$CH	PHOLD Command Processor
IPW$$CI	PINQUIRE Command Processor
IPW$$CJ	PACCOUNT Command Processor
IPW$$CL	PDELETE Command Processor
IPW$$CLD	PLOAD Command Processor
IPW$$CM	Command Processor Root Phase
IPW$$CN	PDRAIN Command Processor
IPW$$CO	POFFLOAD Command Processor
IPW$$CP	PSTOP Command Processor
IPW$$CPF	PFLUSH PNET Command Processor
IPW$$CPS	PSTART PNET Command Processor
IPW$$CR	PRELEASE Command Processor
IPW$$CRE	PRESET Command Processor
IPW$$CS	PSTART Command Processor
IPW$$CT	PRESTART Command Processor
IPW$$CU	PSETUP Command Processor
IPW$$CX	PXMIT Command Processor
IPW$$ER	3540 Diskette Reader
IPW$$IB	RJE,SNA Inbound Processor
IPW$$LD	Network PNET Driver
IPW$$LF	RJE,SNA Logoff Processor
IPW$$LH	RJE,SNA Lgon Processor No. 1
IPW$$LM	RJE,BSC Line Manager
IPW$$LN	RJE,SNA Logon Processor No. 2
IPW$$MP	RJE,SNA Message Processor
IPW$$NS	Notify Support
IPW$$OB	RJE,SNA Outbound Processor
IPW$$OC	Outbound Compaction Manager
IPW$$OF	Offload Queues Routine
IPW$$PL	Physical List
IPW$$PP	Physical Punch
IPW$$PR	Physical Reader
IPW$$PS	Print Queue Status/ Node information
IPW$$NR	Network Receiver
IPW$$NT	Network Transmitter

IPW$$MS	Message Handler
IPW$$MX	Message Distributor / Modification
IPW$$NC	Network Composer
IPW$$NK	Network Compression/ Decompression
IPW$$NM	Network I/O Manager
IPW$$NP	Network Presentation Services
IPW$$NQ(S)	Get Next Queue Set from Chain
IPW$$OE	3540 Diskette Open
IPW$$OT	Open/Close tape
IPW$$PA	Put Account Record
IPW$$PF	Put Account Record for Account File on FBA Device
IPW$$PD	Put Data Record
IPW$$RQ(S)	Reserve Queue Record
IPW$$RY(S)	Queue-file Recovery
IPW$$SC	Scan Reader JECL Statement
IPW$$SL	Get Source-Statement-Library Record
IPW$$SR	PNET,SNA Send/Receive Manager
IPW$$TR	Terminate VSE/POWER Task
IPW$$XJ	Scan Execution JECL Statement

Services

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 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 VSE/POWER permanent area.

The following services are provided:

- Disk and Tape Service
- Message Service
  - Local Message Service
  - Remote Message Service
  - Notify Service
  - Nodal Message Service
- Remote Service
- Resource Management
- Storage Management
- Task Management
- Timer Service
- Interval Timer Service
- Validation Service
- Get Trace Entry Routine
- Virtual Storage Management

Appendages

Appendages provide code which, though physically present in the nucleus phase (IPW$$NU), is logically part of the VSE/Advanced Functions supervisor or of some

Macro	Purpose
Queue management	
IPW$AQS	Add queue set to chain
IPW$DQS	Delete queue set from chain
IPW$FQS	Free queue set
IPW$GQS	Get next queue set from chain
IPW$RQS	Reserve queue record
Data management	
IPW$GDR	Get data record
IPW$PDR	Put data record
Account management	
IPW$CAF	Close account file (delete contents of account file if on FBA device)
IPW$OAF	Open account file (not required for FBA)
IPW$GAR	Get account record (not required for FBA)
IPW$PAR	Put account record
Other functions	
IPW$BUF	Invoke PNET Buffer management
IPW$CNC	Cancel VSE/POWER
IPW$GMS	Invoke general message service
IPW$GSL	Get source statement library record
IPW$IAS	Invoke asynchronous service
IPW$ICP	Invoke command processor
IPW$IOC	Invoke outbound compaction manager (IPW$$OC)
IPW$IOM	Invoke RJE,BSC I/O Monitor, PNET I/O Manager PNET SNA SEND/RECEIVE Manager
IPW$OEF	Open diskette file
IPW$OTP	Open/close tape
IPW$SRJ	Scan reader JECL statement
IPW$SXJ	Scan execution JECL statement
IPW$ULP	Update LUB and PUB tables

Figure 3-3. Function Macros

Macro	Purpose
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
IPW$STM	Set timer interval
Validation service (VS)	
IPW$VDA	Validate data area addresses
Remote Service (RS)	
IPW$SRM	Set remote mask in bit table
Trace service	
IPW$GTE	Get trace entry

Figure 3-4 (Part 2 of 2). Service Macros

Macro	Purpose
IPW$DTB	Define tape control block (TBB)
IPW$DTC	Define task control block (TCB)
IPW$DTE	Define task control block extension area
IPW$DVC	Define PNET VTAM control block
IPW$DVD	Define various Dsect's
IPW$DVS	Define virtual storage control block
IPW$DWA	Define SNA work area (WACB)
IPW$DWC	Define PNET Composer workarea
IPW$DWG	Define PNET Receiver/Transmitter workarea
IPW$DWN	Define Receiver/Transmitter Account area
IPW$DWP	Define PNET Presentation service workarea
IPW$IOR	Define input/output request (RJE,BSC)

Figure 3-5 (Part 2 of 2). Definition Macros

Macro	Purpose
Entry/Exit	
IPW$ALN	Align to storage boundary
IPW$VCA	Validate command authority
IPW$EQU	Establish equates
IPW$GMD	Generate message definition
IPW$GMM	Generate message module
IPW$RET	Restore registers and return to caller
IPW$SAV	Save caller registers

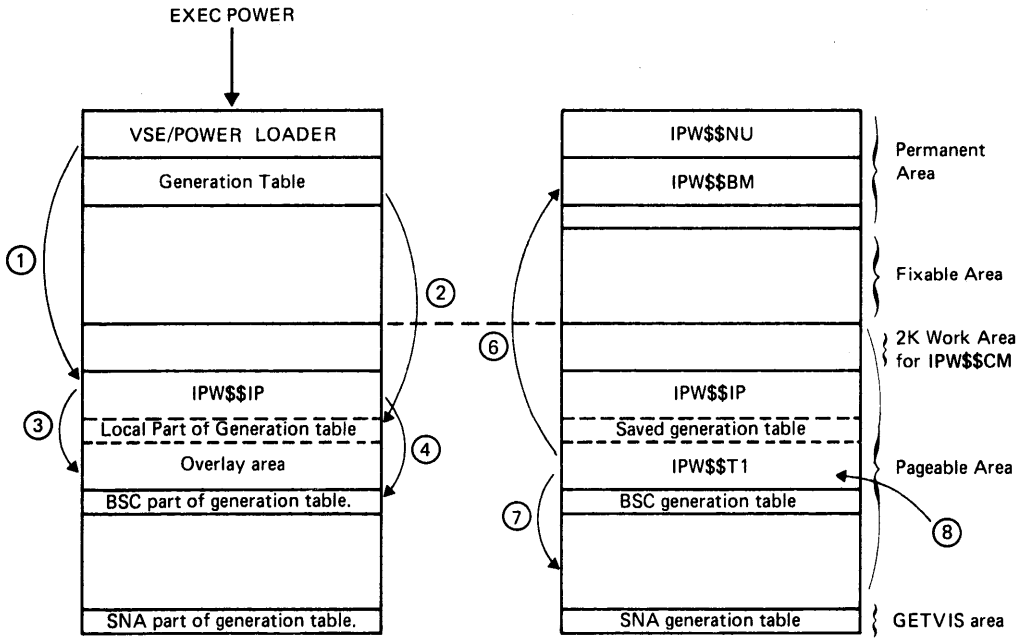
Figure 3-6. Miscellaneous Macros

INITIALIZATION AND TERMINATION

Initialization of VSE/POWER

The initialization of VSE/POWER comprises of the following phases:

- User-generated phase (POWER/username)
- IPW$$IP
- IPW$$I1(S)
- IPW$$I2(S)
- IPW$$I3(S)
- IPW$$I4(S)
- IPW$$I5(S) (optional for Accounting support)
- IPW$$I7(S)



1 VSE/POWER Loader loads IPW$$IP (start of pageable area + 2K).
2 IPW$$IP saves the local part of the generation table internally.
3 IPW$$IP loads the initialization phases into the overlay area and gives control to them.
4 IPW$$I1 saves the BSC portion of the generation table behind IPW$$IP.
5 IPW$$I1 builds the SNA control block (if required) in the GETVIS area and saves the SNA portion of the generation table.
6 IPW$$I1 loads IPW$$NU and IPW$$BM (if BSC is supported) and fixes them.
7 The other VSE/POWER phases are loaded into the pageable area.
8 The last phase loaded by IPW$$IP is IPW$$T1; this is the termination phase of VSE/POWER.

Figure 3-7. Initiation Logic

IPW$$I1:

- Checks if VSE/POWER is already active
- Checks if SYSLOG is assigned to a console device
- Checks if VSE/POWER runs as main task
- Checks if VSE/POWER runs in a virtual partition
- Checks if the VSE/Advanced Functions Supervisor supports JAI, if VSE/POWER accounting is requested
- Checks if the Shared Spooling feature is supported, when generated in the POWER generation, and if not then issues message 1Q0AI and terminates VSE/POWER initialization
- Checks if RJE,SNA is supported
- Validates the remote terminal ids for RJE,SNA
- Builds the remote control block (SNA)

- Checks if the same number of data-file extents are specified as at coldstart time
- Checks when queue file and data file are shared that further data file extents which do not reside on the same volume as the 1st extent are on shared device(s)
- Builds sector table in MCB of first extent
- Formats the data file (if coldstart)
- Validates the size of the track group or block group
- Formats queue file (if cold start)

IPW$$I5:

- Obtains device information of device containing account file
- Opens account file
- Builds and formats ACB
- Acquires input/output area (4K), if Account file resides on FBA device
- If cold start, erases account file
- If warm start, locates last written account record or CI if FBA device by invoking Account file recovery.

IPW$$I7:

- Attaches librarian subtask, if applicable
- Prints status report, if applicable
- Does final housekeeping
- Sets up timer task, if queue file and data file are shared
- Sets up Notify task, if applicable
- Sets up line manager task (if RJE,BSC)
- Sets up SNA control block (if RJE,SNA)
- Builds and formats remote message control block
- Writes VSE/POWER 'start-up' account record
- Performs autostart
- Activates page-fault appendage
- Deactivates initiator task

Initial task selection by VSE/POWER is illustrated in Figure 3-8 .

Warm and Abnormal Warm Start

The decision to perform a 'Warm Start' or an 'Abnormal Warm Start' depends on the contents of a field (the USE count), contained in the master record. When initialization is complete the USE count is incremented by one, and when a VSE/POWER system terminates normally the USE count is decremented by one.

If at initialization time the USE count is zero, meaning that there are no VSE/POWER systems active and that the last VSE/POWER termination was normal, a 'Warm' start is performed. All information required to control the spool files are extracted from the master record, without re-constructing the class chains on disk.

If however the USE count is non zero in a non-shared environment the Queue file recovery program (IPW$$RY) is invoked to re-construct the class chains and to reset all queue sets which are still marked 'in execution'.

If the USE count is non zero in a shared environment then another field, referred to as the SYSID-bucket, is used to check if the last termination of this VSE/POWER system was normal or abnormal. If the bucket contains an entry with the same SYSID as the initializing system, then the last termination was abnormal. In this case 'abnormal' warm start is performed. If however the last termination of this VSE/POWER system was normal, 'warm' start is initiated.

Note: Normal termination always resets the SYSID-bucket for that system to zero if it is in a shared environment.

An additional field, contained in the master record, is the 'SYSID' field, that contains the SYSID of the system that is currently accessing the the Queue file in update mode. This field is important after VSE/POWER initialization, because whenever a VSE/POWER system locks the Queue file, it first checks the SYSID field. If a SYSID is present then Queue file recovery is performed for the SYSID found in the field, without the operator having to re-initialize VSE/POWER in the abnormally terminated system.

Termination of VSE/POWER

VSE/POWER 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$$CM) has set a termination code ("S", "E", "F", or "H") in their TCBs. In case of an I/O error, VSE/POWER tasks can also be terminated by the IPW$$TR phase (see "Abnormal Termination of VSE/POWER Tasks").

After all supported partitions have been released, the VSE/POWER partition is restored for normal VSE/Advanced Functions operation.

The detach routine of task management actually gives control to the terminator routine IPW$$T1.

The terminator routine performs the following functions:

- It writes the final VSE/POWER execution account record.

When one of the VSE/POWER tasks itself detects an abnormal-termination condition, it issues the IPW$CNC (CANCEL) macro instruction, which stores the registers (but not the PSW) in the abnormal-termination save area and branches immediately to the abnormal-termination routine with cancel code X'FF'.

The operator is prompted to specify the printer device on which he wants the dump to be printed. If he specifies an invalid device (such as no valid printer device or device is down), the operator is prompted again. Otherwise SYSLST is assigned to the device just specified.

Additionally the option DUMP is set in the COMREG to obtain a dump at abnormal termination of the VSE/POWER partition.

All VSE/POWER indicators in all partitions running under VSE/POWER are turned off and all dummy assignments are released. Also, all programmer-logical-unit assignments in the VSE/POWER partition are released, except for DASD assignments.

VSE/POWER Multitasking

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

Each VSE/POWER 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 VSE/POWER 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 3-9) 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 VSE/POWER initiator (IPW$$I2). This list contains the wait control block, the task control block of the permanent command processor task, and the task control block of the initiator task. All further additions to and deletions from the task selection list are performed by the task management service.

- Task termination - detach current task.

Each of these components is discussed in the following paragraphs.

Task Initiation

Task initiation is entered from a VSE/POWER 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. This is illustrated in Figure 3-10 .

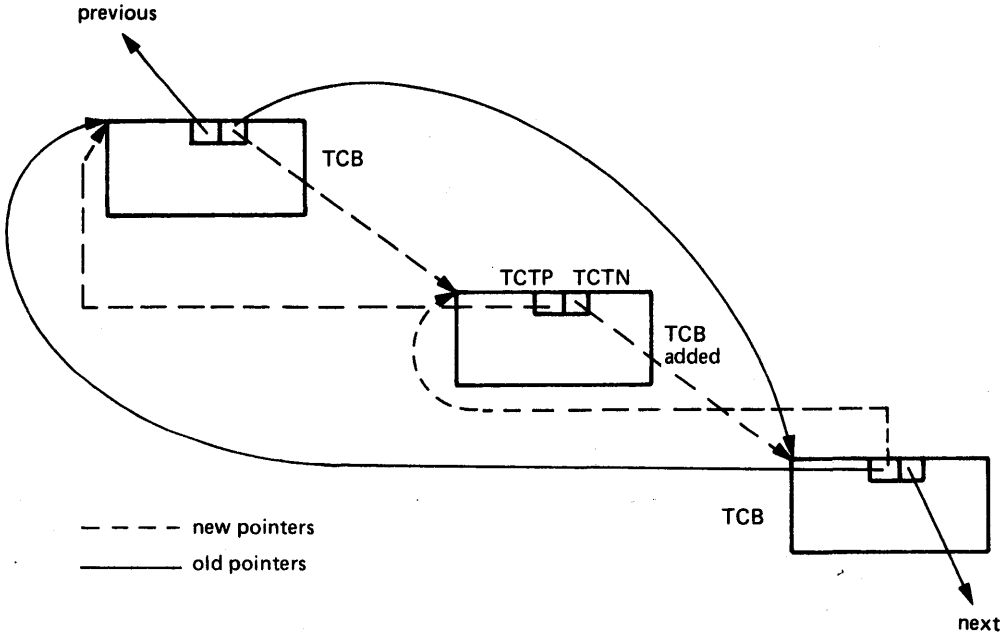


Figure 3-10. Attaching a Task

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, one additional state must be mentioned: P state (page-bound), which is set by the page fault appendage (see "VSE/POWER Appendages") when a task sustains a page fault.

Tasks in the following states are nondispatchable:

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.
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.
B state - the task is waiting for a RJE,BSC or PNET event.

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 VSE/Advanced Functions supervisor. Additionally the **no-work-to-do** ECB is posted when no task is waiting for a locked DMB or, if the Account file is shared, for a locked ACB. VSE/POWER will wait till the occurrence of some related event (I/O completion, for example) causes VSE to return control to the task selection service. The entire task selection process is then repeated.

READER, EXECUTION PROCESSOR, AND WRITER TASKS

The data flow throughout the reader, execution processor, and writer task is summarized by Figure 3-13 .

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 (see Figure 3-14). 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 (/370 mode only).

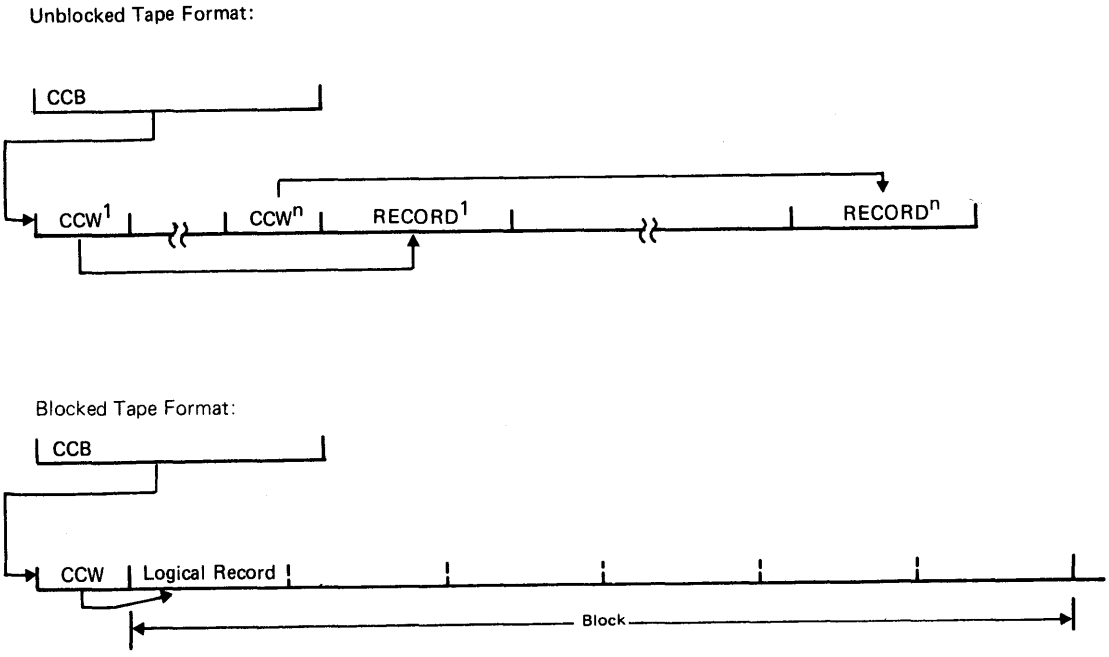


Figure 3-15. Physical Data Area - GETVIS Space

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.
If the routine is entered from the logical reader and no diskette unit is assigned (dynamic RDR support), the PUB table is scanned for a free, operational 3540 device. If no such unit is available, the job is flushed and the operator is informed via message 1Q90I. Otherwise the diskette unit remains assigned to the job until end of job is encountered.

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). A job header record is set up and passed to the put data record routine. 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) of the TCB indicates what action is to be taken by the PD routine.

General-purpose byte posted by LR:

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:

- Holds a copy of the job header record in core anchored to the partition control block of the partition concerned.
- Intercepts first request for output of the user channel program. Acquires storage for the queue record area and data set header record, initializes them with the VSE/POWER defaults and the information obtained from the job header record. Both areas are then anchored to the TCB of the new execution writer task and the task is then attached.
- Handles all input requests from the user channel program.
- In case of a writer-only partition, analyzes JECL statements from a console read/write 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 data set header record, and starts the writer task again.
- Completes the job trailer record with accounting information at end of job time.

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. SETPRT handling is illustrated in Figure 3-16

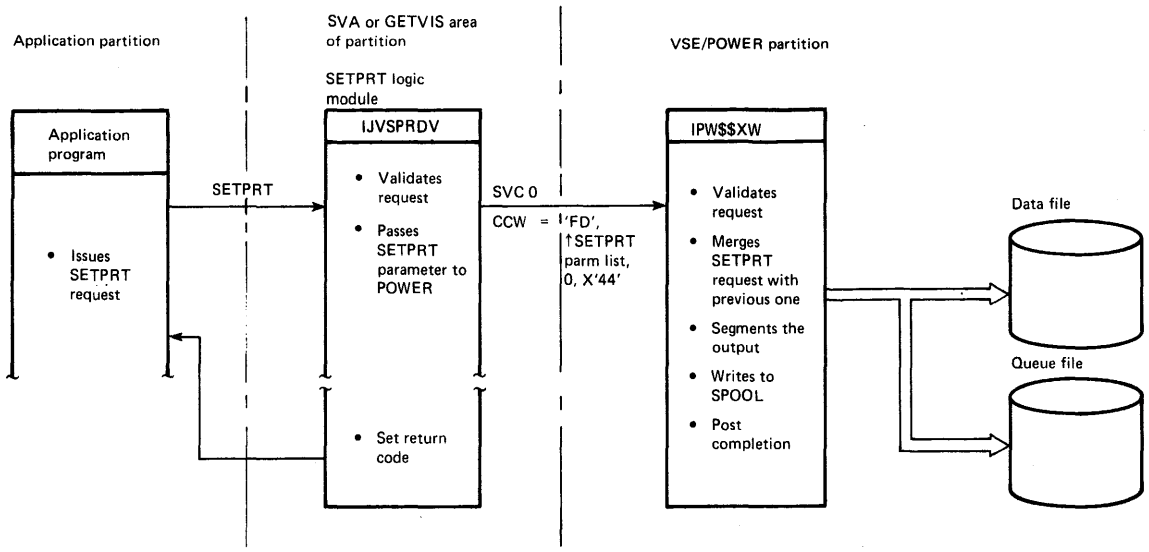


Figure 3-16. SETPRT Handling

The execution processor maintains a control block, called data set header record, which contains the current printer setup of the device being spooled. When a SETPRT parameter list is encountered by the execution processor, the printer setup 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 Job header record and data set header record are then written as first records 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 library and the internal representation of the page format is updated. The data set header record is updated accordingly. The FCB image is validated for accuracy. If a 3800 FCB image is invalid, a message (1Q54I) is written to the operator and the hardware default FCB is used. The LTAB specification is assumed as the internal representation of the FCB.

SETPRT processing is illustrated in Figure 3-17. If a SETPRT parameter list was passed by the logical writer, the partly filled print buffer is emptied and a IPW$IAS TYPE=SERVICE macro instruction is issued to perform the printer setup. Additionally if the DEBUG option was specified in the SETPRT parameter list, SYSLST is temporarily assigned for the duration of the setup. Each output operation will print or punch a number of records by means of command-chained CCWs (see "Physical Work Space"). The output operation is performed with real addresses in the CCWs (EXCP real, System/370 mode only).

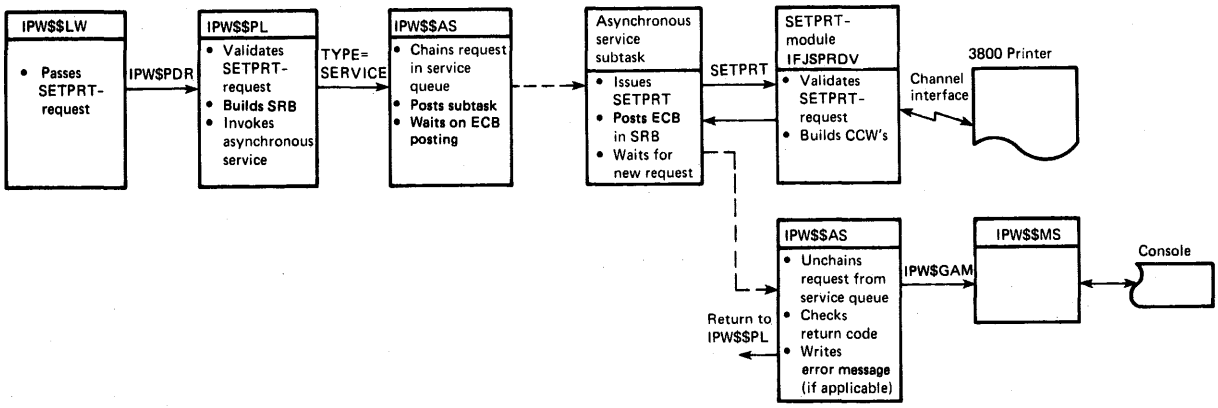


Figure 3-17. SETPRT Request Processing Flow

Logical Writer

A new queue set is addressed by invoking the get next queue set (NQ) function. If no queue set is eligible, a physical writer task is placed in a wait state until a new eligible queue set is added or an existing queue set becomes available. In all other cases the logical writer routine returns to its caller with an indication that there is nothing to do.

The routine ensures that the controlled printer/punch or even remote station is set up with the requested forms. The forms id obtained from the queue record is compared with the one in the TCB of the writer task that specifies the actual setup. If a mismatch requires operator intervention (for 3800, a mismatch of forms, flash or burst status), message 1Q40A or 1QA6A is issued. The operator must then either perform the setup and use the PGO command to continue processing, or else stop the writer task or flush the output via the PSTOP/PFLUSH command respectively.

A warning message (1Q41I) is issued if a different printer/punch device is used at physical print/punch time as was used at execution time. The operator can then decide whether to continue with the output or to flush the output. If he decides to continue, all illegal commands passed to the physical device are ignored and the output may contain invalid data or may lose records.

Start separator pages/cards are produced, if applicable at the beginning of each outout and between copies.

A logical record is retrieved from the data file by invoking the get data record (GD) function.

LOAD Function

This function will be used to restore queue sets, residing on tape, to the VSE/POWER queues. Only those sets can be restored which match with the specified queue identifier of the POFFLOAD command. If the LOAD ALL function is requested then all queue entries will be restored from the tape. The restore function operates independently of block size, which means that different DBLK sizes may be used between save and load time. The queue entries will be restored according to their class and disposition, unless the operator specifies a class.

The POFFLOAD tape format which is identical to the spool tape format is shown in Figure 3-18 .

Figure 3-18. Internal Tape Format

THE SPOOLING PROCESS

Queue File Organization

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

A master record queue identifier M (physically the last record in the file). Contains the class table, node attached table, shared spooling information and other control fields which must be retained between the termination and initialization of VSE/POWER.

Queue records queue identifier F, R, L, or P

A dummy record queue identifier D. Indicates the logical end of the queue records.

An internal record queue identifier I (physically the first record in the file). Contains the seek address of the master record in its first eight bytes.

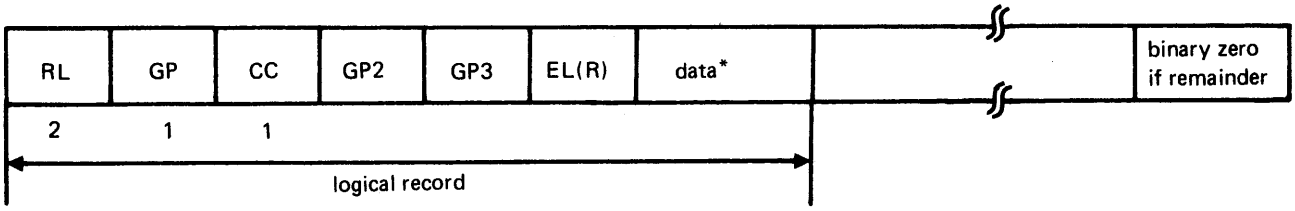
Logically, by means of pointers, the queue records are either a member of the free queue set or a member of a queue set that is in turn a member of a class chain. The free queue set is shown in Figure 3-19 .

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

Data File Organization

The space available on the data file is arranged in track groups (C-K-D) or block groups (FBA). Each track or block group contains an integral number of tracks or blocks, respectively. Each track group or block group has a fixed relationship with a queue record in the queue file. Hence, a queue record in the free queue chain will point to an available track group or block group. A queue record in a class chain will point to a track group or block 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. Figure 3-21 shows the layout of a physical record:



RL length of the logical record

GP general purpose byte (see also sections "Logical Reader (LR)" and"Logical Data Record Area (LDA)")

CC command code associated with the user channel program

GP2 general purpose byte 2

GP3 general purpose byte 3

EL(R) extended record residual length (for the first record extension, is equal to the total record length)

* trailing blanks are suppressed

Figure 3-21. Data Record

The Job Header, Job Trailer and Data Set Header contain information which is required for job routing, execution, printing, punching and accounting.

BASIC HEADER/TRAILER FORMAT: The basic organization of the Job Header or Job TRailer records is shown in Figure 3-22.

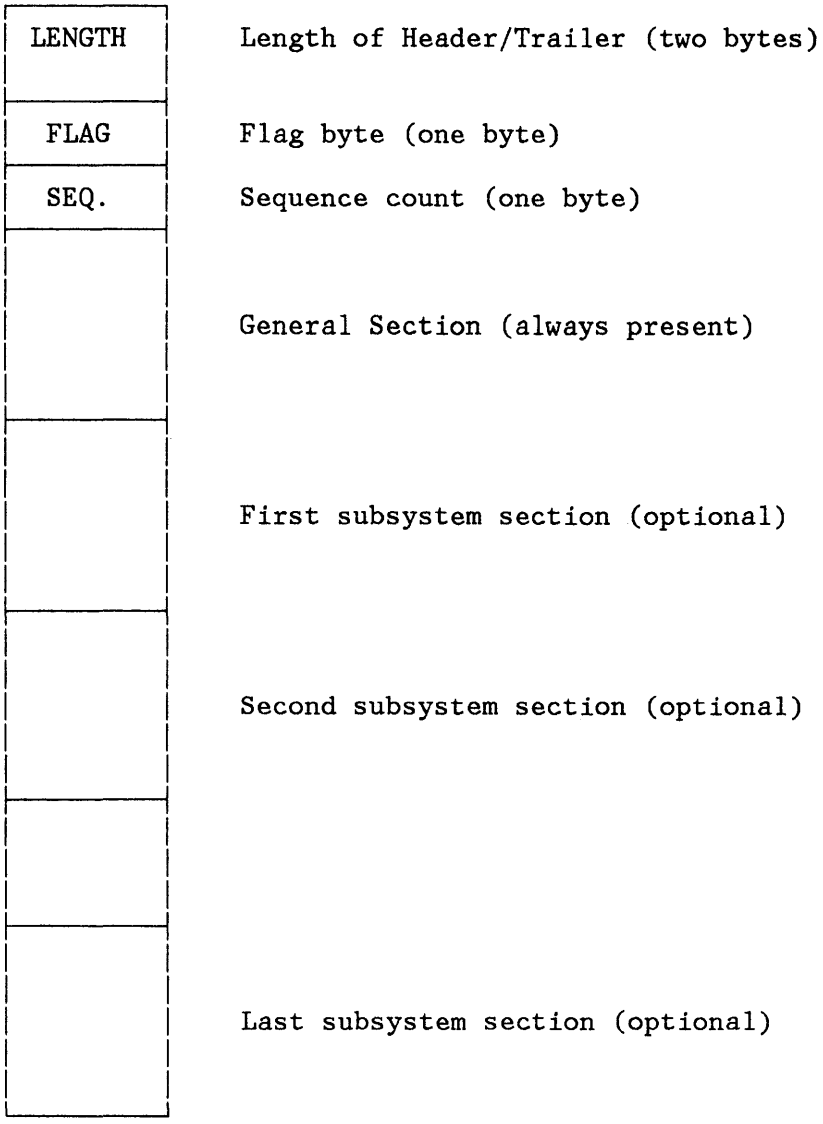


Figure 3-22. Job Header and Job Trailer Format

JOB HEADER AND JOB TRAILER: The Job header record contains four types of information relating to the job as a whole:

- Identification (job name, job number, originator's name)
- Routing Control (execution node name, default print/punch node and remote names).

- 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 chain and updates the pointer in the master record to the next record in this chain. 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 chain of a specific queue.

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

If the execution node for a job (that is the system upon which the job is to be executed) or the destination node for list or punch output is not the local node, the job or output is placed in the transmission queue (XMT). This queue consists of two entries: One for jobs and one for output. The queueing is performed on a priority base only.

The output-available flag is set in the line control block or logical unit control block (LUCB) when the output is destined for a remote terminal that is currently logged on (terminal or logical unit).

When a job or output is destined for processing by a specific SYSID in a shared-spooling environment, the SYSID class table is updated depending upon the specification of the class and SYSID.

If a job or output is destined for another node in the network, a check is made if a connection exists to the prime route node. If so the first inactive transmitter found is posted. If no such connection exists, a check is made if a connection exists to the alternate route node. If so, the first inactive transmitter found is posted.

If the destination of the job or output is unknown, the local operator, and the originator if Notify was specified, are informed via message 1RA1I and the job or output is put in hold status in the XMT queue.

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 or block group is exhausted, a new queue record is obtained from the free queue set (explicitly coded without invoking a function).

If the data record is larger than the DBLK size - 8 bytes, the record is split into record extensions that fit in the data block.

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.

If the resource to be exclusively reserved is either the DMB or the ACB, and the resource is not available, and it is a shared environment, the work-to-do ECB is posted in order to interrupt the T3 time interval.

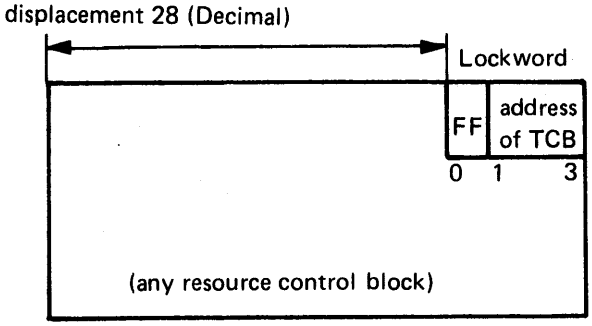


Figure 3-23. Resource LOCKWORD of a VSE/POWER Control Block.

Release Resource

This service is entered when a VSE/POWER 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.

Real Storage Management

Storage management controls the permanently fixed storage allocated via the ALLOCR command to the VSE/POWER 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 pfixed address storage in the VSE/POWER partition (see Figure 3-24). The SCB is locked during handling reserve/release request.

Virtual Storage Management

Virtual storage management controls the GETVIS storage allocated to the VSE/POWER partition. Work space in the GETVIS area for a task is reserved and released as requested by the calling routine. Storage management makes use of the subpooling possibilities to control storage allocation. The smallest unit of storage that may be reserved is 128 bytes.

Reserve Virtual Storage

The reserve-virtual-workspace service is entered when a VSE/POWER task issues a reserve virtual storage (IPW$RSV) macro instruction. The macro permits the caller to specify a pool type with the request.

The service routine rounds up the requested length of the workspace to a multiple of 128 and selects the subpool anchor, according to the type specified by the caller. If the caller demanded alignment of the workspace on page boundary, the appropriate request is passed to VSE/Advanced Functions.

If workspace is available the routine clears the storage area, allocates it to the caller and initializes the workspace header (also referred as virtual buffer control area). The header contains control information used by virtual storage management and precedes each acquired piece of storage.

The workspace is chained as last entry in one of following queues:

- own task queue (head and tail pointer are contained in task TCB)
- other task queue (head and tail pointer are contained in TCB of other task)
- system queue (head and tail pointer are contained in the virtual storage control block or any other major control block).

Release Virtual Storage

The release-virtual-workspace service is entered when a VSE/POWER task issues a release virtual storage (IPW$RLV) macro instruction. The routine removes the storage area from the virtual storage queue and frees the storage area by issuing the VSE/Advanced Functions FREEVIS macro. If the storage area to be freed is not a member of the task virtual storage queue, the head and tail address must be provided by the caller. The routine posts the virtual storage ECB to show that storage is now available and returns to the caller.

Unchain Virtual Storage Element

The unchain-virtual-storage service is entered when a VSE/POWER task issues an unchain virtual storage (IPW$UNV) macro instruction. The routine performs two functions:

1. A specific storage element, addressed by register 1, is removed from the specified virtual storage queue and chained to another queue.

2. The first element of a specified queue, if any, is unchained and chained at the tail of the issuing task virtual storage queue. The address of the element is returned in register 1. If the queue is empty, register 1 is set to zero.

Message Service

Local Message Service

The local message 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 3-26). 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.

Message service is also used to obtain a message from the message definition module, which contains most local and remote messages, by the use of the IPW$GAM macro instruction. It performs one of the following functions:

- Move message into user-supplied area
- Return message address
- Write message to central operator
- Add message to remote message queue

- Local operator
- Any remote operator locally attached
- Any local VSE/ICCF user
- Any user on another node

The message id, target node and remote name, if applicable, are supplied by the calling task in the message request word and in register 0.

Note: No information is passed back to the calling task on whether or not the message was sucessfully queued.
The variable portions of the message text are converted to indicate information pertinent to the specific task or queue entry when combined with this message.

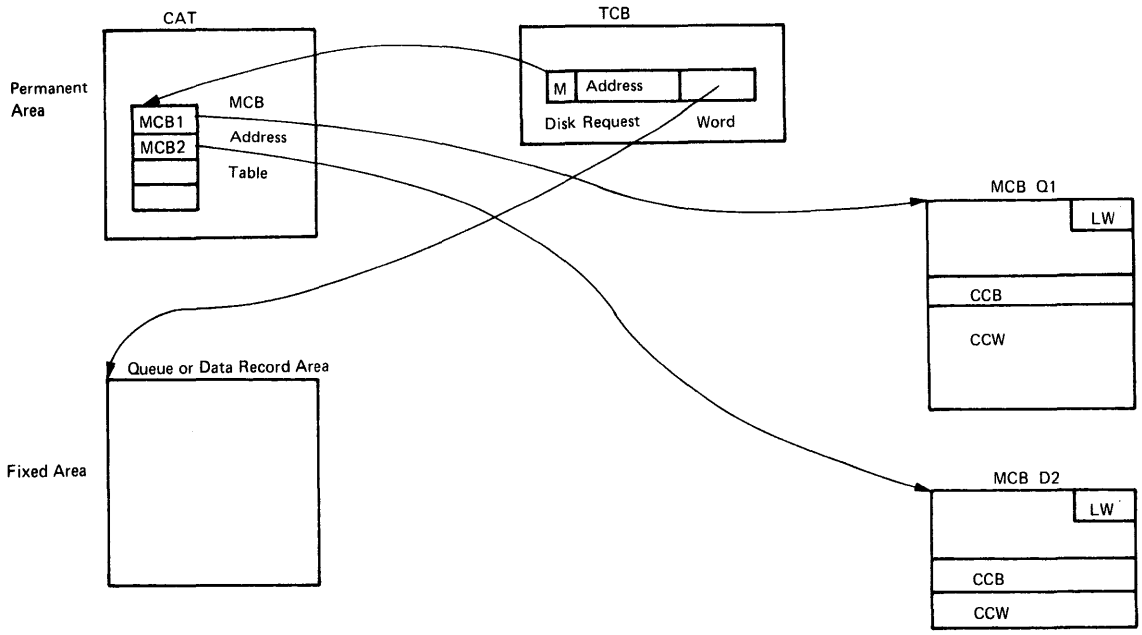


Figure 3-27. Disk Management Control Blocks Relationship

Tape Service

Tape 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 tape control block (TBB). The TBB is associated with the tape device and contains the skeletal channel program. See Figure 3-28.

gets control, it checks if a timer interval is expired. If so, the interval timer routine is invoked which de-queues expired TQE's and posts associcated tasks for work.

Validation Service

Validation 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 user is allowed to access the user's partition, the logical transient area, and the shared virtual area, for read, write, or control operations. This is illustrated in Figure 3-29.

	User Partition	LTA	SVA
DATA AREA	Valid	Valid	Invalid
CCB	Valid	Valid	Invalid
Channel Program (CCW)	Valid	Valid	Valid

Figure 3-29. Areas Checked by Validation Service

Remote Service

Remote service is invoked by the IPW$SRM macro instruction. Depending on the option specified, the bit representing the remote id is either turned on or off in the remote bit mask. The remote bit mask indicates which remote users are signed on at any time.

Get Trace Entry

This routine is invoked by the IPW$GTE macro instruction. The routine allocates a trace entry from the VSE/POWER trace table and returns its address in register 1 to the caller. If the current trace area is filled, the routine swaps to the alternate trace area and if trace logging was requested, a IPW$IAS TYPE=SERVICE macro instruction is issued to dump the filled trace area to the VSE/Advanced Functions dump file. Figure 3-30 shows the two trace areas and how they are used.

- Queue remote messages (BSC and SNA) to the remote message queue.
  If the message is in NMR format, the originating node and/or remote/user id are put in front of the actual message. The message is truncated to its maximum length if applicable. Message modification is performed, if appicable by executing the IPW$GMS TYPE=SUB macro instruction. Then the message text is examined and multiple blanks are deleted from the text. Finally the message is anchored by means of a message index to the line control block if BSC or logical unit control block if SNA respectively.

- Delete messages from the queue when it is completely full with pending messages.
  When the remote message queue is full (255 entries) with pending messages, it is assumed that somebody is monopolizing the queue. This can be the case when a remote printer has not been ready for a while. All messages for that remote user are deleted and replaced by message 1R20I.

- 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. The ALLUSER type message queue contains only a limited number of entries (15). When the queue is full, the queue request is rejected.

- 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 temporarily deleted SNA messages to the SNA live subchain.

For nodal message requests, the function to be performed is indicated in register 0. The following functions are performed:

- Add nodal message record to appropriate node control block.
  On entry the routine acquires storage to hold the NMR to be queued. If no storage could be obtained, the routine returns to the caller with R1=4. Otherwise the message text is copied from the NMR to the just acquired storage area. If the NMR contains a message, originated from the local node, message modication is invoked by issuing the IPW$GMS TYPE=SUB macro instruction. The Network definition table is scanned to find the prime and, if specified, the alternate route node name. The NCB chain is now scanned to check if a connection is established with one of these nodes. If so, the NMR is queued at the tail of the message queue on this NCB. The network driver is then posted, to attach a console transmitter task, if one does not already exist.

- Build nodal message record and add it to appropriate node control block.
  On entry, the routine locates the message to be sent in the message definition module and acquires storage to hold the message. If no storage could be obtained, the message is discarded. Otherwise the message text is copied

Mod id	Message modification	obtained from where
X'01'	PNET BSC trace input	NCB and Reg. 15
X'02'	PNET BSC trace output	NCB and Reg. 15
X'03'	PNET SNA input buffer information	NCB and Reg. 15
X'41'	BSC transmission count	NCB / LCB
X'42'	BSC time out count	NCB / LCB
X'43'	BSC error count	NCB / LCB
X'44'	node name	Node control block
X'62'	member name.member type	SL member element
X'63'	macro name (librarian)	Register 15
X'64'	return code/feedback code (librarian)	Register 15
X'71'	Number of PNET SNA sends	Node control block
X'72'	Number of PNET SNA receive's	Node control block
X'73'	From node and user id	Queue record
X'74'	First operand	Command control block
X'75'	original job number	Queue record
X'76'	From node / user	command control block
X'77'	first 60 bytes of operands	command control block
X'80'	8-byte field	Register 4
X'81'	Sense information	VTAM RPL
X'82'	CCB address	Register 7
X'83'	Unit address address	Task control block
X'84'	RJE identifier	Line control block
X'85'	RTNCD,FDB2	VTAM RPL
X'86'	Forms id	Queue record
X'87'	Return code, reason code	Register 15
X'88'	3800 printer setup message	Queue record
X'89'	Application id	VTAM ACB
X'8A'	Tape address	Tape control block
X'8B'	UCS phase name	TCB extension area
X'8C'	Keyword in error	Register 7
X'8D'	File name	Register 4
X'91'	Job name	Queue record
X'92'	Target node name	Queue record
X'93'	RJE,BSC line address	Line control block
X'94'	LU name	LUCB
X'95'	Job number	Queue record
X'96'	Command code	Command control block
X'97'	SNA stop code	WACB
X'98'	RPL request	VTAM RPL
X'99'	RJE identifier	Task control block
X'9A'	5-digit number	Register 4

Figure 3-31 (Part 1 of 2). Message Modification Characters and Action Table

celed and the operator is informed via message 1Q08I. Next, the connection to VSE/ICCF is established. If VSE/ICCF is not yet active, the Notify task waits until VSE/ICCF is brought up and itself tries to establish the connection to VSE/POWER. After the connection with VSE/ICCF is established, the first message is removed from the Notify message queue and a 'send' is initiated. Once VSE/ICCF successfully retrieves the message, the storage occupied by the message is freed. This process continues until all messages are sent.
If, meanwhile, the connection to VSE/ICCF breaks due to VSE/ICCF normal/abnormal termination, the Notify task attempts to re-connect to VSE/ICCF, however this connection might never complete.

If a severe error is encountered while sending messages to VSE/ICCF the Notify support is terminated, the message queue is emptied, all storage is freed and the operator is informed via message 1Q4BI. The Notify task is posted for immediate termination at VSE/POWER termination by the IPW$$T1 module.

Asynchronous Service

The asynchronous service function of VSE/POWER handles all of the following requests:

- SETPRT
- LFCB
- OPEN/EOV/CLOSE
- LOAD
- Dump particular storage areas
- Communicate with the Librarian

This is done for the following reasons:

1. Most of the called VSE/Advanced Functions service routines run in the SVA under the TIK of the VSE/POWER main task. Any page fault which occurs cannot be correctly handled by the VSE/POWER page fault overlap processing, because the register convention used by the VSE/Advanced Functions service routines do not match the VSE/POWER requirements.

2. Any I/O done by the VSE/Advanced Functions service routines would cause the complete VSE/POWER partition to be put into the wait state until the I/O completed.

3. Any SVC-call would cause the VSE/POWER main task to wait until the service request completed.

When a service request is issued by a VSE/POWER task, the service request block (pointed to by register 1) is chained as the last entry in the appropriate service request queue, and the corresponding 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 return code set by the subtask is analyzed and the appropriate action is taken.

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

COMMAND PROCESSOR

The command processor (IPW$$CM) will be under control either of a permanent TCB located in the first page of the fixable area 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), which is part of the task control block (TCB).

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 VSE/POWER partition in extreme circumstances.

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 VSE/POWER 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 already active an error return to $$BATTNA is taken, resulting in an invalid-statement message or a routine-active message, respectively.

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

The following commands are processed for BSC by the IPW$$BR BSC reader:

- SIGNON
- SIGNOFF
- START
- STOP
- GO
- SETUP

The remaining commands are given to IPW$$CM for processing. The main processor is responsible for de-coding each command and performing the processing necessary to cause appropriate action to the operator's request. A command table, containing the long and short form of the command verb, is used to determine the subprocessor to be entered. If the end of table is encountered or no subprocessor entry point address is present, the command is considered to be invalid. Each entry in the table may have restriction indicators as follows:

- command not allowed during VSE/POWER shutdown period
- command not allowed at autostart time

If one of the conditions is true, the command is rejected. After processing the command, control is returned to task management service.

The following command processing routines exist:

PACCOUNT The following functions are performed by the PACCOUNT command:

- Builds a TCB for a save account task which will save or delete the account file IPW$$SA (IPS$$SF if account file resides on an FBA device).
- Conditionally assigns a free program LUB entry.
- Reserves work space for the TCB and attaches it.

PACT Activates the specified transmitter or receiver and makes it available for network activity.

PALTER Removes if necessary the queue sets specified in the command from their class chains (via IPW$DQS); then SYSID, destination, userid, priority, disposition, class, number of copies, remote ID, and compaction table name can be changed. Finally, the queue sets are returned to the appropriate class chains (via IPW$AQS macro instruction).
PALTER also changes the class(es) of an active partition.

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 0).

PHOLD Takes one or more VSE/POWER jobs out of the dispatchable state and puts them in the hold/leave state.

PINQUIRE displays status information for RJE lines/sessions and the network status:

- 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)
- Status of each connection to other node
- Status of each transmitter/receiver
  - active
  - inactive
  - drained
  - Halting (in process of closing down)

PLOAD Loads the specified Network Definition table from the VSE/Advanced Functions library.

POFFLOAD Creates a TCB according to the first parameter specified (RDR-TCB=LOAD, LST-TCB=SAVE), sets up a parameter list in the just created TCB and attaches the related TCB.

PRELEASE Changes the disposition of the queue sets specified in the command:

- Disposition H (hold) to D (dispatch)
- Disposition L (leave) to K (keep)

PRESET Resets the in-execution flag for all jobs or output belonging to the sysid that was specified in the command.

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.

Account File Processing

Operations on the account file are performed by two functional routines:

- The put account function (IPW$$PA for C-K-D devices, or IPW$$PF for FBA devices) invoked by an IPW$PAR macro instruction,
- The get account function (IPW$$GA for C-K-D devices, or IPW$$GF for FBA devices) invoked by an IPWOAF, IPWGAR, or IPW$CAF macro instruction.

The put account function routine will accept account records for the VSE/POWER partition and the partitions running under control of VSE/POWER (see Figure 3-32). The account records (VARUNB format) will be written to the VSE/POWER 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, as used by the save account function, is broken down into three operations:

- Open account file for get mode processing, invoked by IPW$OAF macro instruction. This function is not supported for FBA devices.
- Get account record to retrieve the next sequential record from the account file, invoked by IPW$GAR macro instruction. This function is not supported for FBA devices.
- Close account file to restore the mode for put account record processing, invoked by the IPW$CAF macro instruction. For FBA devices, this function writes an SEOF (software end of file) to the account file.

Open Account File

The account control block is initialized for read operations (get mode) to retrieve the first record of the account file.

Close Account File

The account file records are erased by writing EOF records on each track for C-K-D devices or on the first block for FBA devices. The account control block is initialized 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$$SA for C-K-D devices or IPW$$SF for FBA devices, 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).

First it checks if there are buffers queued, by the VSE/POWER BSC channel end appendage or ACF/VTAM Send or Receive Exits, on the buffer queue anchored to the PNET Driver TCB. If so, the buffers are re-ordered from LIFO to FIFO sequence. Each buffer is then processed by IPW$$LD1, for BSC nodes, or IPW$$LD2, for SNA nodes.

When all buffers are processed, a check is made for the presence of ACF/VTAM related events and IPW$$LD4 is called to process them. Looping through the nodes (as represented by the chain of node control blocks), checks are made for actions requested for a node and if found the request is passed to module IPW$$LD3. A further check is then made for buffers to process before testing if the ACF/VTAM Interface may be closed in IPW$$LD4. The PNET Driver waits to be posted for work to do and when dispatched continues the loop.

The PNET Driver detaches itself when there are no active nodes and ACF/VTAM interface is closed.

PNET BSC Buffer Processing (IPW$$LD1)

This module handles all buffers sent and received when the nodes are connected via a BSC line. A multi-leaving (MLI) line discipline is used to control the line. Each buffer completed is checked for errors.

If the buffer is found to be in error or contains information that the remote found an error, recovery is attempted. If the error is found to be unrecoverable or cannot be recovered within the retry limits, the node is flagged for termination.

If the buffer is found to be error free, it is analysed to see if it contains data records or NJE stream control information. Data buffers are queued to the proper receiver.

If the buffer contained NJE stream control information then the proper actions, like flagging a receiver creation or flagging and posting of transmitters and receivers, etc., is performed.

The module then invokes the BSC I/O manager to execute the next Send-Receive operation.

PNET SNA Buffer Processing (IPW$$LD2)

This module handles all buffers sent and received within an SNA session (established by module IPW$$S2 when the node was started). For SNA, a buffer also contains the RPL used to send or receive the data, which is then checked for an error and if found the node is flagged for termination and the buffer is freed.

If a SEND completes without error the buffer is freed after effecting any status change for transmitters or receivers as indicated in the buffer.

If a Receive completes without error, the RPL is first analysed for ACF/VTAM commands and if any is found then the node is flagged for termination. Non-command buffers are then analysed in the same way as in IPW$$LD1.

- Request Termination of Transmitters and Receivers
- Delete a Node from Node Attached Table (NAT)
- Attach a VSE/POWER task for Session Creation/Termination
- Request Timer Events from Timer Services

PNET Initialization

If PNET= is specified in the POWER macro, the VSE/POWER initialization processor calls the PNET Initialization phase (IPW$$IN). This module

- sets up the PNET master control block(PNCB) and stores its address in the CAT.

- loads all PNET phases (BSC and SNA) in the pageable area after the last loaded phase for the local part.

- Loads the Network definition table, in the GETVIS area, by invoking the PLOAD command processor.

  Note: The PLOAD command processor checks if the Network definition table is valid.

- Acquires virtual storage for the temporary node attached table(NAT). This table is used as interface between the PNET Driver and the Timer task and contains entries for each node which was signed-on or signed-off since the last time interval. The timer task updates the NAT-table contained in the Master record according to the entries found in the temporary NAT-table.

  Note: The temporary NAT-table is only acquired in a shared spooling environment.

If an error occurs during PNET initialization, the appropriate error message is written to the system console operator and VSE/POWER is terminated normally.

PNET STARTING A BSC CONNECTION: The control flow with VSE/POWER when a BSC connection to another node is established, is shown in Figure 3-33 on page 3-78

PNET STOPPING A BSC CONNECTION: The control flow with VSE/POWER when a BSC connection to another node is terminated, is shown in Figure 3-34 on page 3-79

Stop a Connnection

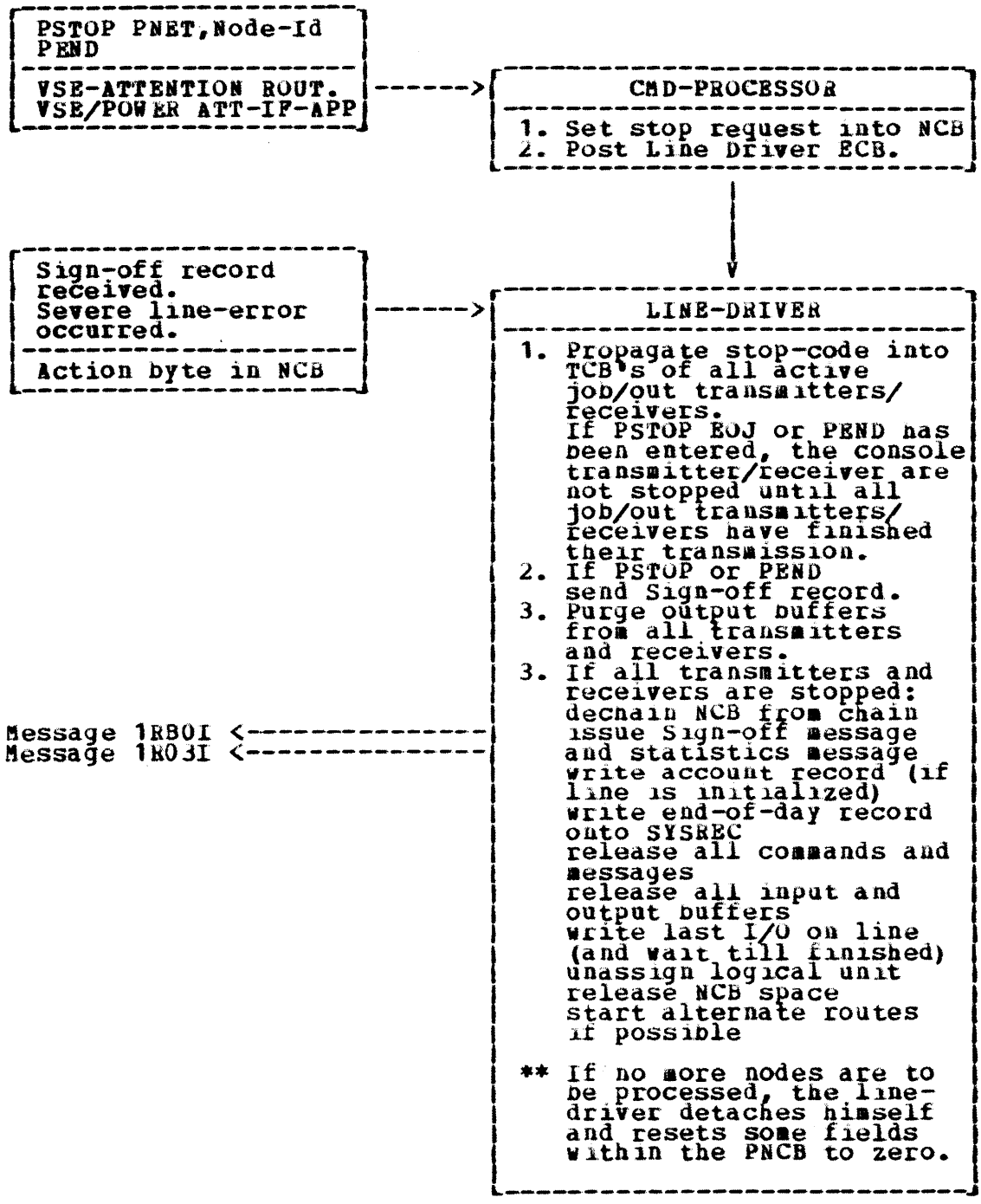


Figure 3-34. Control Flow When Stopping a PNET BSC Connection

is issued including the reason codes causing the failure. The reason codes are taken from the ACF/VTAM RPL RTNCD and FDBK2 fields. The ACF/VTAM interface is closed the PNET Driver ECB is posted and the subtask is detached from VSE.

SUCCESSFUL ENABLING OF THE ACF/VTAM SCIP EXIT.: Message 1RE0I is displayed to inform the operator that the ACF/VTAM interface has been opened successfully and that session requests can be sent to, and received from, remote nodes.

The VSE subtask IPW$$S1 posts the PNET driver ECB and issues a VSE/Advanced Functions WAIT. The subtask remains in the WAIT until it is posted by the PNET Driver to perform the disabling of the SCIP exit and to close down the ACF/VTAM interface.

Disabling the ACF/VTAM SCIP Exit.

The ACF/VTAM SCIP Exit is quiesced when no more session requests originating from remote nodes can be accepted, for example after a PEND has been issued by the operator.

The ACF/VTAM SETLOGON QUIESCE request is issued and the subtask IPW$$S1 enters a wait state again until it is posted in order to perform the CLOSE request.

ACF/VTAM SETLOGON QUIESCE FAILURE.: VSE/POWER could not quiesce the ACF/VTAM SCIP exit. i.e. session requests issued by remote nodes can still be received. Error message 1RD4I is issued including the reason codes causing the failure. The reason codes are taken from the ACF/VTAM RPL RTNCD and FDBK2 fields. The ACF/VTAM interface is closed and the the PNET Driver ECB is posted and the subtask is detached from VSE.

Closing the ACF/VTAM Interface

The subtask IPW$$S1 is posted by the PNET driver in order to close down the ACF/VTAM interface by means of the ACF/VTAM CLOSE macro. Message 1RE1I is issued after successful completion of the CLOSE request indicating that no more PNET SNA functions can be performed.

ACF/VTAM CLOSE FAILURE.: The ACF/VTAM interface could not be closed properly. Error message 1RD5I (including the reason code) is issued. The reason code is taken from the ACF/VTAM ACB Error Field. The PNET Driver ECB as well as the VSE/POWER Master ECB are posted and the subtask is detached from VSE.

PNET Session Establishment

The IPW$$S2 routine is invoked in order to establish a session between the local and the remote node either on behalf of a primary or of a secondary application program. It is attached as a VSE/POWER task by the PNET Driver.

The IPW$$S2 routine performs following functions :

Both nodes have to agree on the buffersize used during data transfer between the two nodes and the process is as follows:

- The node with the higher nodename sends the NJE Type 4 FM Header which contains the buffersize as defined in the NCB.
- The node with the lower nodename receives the NJE Type 4 FM Header and sends its own FM Header including the buffersize as defined in its NCB.
- The node with higher nodename receives the NJE Type 4 FM Header.
- Both nodes compare their own buffersize with the remote buffersize and the smaller of the two values is used for data transfer between the two nodes.

The IPW$$S2 task is detached after succesfull exchange of the buffersizes.

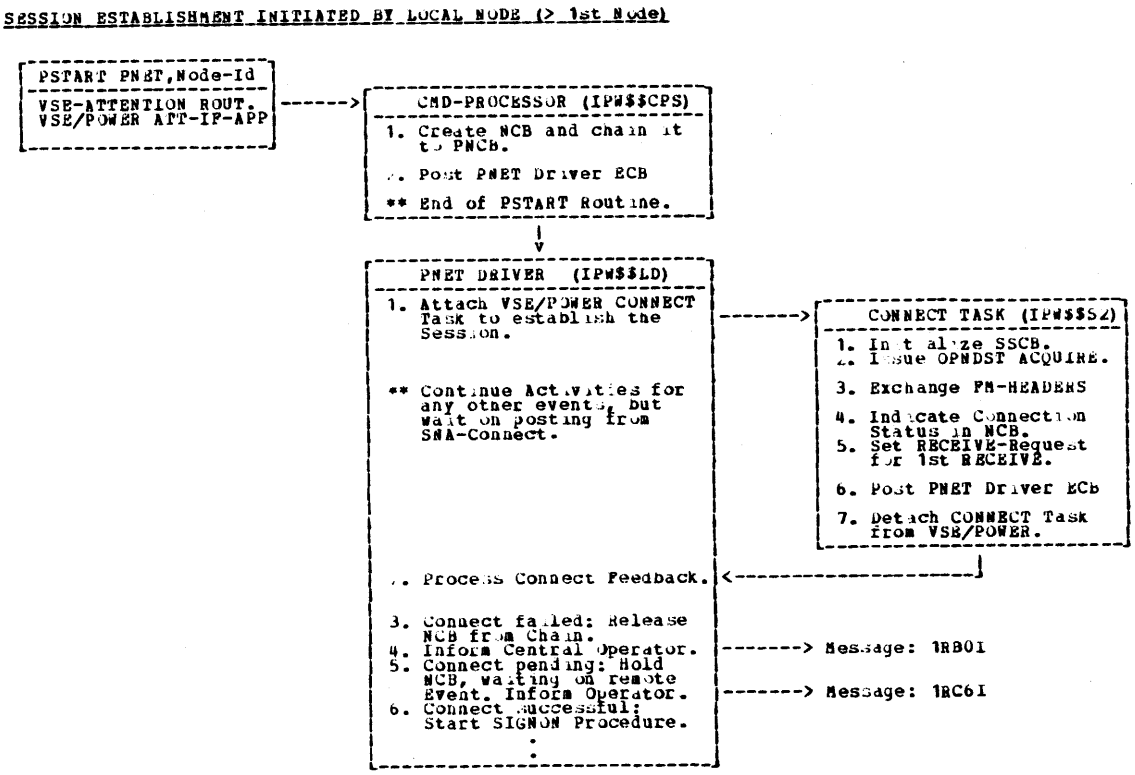


Figure 3-36. PNET SNA Session Establishment. For Non-first node.

ACF/VTAM OPNDST ERROR: Two cases have to be distinguished :

1. The NCB will be kept, which means that the local PNET system will be able to receive a session request from this node. Error message 1RD9I is displayed indicating the reason code and message 1RC6I is displayed indicating the the local node remains in a pending status.

2. The NCB will not be kept, which means that an unrecoverable error has occurred. Error message 1RD8I is displayed indicating the reason of the OPNDST failure.

SEND/RECEIVE ERROR DURING FM HEADER EXCHANGE : Two cases have to be distinguished :

1. An ACF/VTAM error has occurred which means that the exchange of the FM Headers has not been successful. Error message 1RD8I is displayed indicating the reason code and the session has to be terminated by the IPW$$S3 task.

2. Invalid data (incorrect or no FM Header) has been received. Error message 1RD9I is displayed indicating the reason of the failure. The session has to be terminated by the IPW$$S3 task.

REMOTE INITIATED SESSION ESTABLISHMENT

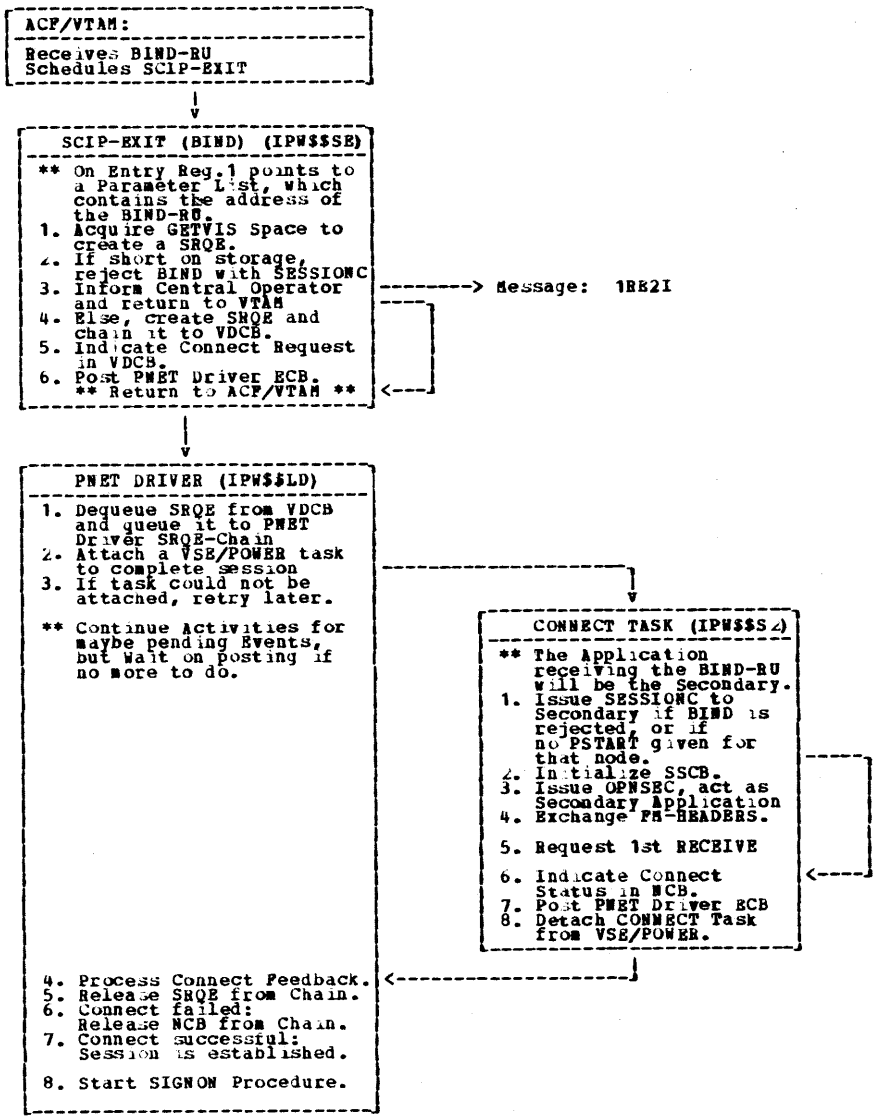


Figure 3-37. PNET SNA Remote Initiated Session.

ACF/VTAM OPNSEC ERROR: The NCB is not kept which means that an unrecoverable error has occurred. Error message 1RD8I is displayed indicating the reason of the OPNSEC failure.

A VARY NET,INACT,SID=applid,TYPE=FORCE should be entered in order to terminate the session.

Secondary Application Program Rejects Session Request.

The operator at the remote console has entered the

```
PSTART PNET,nodeid(,nodepassword)
```

command.

SSENSMO=X'21' Invalid session parameters.

This system sense information is displayed at the remote console with error message 1RD9I.

The IPW$$S2 task detaches itself from VSE/POWER after successful completion of the SESSIONC request. No session has been established.

Error message 1RD8I is displayed in case of SESSIONC failure. The still outstanding session request should be terminated by entering VARY NET,INACT,SID=nnnnnnnn,TYPE=FORCE.

PNET Session Termination

The IPW$$S3 routine is invoked in order to terminate a session which has been established by the IPW$$S2 routine. It is attached as a VSE/POWER task by the PNET Driver.

The IPW$$S3 routine performs following functions :

1. terminates a session in behalf of a primary application program by issuing the ACF/VTAM CLSDST request.

2. terminates a session in behalf of a secondary application program by issuing the ACF/VTAM TERMSESS request.

3. receives the session termination request from the primary application program and acts on behalf of a secondary application program. The IPW$$S3 routine receives the UNBIND command.

4. receives the session termination request from the secondary application program and acts on behalf of a primary application program. The IPW$$S3 routine receives the LOSTERM RC=20 condition.

Primary Application Program Terminates the Session

The operator at the local console enters a PSTOP PNET,nodeid command or another condition (like ACF/VTAM Halt) leads to a termination request. The general logic flow is shown in Figure 3-38.

The PNET Driver attaches the routine IPW$$S3 as a VSE/POWER private task. The task waits for the completion of the IPW$$S2 task which has established or which is just establishing the session. The necessary ECB is located in the NCB representing the session.

The ACF/VTAM request CLSDST is issued to disconnect the application programs located on the local and remote node. The CLSDST request causes an UNBIND command to be sent to the remote secondary application program (via the ACF/VTAM SCIP Exit) thus informing the secondary application program about session termination. The NCB is flagged to indicate that it can be freed by the PNET Driver.

The primary application program has to issue an ACF/VTAM CLSDST request so that the session is terminated. The secondary application program receives an UNBIND command for which it has waited after issuing the TERMSESS request.

The NCB is flagged that it can be freed by the PNET Driver. The PNET Driver ECB is posted and the IPW$$S3 task detaches itself from VSE/POWER The SNA Session Control Block (SSCB) is freed later by the PNET Driver.

Primary Application Program Receives Session Termination Request.

The operator at the remote console enters PSTOP PNET,nodeid command or another condition (like ACF/VTAM Halt) leads to a termination request. The general flow is shown in Figure 3-39.

The PNET Driver attaches the routine IPW$$S3 as a VSE/POWER private task. The tasks waits for the completion of the IPW$$S2 task which has established or which is just establishing the session. The necessary ECB is located in the NCB representing the session.

LOSTERM condition with reason code 20 has been passed to PNET through the ACF/VTAM LOSTERM Exit. A CLSDST request (as described above) has to be issued in order to terminate properly the session. The CLSDST request causes an UNBIND command to be sent to the remote secondary application program (via the ACF/VTAM SCIP Exit) thus informing the secondary application program about session termination. The NCB is flagged that it can be freed by the PNET Driver. The PNET Driver ECB is posted and the IPW$$S3 task detaches itself from VSE/POWER. The SNA Session Control Block (SSCB) is freed later by the PNET Driver.

freed by the PNET Driver. The PNET Driver ECB is posted and the IPW$$S3 task detaches itself from VSE/Advanced Functions. The SNA Session Control Block (SSCB) is freed later by the PNET Driver.

ACF/VTAM CLSDST OR TERMSESS FAILURE.: In case an ACF/VTAM error has been encountered during execution of the CLSDST or TERMSESS request, error message 1RD8I is issued including the reason code (which consists of the RPL RTNCD and FDBK2 codes. Error message 1RC4I will be displayed when ACF/VTAM suffers from temporary storage shortage and cannot perform the requested function.

The console operator should try to cancel the still existing session by using the VARY NET,INACT,SID=xxxxxxxx,TYPE=FORCE

PNET ACF/VTAM Exits

The module IPW$$SE contains the ACF/VTAM Exit Routines for the PNET SNA Support. The following exits are supported:

SCIP
NSEXIT
LOSTERM
TPEND

The exits are required to handle the special events scheduled by ACF/VTAM.

The exit routines run under control of the VSE/Advanced Functions subtask under which the ACF/VTAM OPEN was performed. During OPEN time the exits are enabled to ACF/VTAM via the SETLOGON command issued by module IPW$$S1.

Because the exits run under the control of the VSE/Advanced Functions subtask and cannot share VSE/POWER resources, all events which cause an exit to be scheduled are passed to the PNET Driver by setting various indications in control blocks. The following exit routines are supported and perform the following functions.

SCIP Exit

The SCIP exit is scheduled in the following circumstances:-

- BIND Request Unit

  As part of the session establishment an OPNDST command is executed which triggers the sending of a BIND-RU to the other end of the session. The received BIND-RU causes ACF/VTAM to schedule the SCIP-Exit with a parameter list containing the address of the received BIND-RU.

  The BIND-RU is used to form a Session Request Element (SRQE) in the GETVIS Space of the VSE/POWER partition. The SRQE is chained to the PNET master control block (PNCB) and the PNET Driver is informed that a remote session request has arrived.

The following situations cause the LOSTERM Exit to be scheduled:

20 The secondary end issued a TERMSESS... TYPE=UNCOND Command to terminate the session. ACF/VTAM at the primary side schedules the exit. In this case, the exit informs the PNET Driver to terminate the session immediately.

32 Network Operator initiated conditional terminate.

This event causes the PNET Driver to be informed that a conditional terminate should take place, which implies that all transmitter and receiver tasks may continue until end-of-job, before the session is terminated.

12 Network Operator initiated ACF/VTAM HALT.

This event causes an immediate termination of ACF/VTAM. The PNET Driver is informed that ACF/VTAM is terminating immediately and propagates an immediate termination to all PNET SNA sessions.

36 ACF/VTAM Buffer Limit exceeded.

If the buffer limit defined by the NCP Generation is exceeded, the exit is scheduled. This event causes immediate termination of the session and the PNET Driver is informed about this event.

The operator is informed with message 1RD7I that the LOSTERM exit has been scheduled.

NSEXIT Exit.

The CLEANUP-RU is the only type of RU which is supported in this exit. A CLEANUP-RU is created by ACF/VTAM under the following circumstances: The general flow is shown in Figure 3-40.

```
Operator initiated command:
VARY NET,INACT.....
VARY NET,TERM,SID=nnnnnn,TYPE=FORCE or
Unexpected CLOSE issued from application.
```

The received CLEANUP-RU leads to an immediate termination of the session. The exit informs the PNET Driver about this situation.

PNET SEND/RECEIVE Function

The SEND and RECEIVE functions of the PNET-SNA support are located in the module IPW$$SR. This module also contains the SEND/RECEIVE exit routines for asynchronous processing. The functions are called either:

- by the PNET Driver, in which case the routines run under control of the calling task TCB acting as a function module.

  - The PNET Driver calls the SEND function by means of the IPW$IOM macro instruction when at least one output buffer has been queued by the PNET Driver into the 'TO-BE-SENT AHEAD' queue and no SEND is currently in progress.

  - The PNET Driver calls the RECEIVE function by means of the IPW$IOM macro when at least one input buffer has been queued by the PNET Driver to the 'FREE INPUT AHEAD' queue and no RECEIVE is currenlty in progress.

- by the SEND/RECEIVE exit routines, which are scheduled by ACF/VTAM after final completion of the request, by means of the IPW$IOM macro instruction.

  Note: Both the exits as well as the invoked SEND/RECEIVE functions run under control of the PNET SNA subtask (IPW$$S1).

The following technique is used to prevent concurrent SEND/RECEIVE being executed:

If the SEND or RECEIVE functions are called from the PNET Driver, the function gates are locked and remain locked until no more buffers are available to send or to receive requests or responses. Due to asynchronous processing, the SEND/RECEIVE exits are scheduled by ACF/VTAM at final completion time. The currently held input or output buffer is queued to the PNET Driver 'BUFFER QUEUE' for later processing. The SEND or RECEIVE exit then initiates a new SEND or RECEIVE request by executing the IPW$IOM macro. The appropriate function gate remains locked unless an error is detected or no buffer can be sent or received.

Two additional queues are introduced, for performance reasons, in the SNA processing:

- To be sent ahead queue
- Free input ahead queue

The queues provide the capability to initiate another SEND/RECEIVE from the appropriate exit which runs under the TIK of the PNET SNA subtask. The normal input/output buffer queues are maintained by buffer service using the VSE/POWER resource management which is not available when running as VSE/Advanced Functions subtask.

The 'ahead' queues are filled by the PNET Driver on demand of the SEND/RECEIVE function invoked by the exits. Once a SEND/RECEIVE has been initiated by the PNET Driver, the next SEND/RECEIVE is automatically issued by the exit when the previous has finally completed. This process continues until no buffers are available or an error has been encountered.

SNA SEND FUNCTION

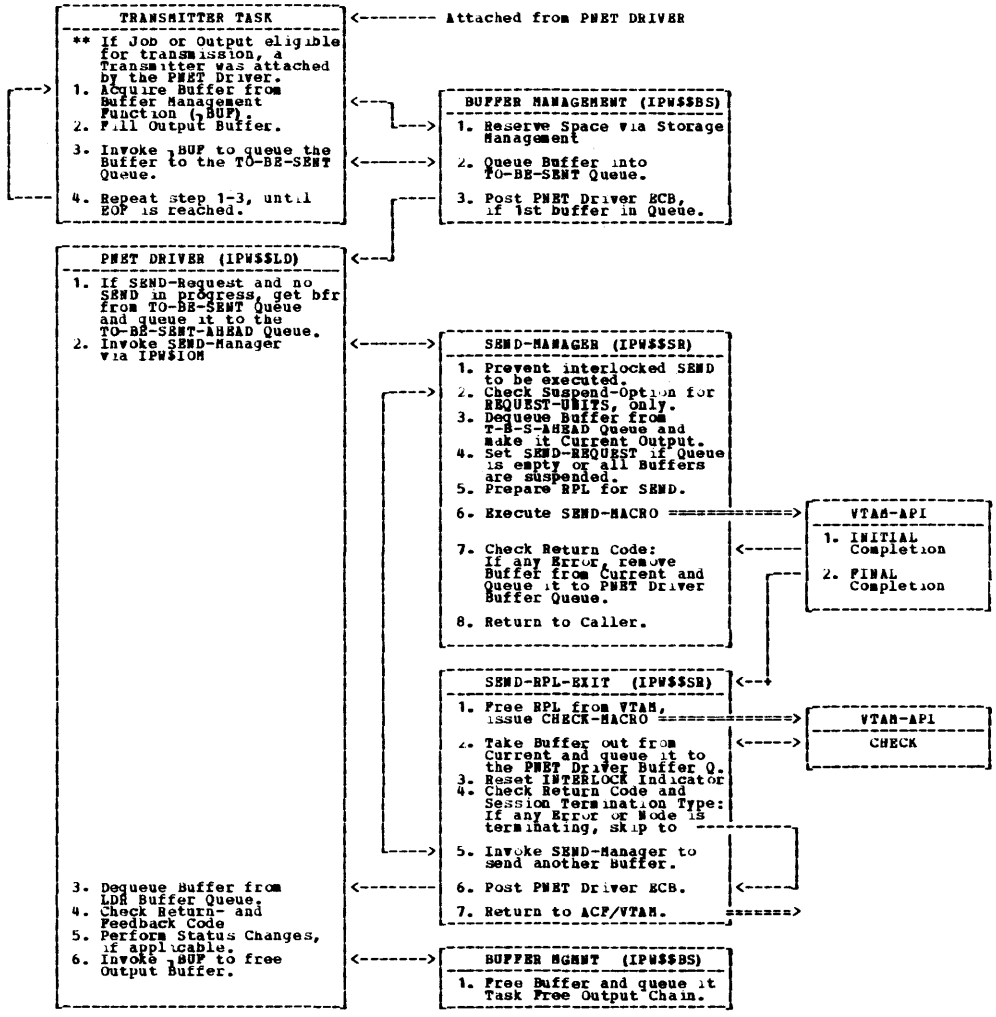


Figure 3-41. PNET SEND Function

RECEIVE FUNCTION: The RECEIVE-Function checks the RECEIVE-Gate and returns to the caller with return code X'FF' in register 15 if the gate is already locked. Otherwise the gate is locked to prevent more than one RECEIVE being executed at a time. The FREE-INPUT AHEAD Queue is examined for a free buffer. If no buffer is available, the PNET Driver is informed by setting the 'RECEIVE-Request' in the NCB, the gate is opened and return is made to the caller. Input buffers may contain a pregenerated RPL which is modified when executing the RECEIVE macro. If no RPL is available, the pregenerated RPL of the SSCB is moved into the buffer and modified when executing the RECEIVE macro.

The initial return code is checked and if no error is indicated return is made to the caller. If the initial completion fails, the input buffer is chained to the PNET Driver buffer queue for further processing, the gate is unlocked, and return is made to the caller.

The general flow is shown in Figure 3-42.

a free buffer and if one is available the RECEIVE-Function is called, from the exit, to start another RECEIVE. The gate remains locked until any error is detected or until no input buffer is available. In all cases, the PNET Driver task is posted to perform the required actions for the queued buffer.

PNET Transmitter

The PNET transmitter consists of two different transmitters, the job or output transmitter and the console transmitter.

Job/Output Transmitter

The PNET Transmitter is responsible for transmitting jobs or output to other nodes in the network. For each active transmitter there must be an active receiver on the other end of the line. Figure 3-43 .

1. Initially when the connection between the two nodes is established and sign-on processing completed, then one job and one output Transmitter are attached.

2. When a job or output eligible for transmission is put into the XMT queue and the final destination is reachable via this connected node.

3. When a PACT command is issued for the transmitter.

4. When a job or output is returned to the XMT-queue with its original disposition by means of the IPW$DQS HOLD macro. The transmitter going to be drained checks if there are other inactive transmitters of the same type and if so attaches one.

5. When a connection between two nodes is broken, then the first inactive transmitter for an alternate route node is attached.

Before attaching the transmitter the PNET Driver ensures that the connection is not drained (either done by PSTOP PNET,nodeid,EOJ or PEND command given by the central operator) and equips the transmitter with a workarea.

When the job or output transmitter is initially entered, the workarea previously acquired by the PNET Driver is initialized. The IPW$NQS macro is then executed to obtain the next eligible queue set (job or output, dependent on the type of transmitter). The 'Get Next Queue Set' routine scans the XMT queue for a job or output which the transmitter can send and selects the oldest highest-priority job or output queue entry which is destined to, or routed via, the node which the transmitter is serving.

A particular transmitter may be eligible to send to only a few of the nodes in the network. Eligibility is determined by the 'Get Next Queue Set' routine, which determines the best path to reach any node.

If the 'Get Next Queue Set' routine returns to the transmitter without selecting a queue entry, the transmitter frees the storage area used as workspace, informs the PNET Driver about the termination and detaches itself. If however, a queue entry was selected, the transmitter initializes the Account record with the transmission start time, date and information extracted from the queue record. If the previous transmission of a job or output was aborted, the IPW$STM macro is issued to allow the receiver on the other end of the line to do its cleanup processing before the next request for transmission (RIF) is sent.

The Get Data record routine is invoked via macro IPW$GDR to obtain the first data record (the job header record). The Composer is called to generate a request for permission to transmit (RIF) and to schedule the buffer for transmission.

If permission is not granted (the PNET driver has set the immediate stop code in the TCB on receipt of the negative permission) the transmitter informs the central operator via message 1RA9I and the IPW$DQS HOLD macro is issued to return the queue entry to the XMT Queue with its original disposition.

The transmitter task entry table, contained in associated NCB, is scanned for any inactive transmitter of the same type. If found, the PNET Driver is posted

The Composer on behalf of the transmitter waits then for an acknowledgement (receiver cancel from the other side).

The cancel code is stored in the account record. The operator is informed via message 1RA9I, and in the case that the transmitter will also be drained, via message 1RA8I. The queue set is returned to the XMT queue with a disposition requested by the saved stop state. If the transmitter is to be drained, then the PNET driver is requested to start another not drained transmitter. If applicable, additional information is written into the account record. The logical data area and queue record area are then released.

If the saved stop state does not require an immediate and unconditional termination of the transmitter task, any subsequent transmission will start again from the beginning.

Console Transmitter

The Console transmitter associated with a node is responsible for sending messages and commands, in the NMR format, to that node. The task is attached by the PNET Driver when a message/command is put in the message/command queue of the node concerned.

When the Console transmitter is initially entered, the routine initializes the work area which was acquired by the PNET Driver. The IPW$ICS REQ=GET macro is then issued to obtain the first message/command from the queue. The NMR is then passed to the Composer which converts the record into the form required for transmission, including compression and blocking. After passing the record, the storage area occupied by the NMR is released by means of the IPW$ICS REQ=DEL macro instruction.

This process continues until all messages/commands are scheduled for transmission. When the message/command queue is empty, the Composer is called to schedule the partially filled buffer for transmission. The Composer returns to the transmitter when the last buffer has been transmitted. If, meanwhile, another message/command was put into the queue associated with the node, the Console transmitter starts the whole process again. Otherwise the routine informs the PNET Driver about the termination and, after releasing all prior acquired workarea, detaches itself.

PNET Composer

The Composer runs under the control of the calling task and acts as a function module. It is responsible:

- to set up an MLI control record (eg. RIF, PGR) and to schedule the record for transmission.
- to prepare all records (eg. data records, nodal message records, job header records etc.), with the exception of topology records, in the format required for transmission. Framing the records with RCB and SRCB for BSC or RID for SNA respectively, compressing the record and blocking.

Normal Record (Not spanned)

	<----------RECL------------->	
RECL	CCTL*	Data
1 byte	1 byte	

Spanned Record

.		<------SEGL1------>				<-SEGL2->
TRECL	SEGL	CCTL*	Data....	--------	SEGL	Data....
2 bytes	1 byte	1 byte			1 byte	

RECL Length of data in unspanned record

CCTL Command code

TRECL Total data length of segmented record

TRECL=SEGL1+SEGL2+.....SEGLn

SEGL Length of data in segment

* means that this field is optional.

Figure 3-44. Format of Output Records

NETWORK CONTROL RECORD PROCESSING: Network control records are segmented into records with a maximum length of 256 bytes. Each segment is provided with a 4-byte header containing a sequence count and a continuation indicator. The first segment of each network control record must be the first record in the output buffer. The buffer may then contain other records after the control record.

MLI CONTROL RECORD PROCESSING: All MLI control records, with the exception of ABORT, are built in small output buffers of fixed length and of a size capable of holding any type of MLI record. The buffers are obtained by means of

```
IPW$BUF MODE=OUT,TYPE=CNTRL
```

The Receiver calls the presentation service routine, which is responsible for obtaining physical buffers from the received input queue pertaining to the receiver, de-blocking and de-compressing each record and passing the logical record, on demand, to the receiver.

The receiver is attached by the PNET Driver when the corresponding transmitter on the other end of the line requests permission to transmit by sending a 'request to initiate a function' (RIF) control record. If, however, either VSE/POWER is in shutdown period or the node is to be terminated, the PNET Driver rejects that request by sending a negative permission (NPGR) control record.

When the receiver is initially entered, the workarea already acquired by the PNET Driver is initialized. The receiver then checks if any termination code has been set in the TCB (for example the operator has drained the receiver) and if so, the receiver terminates by calling the composer in order to send a negative permission to the corresponding transmitter. Otherwise the Composer is called to prepare a 'permission granted' (PGR) control record and to schedule the record for transmission. After the buffer has been successfully sent, the Composer returns to the receiver which then calls Presentation service in order to obtain the first logical record.

If Presentation service returns with a bad return code, the receiver aborts the current data stream. Otherwise the record control byte (RCB) and sub-record control byte (SRCB) which are returned by presentation service, are checked to determine what type of record was received (job header, job trailer, data set header, end-of-file, abort, normal data record, or nodal message record). The layout of the RCB can be found in Figure 3-50 and that of the SRCB in Figure 3-51.

The received record type is checked against the record type associated with the receiver; only records pertaining to the receiver and nodal message records are expected. If the received record was not expected, the receiver aborts the transmission.

If a network user exit is present, any network control records (job header, job trailer and data set header) as well as any JECL and JCL statement are passed to the exit before the appropriate routine is called which handles the type of record that was received. If, however, the user exit indicates that deletion of the record is required, then the receiver branches to obtain the next record from presentation service.

Note: Network Control records may not be deleted by the user exit.

JOB HEADER RECORD PROCESSING: A job header record is expected only once per job and must be the first record received. If such a record is received in the middle of a transmission, the receiver aborts the transmission.

When a job header record is received, the receiver prepares to receive the job/output.

- The IPW$RQS macro is issued to reserve a queue record and its associated datafile space.

record to the queue record. If a VSE/POWER section is present, the 3540 cuu address is also copied into the queue record and the TCB is flagged that now 3540 data records follow.

The JES2 output transmitter may send more than one data set header record before sending the data records. In this case, each data set header record may indicate a different destination for the same data set. The receiver splits each data set into multiple queue entries (data sets) - one for each destination. This means that when the data stream is transmitted again, it is sent as multiple streams, each containing the data set and each independant of the other.

For every data set header record so received, a data set control block (DSCB) is built, including queue record allocation and reserving of data block space. The DSCB is placed in the active DSCB chain. Each subsequent data record is now spooled for each queue entry in the active DSCB chain. The DSCB contains all required spooling information for the queue and data-file.

If another data set header record is received after data has been received, then the receiver suspends the spooling of all queue entries currently in the active DSCB chain by writing a NOP-record with the end of block indication and releasing the data block buffer afterwards. All data sets in the active DSCB chain are then placed in the suspended DSCB chain.

The suspended DSCB chain is then scanned for a match with the new data set header record. A match is found when the output characteristics such as forms id, FCB, destination, etc. , the output class and priority are identical.

If such a DSCB is found, which means that there already exists a incomplete queue entry which can be further used, the DSCB is removed from the suspended DSCB chain and put into the active DSCB chain. Additionally, buffer space for the data block is reserved and anchored to the DSCB.

If no DSCB is found, a new DSCB is created and initialized with the information from the job header record and the just received data set header record. A new queue record is reserved by means of the IPW$RQS macro and storage for the data block is obtained. Both the queue record and the data block are anchored to the DSCB being built. The job header record and the data set header record are then written as first records on spool.

DATA RECORD PROCESSING: When receiving a job, the record is written on spool by issuing the IPW$PDR macro. The account information is updated accordingly.

When receiving print data, each ASA character is converted into machine control characters. When receiving punch data, each ASA character is converted into X'21' (write, feed, select stacker 1). The record, or in case of ASA conversion the 2 records, are spooled for each queue entry described by a DSCB entry in the active DSCB chain. The account record is updated accordingly.

NODAL MESSAGE RECORD PROCESSING: The record is passed to the message distributor for further processing by issuing the IPW$GMS TYPE=DIST macro instruction.

JOB TRAILER RECORD PROCESSING: After all data has been transmitted, the transmitter sends the job trailer record. When the receiver receives the job trailer record it starts termination processing of the job or output. The job trailer is

When a receiver cancel record is sent, the receiver enters a data stream loop, waiting for an abort or end-of-file record from the transmitter. Each record received while in the this condition is purged, unless it is a nodal message record which is passed to the message distributor via the IPW$GMS TYPE=DIST macro instruction. The purge loop is left when either the PNET Driver encounters a line error or a signoff record is sent or received. In this case the PNET Driver propagates the error condition to the receiver. Any workspace previously acquired, such as the job header record area or the alternate presentation buffer area, are released and returned to the VSE/POWER storage pool.

If applicable, message 1RA8I is issued to inform the operator that the receiver is now drained. Any buffer still in the receiver input buffer queue is released and its storage returned to the VSE/POWER storage pool. Finally the account record is completed with the transmission stop time and the account record is written by means of the IPW$PAR macro instruction. The cancel code is set to X'50'. The PNET Driver is informed that the receiver is going to be detached.

PNET I/O Manager

The I/O Manager is responsible for issuing all I/O's for a PNET BSC line using the MLI line discipline. Whenever an I/O is to be initiated, the PNET Driver requests this by indicating the type of request in the node control block (NCB) addressed by R1.

Depending on the request code a channel program is constructed in the node control block and the I/O is started via a SVC 0. In addition the block control byte (BCB) sequence count is updated accordingly and the just sent buffer is completed with the current FCS bytes. No wait is performed after theSVC0 andimmediate return is made to the caller.

The request code consists of a main request (first four bits) and a sub request (last four bits). Figure 3-46 shows the various CCW sequences that can be built by the I/O Manager.

X'10' write / read sequence

- write text / empty buffer
- write DLE ETB
- read response / text
  The current status of the to-be-sent output queue is examined and depending on its condition either an empty buffer is sent, the next output buffer transmitted or an 1,5 second delay initiated. If a wait-a-bit was just received an empty buffer is sent to acknowledge the received input buffer. If a free input buffer is available or a new buffer can be obtained from the VSE/POWER storage pool this buffer is used as input buffer and the READ CCW is updated accordingly. Otherwise, the PNET Driver input buffer is used and the wait-a-bit flag is set.

X'20' NAK sequence

- write NAK
- read response / text
  If a free input buffer is available or a new buffer can be obtained from the VSE/POWER storage pool this buffer is used as input buffer and the READ CCW is updated accordingly. Otherwise the PNET Driver input buffer is used.

X'21' ACK0 sequence

- write ACK0
- read response / text
  If a free input buffer is available or a new buffer can be obtained from the VSE/POWER storage pool, this buffer is used as input buffer and the READ CCW is updated. Otherwise the PNET Driver input buffer is used.

Figure 3-46 (Part 2 of 2). Networking CCW Sequences

PNET Presentation Service

This routine is responsible for obtaining physical buffers from the 'received input queue' and passing decompressed logical records one at a time to the receiver. The Presentation service routine runs as a function module under the TCB of the calling task.

The logical record is decompressed in the presentation buffer, which has the same size as a data buffer (DBLK) and is appended to the Presentation service workarea passed by the caller. The decompression routine in the IPW$$NK module is used for the decompression of the records.

ForSNA,eachdatarecordorsegment in the buffer (RU) is preceded by a 3 byte record identifier (RID) appended to the record during compression at the transmitting node. The RID identifies the type of record and contains the length of the decompressed data record. Since the RID has undergone compression at the transmitting node, it must go through decompression at the receiving node. Thus the decompression routine is called to decompress a RID into a 3-byte workarea in

byes in length) for transmission. If the header/trailer record occupies more than one logical record, the presentation service routine rebuilds the entire header/trailer before passing it to the receiver. As each piece is received its sequence number is verified and if an error is encountered a return code is passed to the caller.

Additionally, the general section is checked for the minimum required length. If the section is too short, the routine expands it to the required length by moving down any data following the general section and padding the section with binary zeros. If the section is expanded, both the overall length and the section length are updated. This is necessary, because the general section may be enlarged from one release to another and not all nodes within a network may have the same release level.

If normal or abnormal end-of-file is received, appropriate flags are set in the TCB of the calling receiver and immediate return to the caller is made.

If the buffer is processed (all logical records have been passed to the receiver), the buffer is returned to the free input queue by issuing the IPW$BUF MODE=IN,TYPE=FREE macro instruction.

PNET Buffer Service

This routine is responsible for queueing all incoming buffers to the 'received input queue', to queue all outgoing buffers in the 'to-be-sent queue', and to supply buffers for both the transmitters and receivers. The buffer service routine runs under control of the calling task and acts as a function module. The routine is called by means of the IPW$BUF macro instruction.

The buffers required to process BSC nodes are provided from real storage, while those required to process SNA nodes are provided from virtual storage (GETVIS).

For a detail description of the functions provided, see Appendix D, "Format of Internal Macros" -"IPW$BUF - PNET Buffer Management" on page D-1.

PNET Compression/Decompression

The PNET compression function is used to condense duplicate character strings to reduce tele-processing transmission volume and thus transmission time. In the case of three or more duplicate non-blank characters, the string is replaced with two bytes: a 'String Control Byte' (SCB) followed by the character itself. Two or more duplicate blank (X'40') characters require only the SCB indicating the character string length. Strings of non-duplicate characters are also preceeded by a SCB to indicate the length. The decompression function uses the SCB's to reconstruct the original data.

The SCB's and the method of compression/decompression differ according to the caller's indicated TP line discipline (BSC or SNA). For compressed BSC data only, each record ends with a special end-of-record SCB. Therefore BSC compression appends this SCB to each compressed data string offered as input.

Multi-Leaving Format

The basic element for multi-leaving transmission is the character string. One or more character strings are formed from the smallest external element of transmission - the logical record. For efficiency in transmission, each record is reduced to a series of character strings of two basic types:

- a variable length non-identical series of characters
- a variable number of identical charaters

Because of the frequent occurrence of blank characters, a special case is made for identical characters when the duplication character is a blank. A 1 byte control field, called a string control byte (SCB), precedes each character string to identify the type and length of the string. Thus a string of nonidentical characters is represented by an SCB followed by the nonduplicate characters. A string of consecutive, duplicate, nonblank characters is represented by an SCB and a single character. The SCB indicates the duplication count and the character following indicates the character to be duplicated. In the case of an all-blank character string, only an SCB is required to indicate both the type and number of blank characters. Figure 3-49 describes the supported SCB's.

Binary	Meaning
0000 0000	End-of-record If first SCB, this also indicates end-of-file
0100 0000	Abort transmission
100b bbbb	"bbbbb" blanks are to be inserted
101d dddd	The single character following this SCB is to be duplicated "ddddd" times
11cc cccc	The "cccccc" characters following this SCB are to be inserted

Figure 3-49. String Control Byte (SCB)

A data record to be transmitted is therefore segmented by the transmitting program into the optimum number of characters to take full advantage of the identical character compression. A special SCB is utilized to indicate the grouping of character strings which compose the original logical record. The receiving program can then re-construct the original record for processing.

In order to group logical records together in a single transmission block, an additional 1-byte control field precedes the group of character strings representing the original logical record. This field, the record control byte (RCB), identifies the general type and function of the logical record (input stream, print stream). A particular RCB type has been designated to pass control information between the various systems. To provide for simultaneous transmission of similar functions (such as multiple input streams), a stream indentification code is included in the RCB. Figure 3-50 shows the various supported RCB's.

RCB	SRCB
00	None
80	RCB of job/output stream to be held or X'FF' for all streams to be held
90	RCB of function to be initiated
A0	RCB of function to be initiated
B0	RCB of function to be cancelled
C0	RCB of function which is complete
D0	RCB of job/output stream to be released X'FF' for all streams to be released
E0	Expected count (received count is in BCB)
F0	An identification character as follows: A = Initial RJE SIGN-ON B = Final RJE SIGN-OFF C = Print initialization record D = Punch initialization record E = Input initialization record F = Data set transmission initialization G = System configuration status H = Diagnostic control record I = Initial network SIGN-ON J = Response to initial network SIGN-ON K = Reset network SIGN-ON L = Accept (concurrence) network SIGN-ON M = Add network connection N = Delete network connection O-Z = Reserved for future use
91	1000 0000 (X'80')
92	1000 0000 (X'80')
93-F3	1000 0000 (X'80')
94-F4	Carriage control information as follows: 1010 00nn - Space immediately "nn" spaces 1011 cccc - Skip immediately to channel "cccc" 1000 00nn - Space "nn" lines after print 1000 1100 - Load printer FCB image 1001 cccc - Skip to channel "cccc" after print 1000 0000 - Print and suppress space
95-F5	1000 1111 (X'8F')
96-F6	Undefined
97-F7	Undefined

Figure 3-51 (Part 1 of 2). Sub-Record Control Byte (SRCB)

Binary	Meaning
r... .... r...	Reserved (must be 1... .... 1... ....)
.0..	Normal state
.1..	Suspend all stream transmission
..rr .... ..rr	Reserved for future use
.... .1..	Remote console stream identifier
.... 1...	Function stream identifier for: NJE job transmission stream number 1
.... .1..	Function stream identifier for: NJE job transmission stream number 2 NJE SYSOUT transmission stream number 7
.... ..1.	Function stream identifier for: NJE job transmission stream number 3 NJE SYSOUT transmission stream number 6
....1	Function stream identifier for: NJE job transmission stream number 4 NJE SYSOUT transmission stream number 5
.... 1...	Function stream identifier for: NJE job transmission stream number 5 NJE SYSOUT transmission stream number 4
.... .1..	Function stream identifier for: NJE job transmission stream number 6 NJE SYSOUT transmission stream number 3
....1.	Function stream identifier for: NJE job transmission stream number 7 NJE SYSOUT transmission stream number 2
....1	Function stream identifier for: NJE SYSOUT transmission stream number 1

Figure 3-52. Function Control Sequence (FCS)

For error detection and correction purposes, a block control byte (BCB) is added as first character of each block transmitted. The BCB, in addition to control information, contains a modulo 16 block sequence count. This count is maintained and verified by both the sending and receiving systems to prevent lost or duplicated transmission blocks. The layout of the BCB can be found in Figure 3-53.

An RU consists of as many logical record identifiers and corresponding logical records as will fit in the specified RU size as specified by the BUFSZ parameter. No logical record may be larger than 256 bytes plus 3 bytes for the record identifier(RID). PNET sends only one type of record (network topology, stream control or data record) within a SNA RU. PNET compresses an entire RU, i.e. everything from the beginning to the end of the RU, without regard to record identifiers.

DLE	BSC Control Character
STX	BSC Control Character
BCB	Block Control Byte
FCS	Function Control Sequence
FCS	Function Control Sequence (Continued)
RCB	Record Control Byte for Record 1
SRCB	Sub-Record Control Byte for Record 1
SCB	String Control Byte for Record 1
data	Character string
SCB	String Control Byte for Record 1
data	Character string
SCB	Terminating SCB for Record 1 (end-of-record)
RCB	Record Control Byte for Record 2
SRCB	Sub-Record Control Byte for Record 2
SCB	String Control Byte for Record 2
SCB	Terminating SCB for last record
RCB	Transmission Block Terminator (end-of-block)
DLE	BSC Control Character
ETB	BSC Control Character

Figure 3-55. Multi-Leaving Buffer Format for BSC

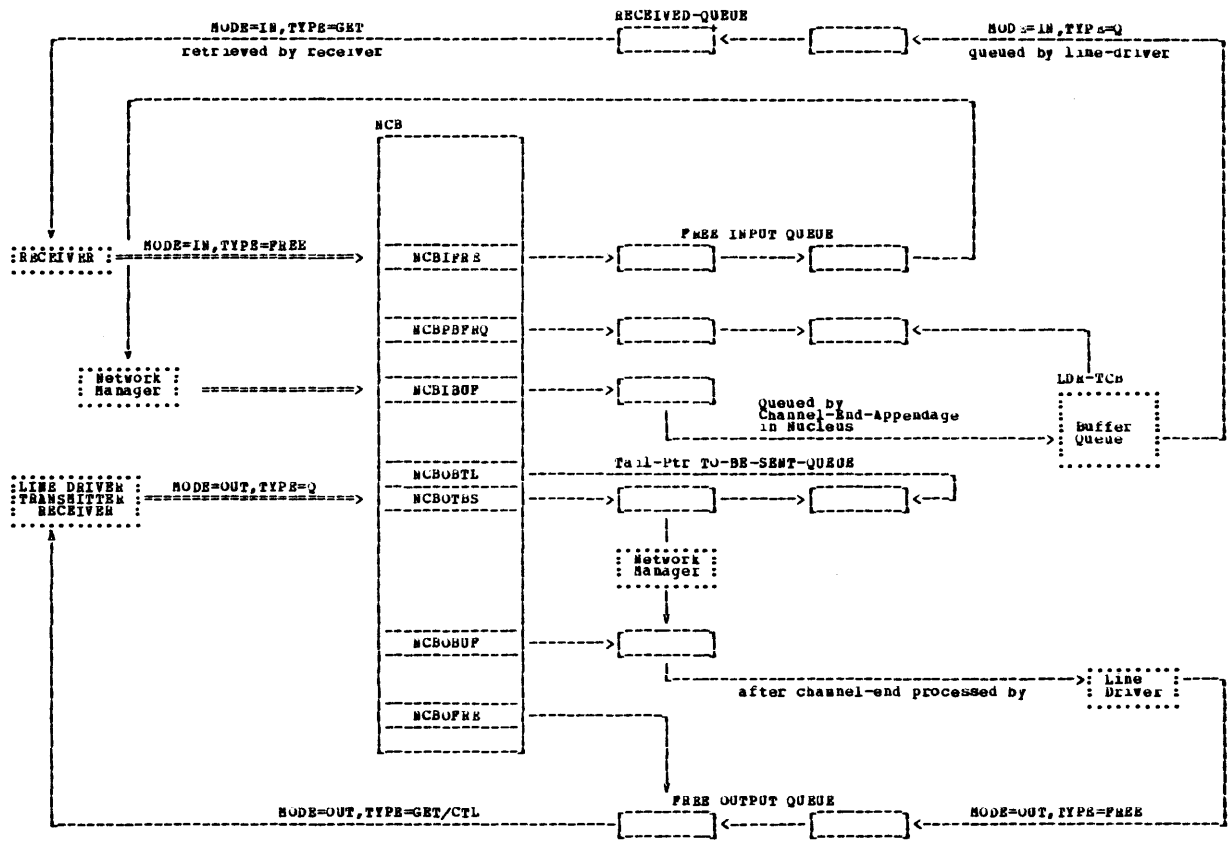


Figure 3-57. PNET BSC Buffer Relationship and Queueing

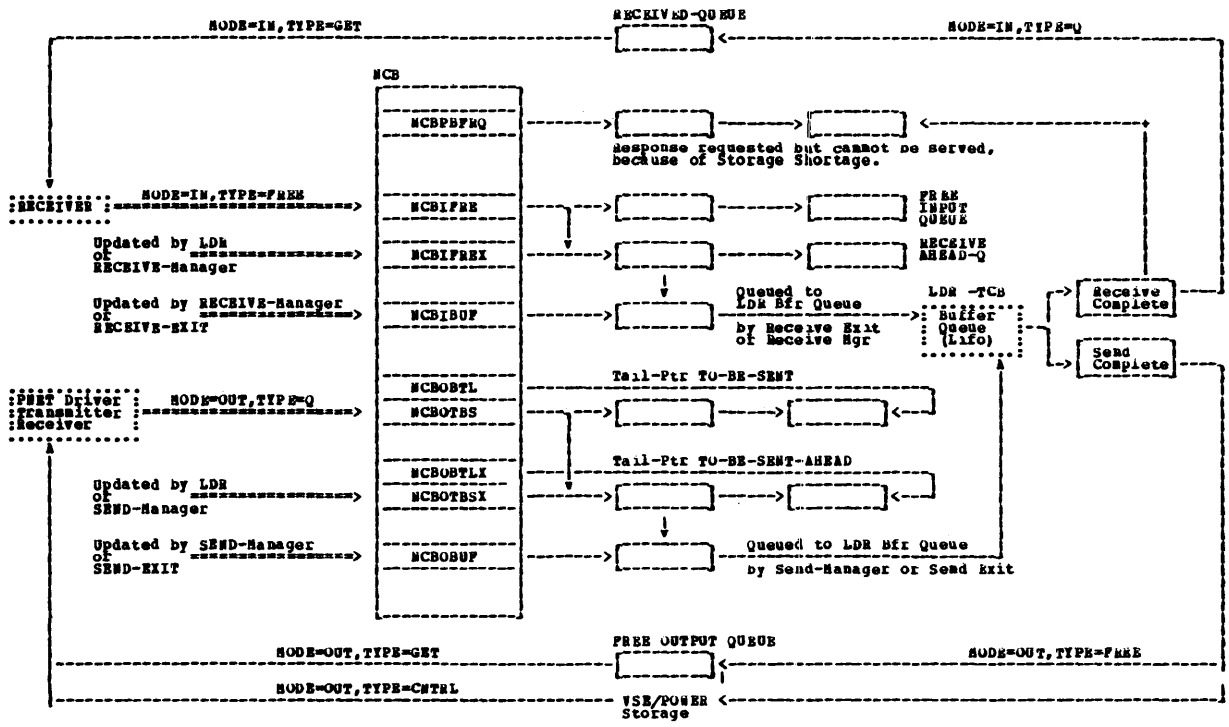


Figure 3-59. PNET SNA Buffer Relationship and Queueing

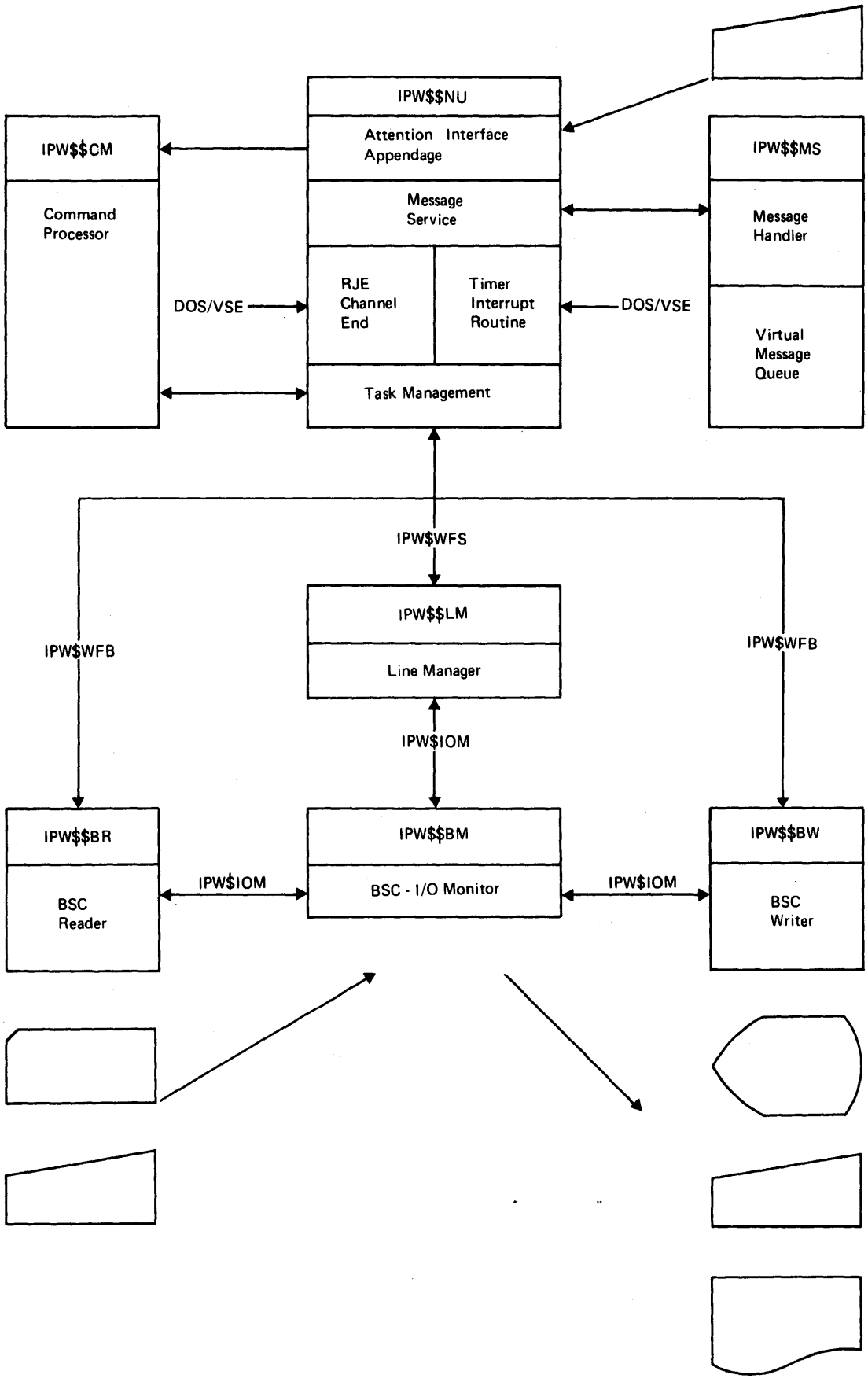
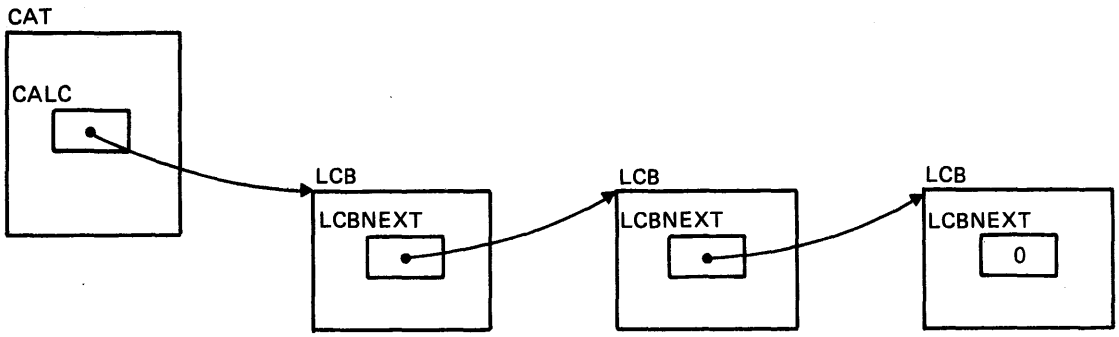


Figure 3-60. RJE,BSC Relationship

LCB Scanning:



Channel End Processing:

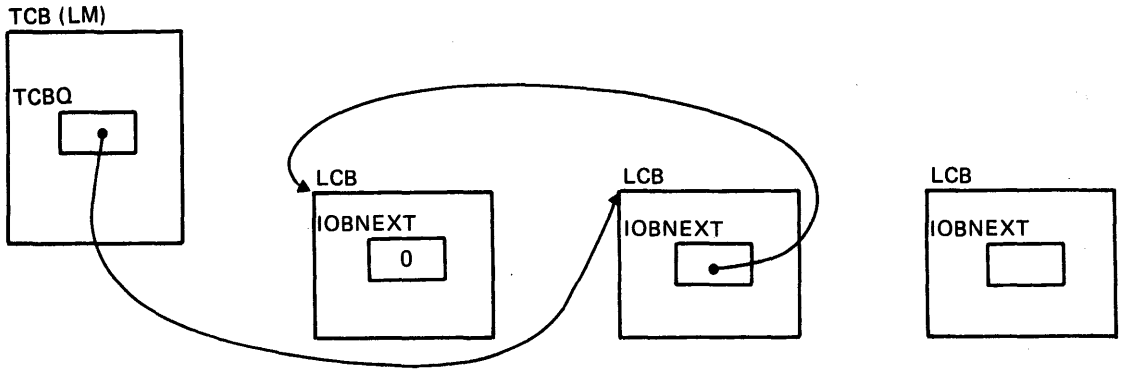


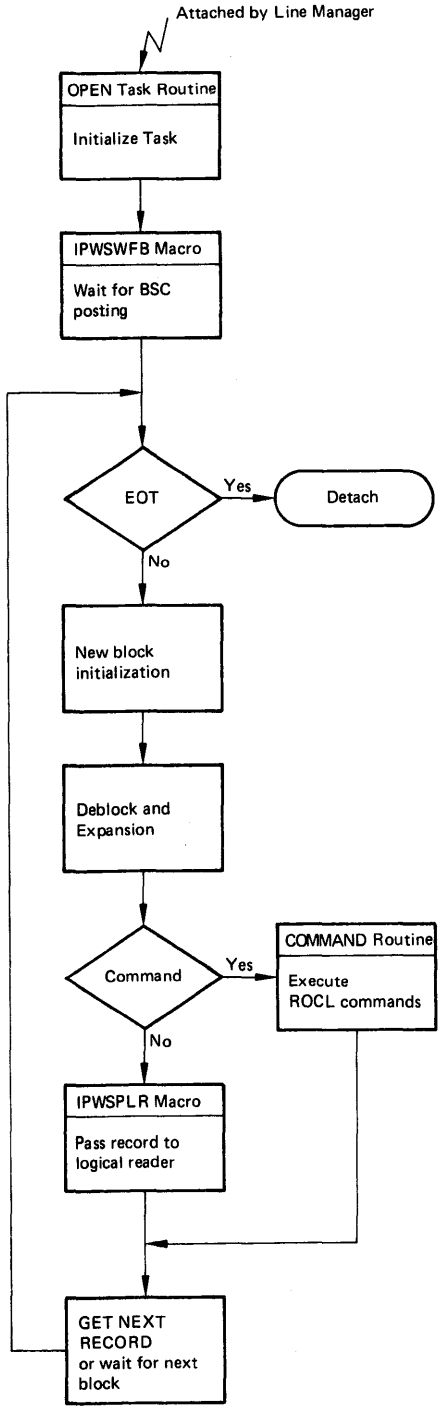
Figure 3-61. LCB Activity Checking and Channel End Processing

When a line start request is detected during LCB activity checking, two initial CCW chains are set up.

- Leased line
  - DISABLE
  - SET MODE
  - ENABLE
  - PREP
  - READ
- Switched line
  - DISABLE
  - SET MODE
  - ENABLE
  - NOP
  - READ

state until it is posted by the line manager when the first block has been received. The data records are deblocked and expanded. An RJE,BSC command, read at job boundary is either processed by the RJE,BSC command processor routine or a temporary command processor is attached. Logical data records are passed to the logical reader. After the last record of the buffer received has been processed, the buffer will be made available to the line. If no software EOJ or hardware EOT has been read in the meantime, the task will wait for it. The next received block will be processed like the first one. After EOT on job boundary, the task detaches itself. If EOT is not detected on job boundary the task puts itself in the BSC wait state, waiting for real end of job.

RJE,BSC reader flow is illustrated in Figure 3-63 .



transmitted, the BSC writer task is called as a message task and no logical interface is opened.

If there is list or punch output to be transmitted, the logical interface to the logical writer is opened. Either logical data records from the logical writer or messages from the remote message queue are obtained and are grouped into physical data groups.

After being grouped, the data is written out to the terminal by the BSC I/O Monitor at the time the next I/O operation is performed. At end of job the logical writer is again called to delete the queue set from the queue according to its disposition. If no line turnaround after EOJ is specified, the task obtains the next queue set records from the logical writer, if there are any, and continues transmission until all queue sets have been transmitted.

At the end of a queue set, the writer task indicates that an EOT has to be sent by the line manager after the last buffer has been sent.

When a forms change is required, the RJE,BSC writer task puts itself in the inactive state (IPW$WFO) awaiting remote operator response (SETUP or GO command) after the forms mount message is successfully transmitted to the remote station via a separate message task.

A BSC writer task detaches itself either at the end of a queue set, or when all queue sets of the specified class(es) have been transmitted.

RJE,BSC writer flow is illustrated in Figure 3-64.

Manager requests this by indicating the type of activity in the line control block and linking to the I/O Monitor.

On entrance, the request code being set is analyzed by dividing it into main and subrequest. The request code used to start I/O operations via SVC 0 is divided into the following categories:

- Line preparation and initiation
- Line turnaround from read to write mode and vice versa
- Read sequence
- Abnormal requests
- Write sequence
- Read only requests.
- Retry.

If the analysis results in a valid subrequest, the channel command words used to fulfill the request are created in the related line control block (LCB) and the I/O operation is started via SVC 0.

No WAIT is performed after SVC 0. Control is given back to the Line Manager immediately, which waits for I/O completion or processes asynchronous line activities.

The RJE,BSC I/O Monitor runs under control of the caller's TCB acting as a function module.

RJE,SNA

VSE/POWER 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. VSE/POWER 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 TIK of the VSE/Advanced Functions application task, the VSE/Advanced Functions supervisor handles VTAM page faults as if they were VSE/POWER page faults. In order to minimize the effect of these page faults on non-RJE tasks, VSE/POWER attaches a VSE/Advanced Functions subtask under whose PIB VTAM processing can be executed.

The VSE/POWER 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 manager)
- IPW$$MP (SNA message processor)
- IPW$$LF (SNA logoff processor)
- IPW$$VE (VTAM exit routines)

Figure 3-65 shows the VSE/POWER RJE,SNA interrelationship.

any inbound or outbound process related to a work station and its associated sessions. The SNA manager attaches a VSE/Advanced Functions subtask to the VSE/POWER maintask in which the VSE/POWER 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 VSE/POWER system to VTAM and consists of LOGON, LOSTERM, TPEND and DFASY exits. After the interface to VTAM is opened any logon request to VSE/POWER will be queued by VTAM. After the OPEN ACB request has been completed successfully VSE/POWER issues a SETLOGON START macro to enable VTAM to schedule the VSE/POWER LOGON exit routine.

Logon Processing

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

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

The first LOGON routine (IPW$$LH) in VSE/POWER processes all LOGON requests in the LOGON queue. For each LOGON request the routine performs the following functions:

- Utilizing LOGMODE table entry address and a LOGON work area, it requests the user DATA and BIND parameters from VTAM issuing the INQUIRE SESSPARM macro.
- It performs syntax checking of the REMID and verifies its existence as specified in the PRMT macro.
- It checks the corresponding password if specified.
- It moves the 16 bytes of user information of the data into the session account record without validity checking.
- 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.
- 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

- Ignored parameter (I)
- Enforced parameter (E)
- Mandatory parameter (M)
- Variable parameter (V).

Ignored parameters are neither tested nor modified by VSE/POWER. Enforced parameters are dictated by VSE/POWER on the secondary logical unit. Mandatory parameters are tested, and if incorrect, the logon request is rejected. The variable parameters are copied and affect processing. For a coding example, refer to Appendix A in the VSE/POWER Remote Job Entry User's Guide , which describes LOGON Mode Table and BIND parameter requirements.

Byte	Bits	Content	Description
0	0-7	X'31'	BIND RU code
1	0-3	B'0000'	Format: 0 (M)
	4-7	B'0001'	Bind type non-negotiable. (E)
2		X'03'	FM profile 3. (M)
3		X'03'	TS profile 3. (M)
Primary NAU Protocol			
4	0	B'1'	Multiple RU per chain. (M)
	1	B'0'	Immediate response mode. (M)
	2-3	B'10'	Definite response chains. (M)
	4-5	B'00'	Reserved. (E)
	6	B'0'|B'1'	Compression may not be | may be used. (V)
	7	B'1'	Primary may send EB. (M)
Secondary NAU Protocol			
5	0	B'1'	Multiple RU per chain. (I)
	1	B'0'	Immediate response mode. (M)
	2-3	B'10'	Definite response chains. (M)
	4-5	B'00'	Reserved. (E)
	6	B'0'	Compression may not be used. (E)
	7	B'1'	Secondary may send EB. (M)
Common LU Protocol			
6	0	B'0'	Reserved. (E)
	1	B'1'	FM headers may be used. (M)
	2	B'1'	Brackets are used for RLU and SLU. (M)
	3	B'1'	Bracket termination rule 1. (M)
	4	B'0'|B'1'	Alternate code may not be used | may be used (ASCII). (V)
	5-7	B'000'	Reserved. (E)

Figure 3-66 (Part 1 of 3). BIND Format

Byte	Bits	Content	Description
18	0	B'1'	BS, CR, INP, ENP, LF, HT, VT allowed. (M)
	1	B'1'	SHF allowed. (M)
	2	B'1'	SVF allowed. (M)
	3	B'1'	SVF, SEL allowed. (M)
	4-6	B'000'	Reserved. (E)
	7	B'1'	TRN, IRS allowed. (M)
19		X'00'	Reserved. (E)
20	0	B'0'|B'1'	Document output not allowed | allowed. (V, default = B'1')
	1	B'0'|B'1'	Card format not allowed | allowed. (V, default = B'1')
	2	B'0'	Exchange media not allowed. (E)
	3	B'0'	Disk, data management not allowed. (E)
	4-7	X'0'	Reserved. (E)
SLU Usage Indication (Inbound)			
21	0	B'0'	Interrupt (number of levels). (E)
	1	B'0'	No compaction. (E)
	2	B'0'	No PDIR. (E)
	3-7	B'00000'	Reserved. (E)
22		X'00'	Reserved. (E)
23	0	B'0'	BS, CR, INP, ENP, LF, HT, VT not allowed. (E)
	1	B'0'	SHF not allowed. (E)
	2	B'0'	SVF not allowed. (E)
	3	B'0'	SVF, SEL not allowed. (E)
	4-6	B'000'	Reserved. (E)
	7	B'1'	TRN, IRS allowed. (I, default = B'1')
24		X'00'	Reserved. (E)
25	0	B'0'	Document output not allowed. (E)
	1	B'1'|B'0'	Card fmt allowed | not alld. (V,def.=B'1')
	2	B'0'|B'1'	Exchange media not allowed | allowed (V)
	3	B'0'	Disk, data management not allowed. (E)
	4-7	X'0'	Reserved. (E)
26		X'00'	Reserved. (E)

Figure 3-66 (Part 3 of 3). BIND Format

Byte	Bit	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'	Ignored
	1-3	MEDIA	B'000'	CONSOLE
			B'001'	EXCHANGE MEDIA (only inbound)
			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		(See Note)	DS selection
	3	DST	B'0'	Basic exchange not supported
			B'1'	Basic exchange supported (inbound only)
	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 must be <= 128
6-n				DSNAME which is defined by architecture in bytes 6-n is not supported by VSE/POWER.

Figure 3-67. FMH1 Format

Note: 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)

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. VSE/POWER only supports FMH2 outbound, but not inbound.

The format of the FMH2 is shown in Figure 3-69.

Byte	Bit	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 in the form MM/DD/YY.
12-19		TIME	HH.MM.SS	Time of queue set creation in the form HH.MM.SS.
20-27		FORMS	C'ccccbbbb'	Forms name in the form C'ccccbbbb' Default is all blanks. The forms can be provided by the * $$ LST FNO= parameter, by the LFCB macro, or by the SEGMENT macro, JECL= operand, where a * $$ LST FNO= is specified.
28-35		FCB	C'cccccccc'	FCB name (1-8 characters)in the form C'cccccccc' left justified. Default is all blanks. The FCB can be provided by the * $$ LST FCB= parameter, by the LFCB macro, or by the SEGMENT macro, JECL= operand, where a * $$ LST FCB= is specified.
36-43		TRAIN	C'bbbbbbbb'	Not supported

Figure 3-69 (Part 1 of 2). FMH2 Format

Byte	Bit	Name	Content	Description
0	0-7	length	(See Note)	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	(See Note)	Compaction table characters

Figure 3-70. FMH3 Format

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

length = 4 + 256 - (m x m) for m < 16
length = 4 + 16 for m = 16

where m is the number of master 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 RU 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 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.

- 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.

- 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 (implicitly or explicitly) is already in use. If in use it rejects the inbound flow.

- 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 VSE/POWER through the LOSTERM exit. If the remote operator issues a conditional LOGOFF VTAM notifies VSE/POWER 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 VSE/POWER where it is handled as a conditional LOGOFF request for all sessions of a given work station.

The work station may log off 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.

VSE/POWER 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 boundaries, that is, 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.

Routine	Called/ Attached by	Returns to	Function or Notes
IPW$$IB Inbound Processor	IPW$$SN	IPW$$NU	Issues RECEIVE Specific request to VTAM to receive data and then de-blocks the data for spooling by IPW$$LR. Processes remote operator commands: • START • STOP • FLUSH • RESTART • SETUP • GO • SIGNOFF and transfers all other commands to IPW$$CM for processing. Posts the outbound processor command following GO, or RESTART when intervention is required. Posts the SNA manager and detaches.
IPW$$LF Logoff Processor	IPW$$SN	IPW$$NU	Logs off a logical unit using the VTAM macros SESSIONC and CLSDST. Sends message "1V12I" to the remote terminal and then sends the central operator the message "1V11I". Posts the SNA manager and detaches.
IPW$$LH Logon Processor 1	IPW$$SN		Establishes 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 Posts the SNA manager and detaches.

Figure 3-71 (Part 1 of 4). Description of RJE,SNA Routines

Routine	Called/ Attached by	Returns to	Function or Notes
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$$SN. Attaches a VSE subtask. Issues VTAM "RECEIVE ANY" macro. Prints central operator message "1V04I" and waits on ECB posting.
IPW$$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$$SN ECB and VSE/POWER master ECB, and waits on posting by IPW$$SN. At termination time, the VTAM macro SETLOGON QUIESCE is called to halt further LOGON requests.
IPW$$SN Segment MAIN	Posted by: • VTAM due to: RECEIVE ANY input via VTAM exit • VSE/POWER routines: IPW$$AQ IPW$$CM IPW$$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 VSE/POWER 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) • Logon/logoff tasks Then a loop back is made to wait on further posting.

Figure 3-71 (Part 3 of 4). Description of RJE,SNA Routines

RJE,SNA Control Blocks/ Work Areas	When Created:	Freed:	Created by:	Stored at:	Purpose/General Contents
CAT (Control Address Table)	At VSE/POWER Initialization	At VSE/POWER Termination	IPW$$I1	Real storage area	Pointers to modules and major control blocks.
SNCB (SNA Control Block)	At VSE/POWER Initialization	At VSE/POWER Termination	IPW$$I7	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.
LRCB (Logon Request Control Block)	At first LU LOGON	When (last) LOGON processed (IPW$$LH)	IPW$$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 Initialized by IPW$$LN	At work station logoff time (IPW$$LF)	IPW$$LH	GETVIS area	Contains information required for LU session, e.g. variable BIND parameter information
Logon LUCB	At LOGON of first LU of work station	At RJE,SNA Termination (IPW$$SN)	IPW$$VE-Segment LOGON	GETVIS area	Contains information required for LU session. Used as work area during logon processing.
SUCB (SNA Unit Control Block)	Space reserved at LOGON of first LU of work station	At LOGOFF of last LU of work station (IPW$$LF)	IPW$$LH	GETVIS area	Contains information pertaining to the work station of two types: a) General information, for example: REMID, SESSLIM. b) Device information for LSTn, PUN, RDR and Console, for example: class. Created by copying the LOGON SUCB onto the reserved SUCB area. One is created for each work station logged on.
Logon SUCB	At logon of first LU of work station	At RJE,SNA Termination IPW$$SN	IPW$$VE-Segment	GETVIS area	Used as a work area during LOGON processing.
WACB	• WACB for inbound interruption	• At LOGOFF of last session of work station IPW$$LF	IPW$$LH	GETVIS area	Contains VTAM RPLs, RU buffers and some BIND information. One exists for each SUCB; another exists for each LUCB logged on for inbound; and another exists for each LUCB logged on during outbound data or message handling.
	• WACB for LU inbound data	• At LOGOFF of LU IPW$$LF	IPW$$LH	GETVIS area	
	• WACB for LU outbound data	• At LSTn/ PUN task termination (MP,OB)	IPW$$OB IPW$$MP	GETVIS area	
LOGON WACB	At LOGON of first LU of work station	At RJE,SNA Termination (SN)	IPW$$VE-Segment	GETVIS area	Used as work area for LOGON processing
RMCB (Remote Control Block)	VSE/POWER Initialization	VSE/POWER Termination	IPW$$I7	GETVIS area	Contains information from the PRMT macro.

Figure 3-72. Description of RJE,SNA Control Blocks and Work Areas

RJE,SNA – Logon

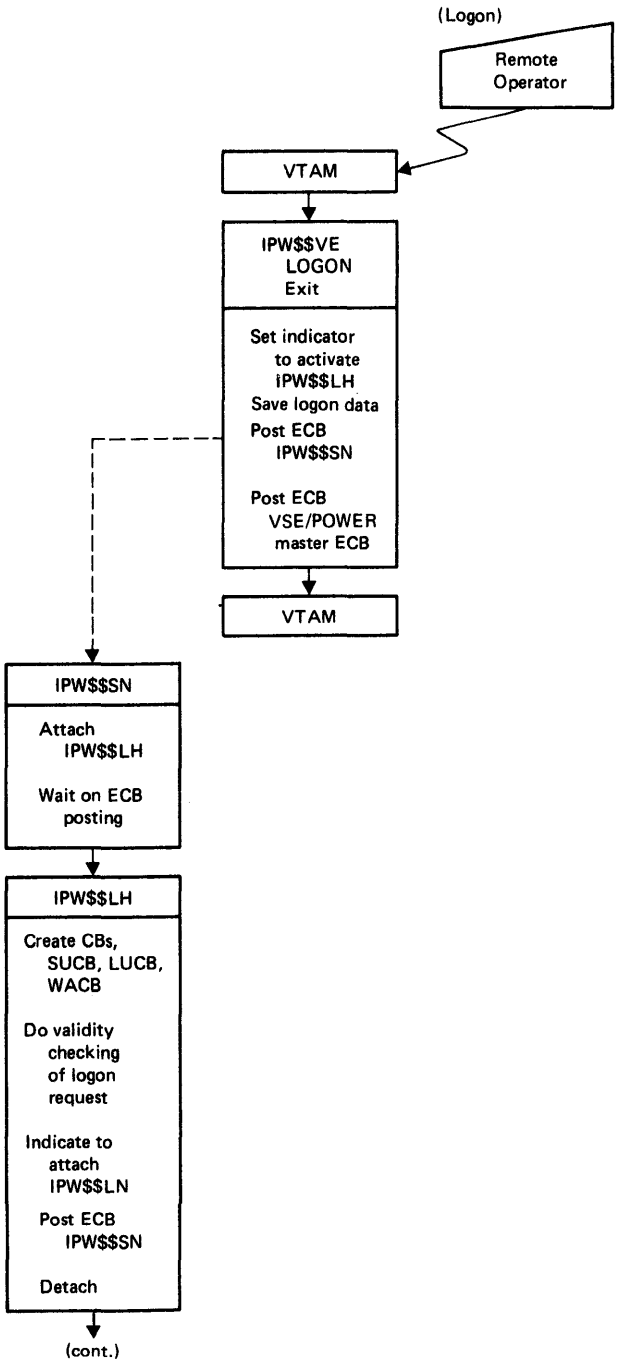


Figure 3-73 (Part 2 of 9). RJE,SNA Execution Flow

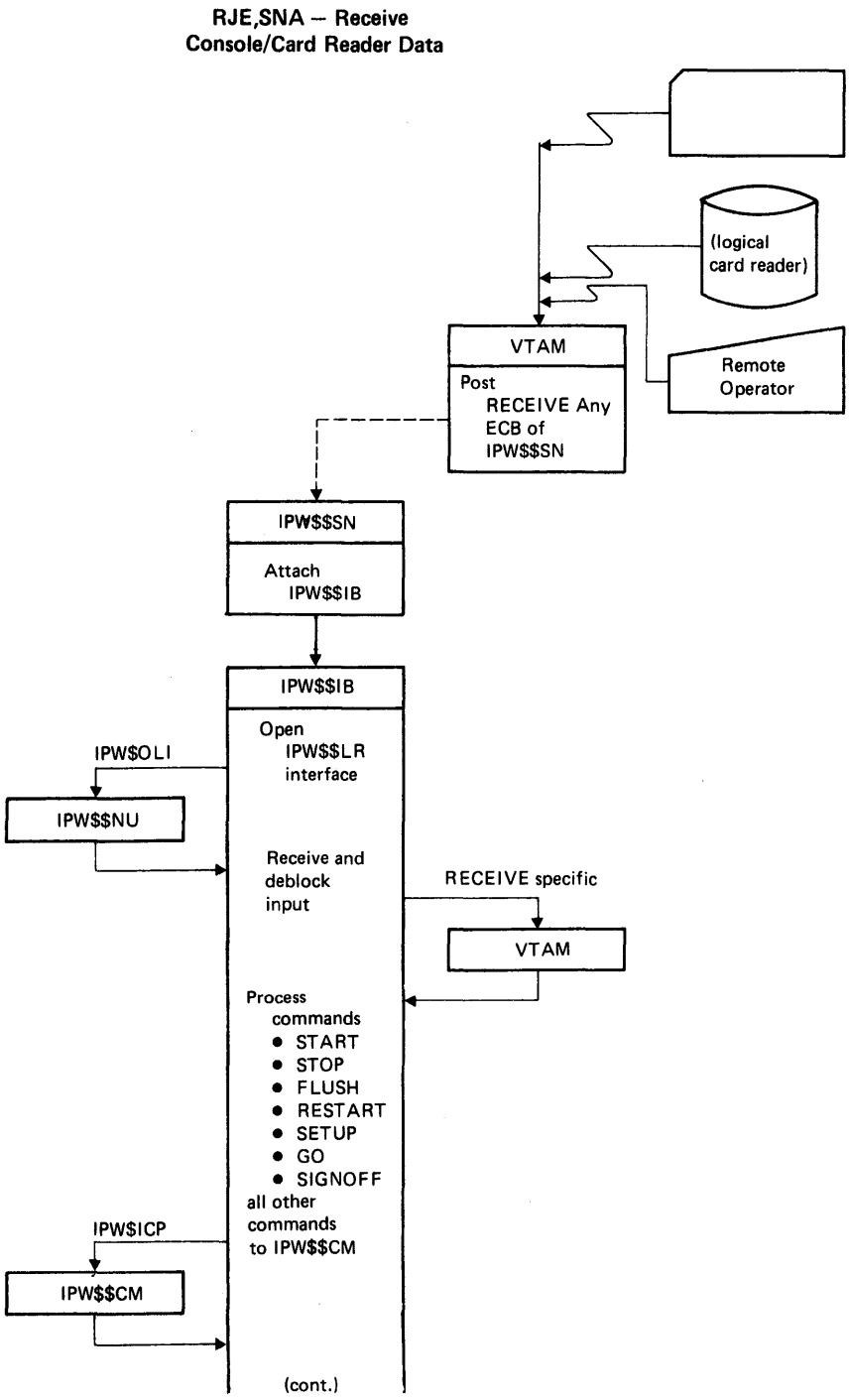


Figure 3-73 (Part 4 of 9). RJE,SNA Execution Flow

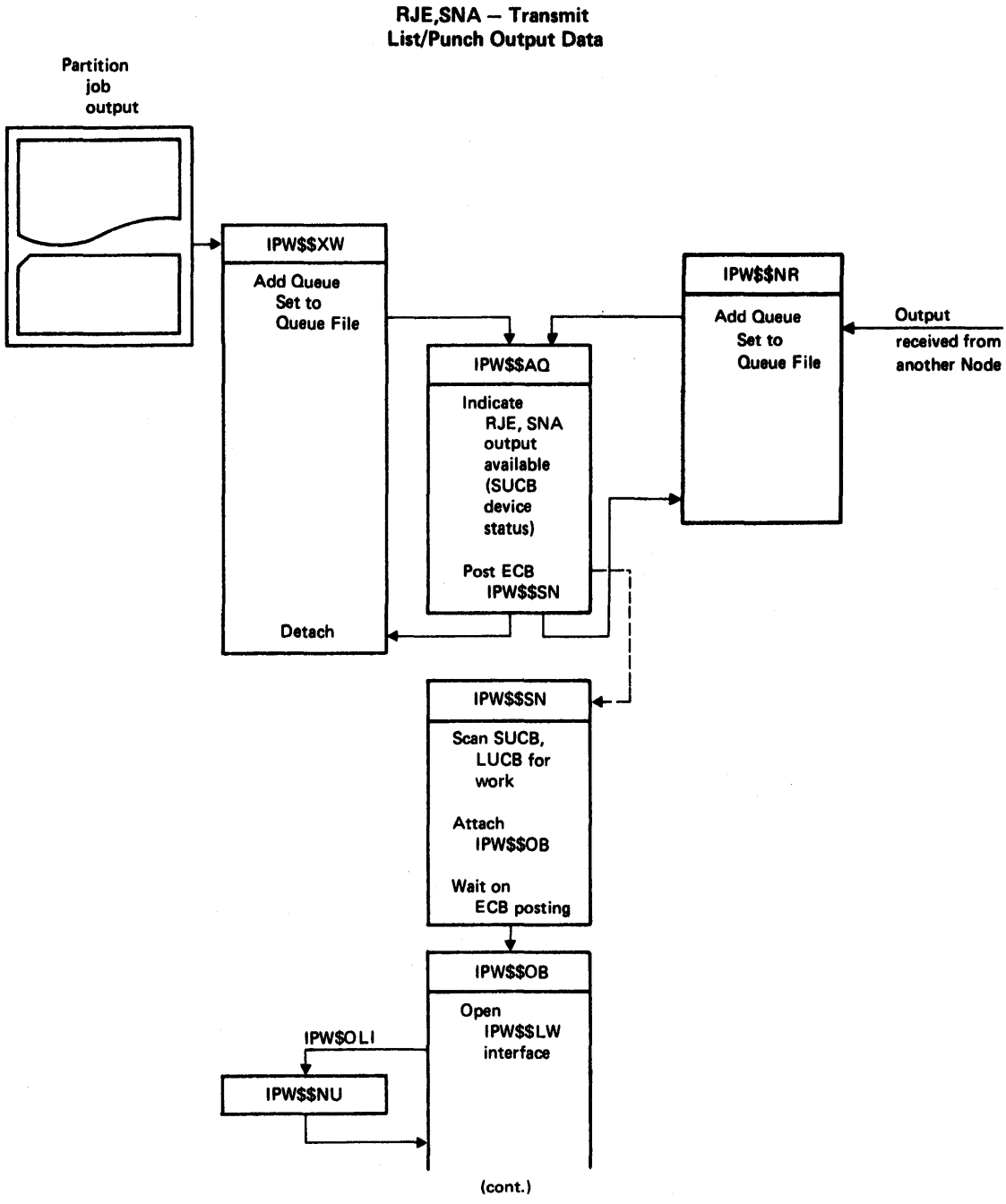


Figure 3-73 (Part 6 of 9). RJE,SNA Execution Flow

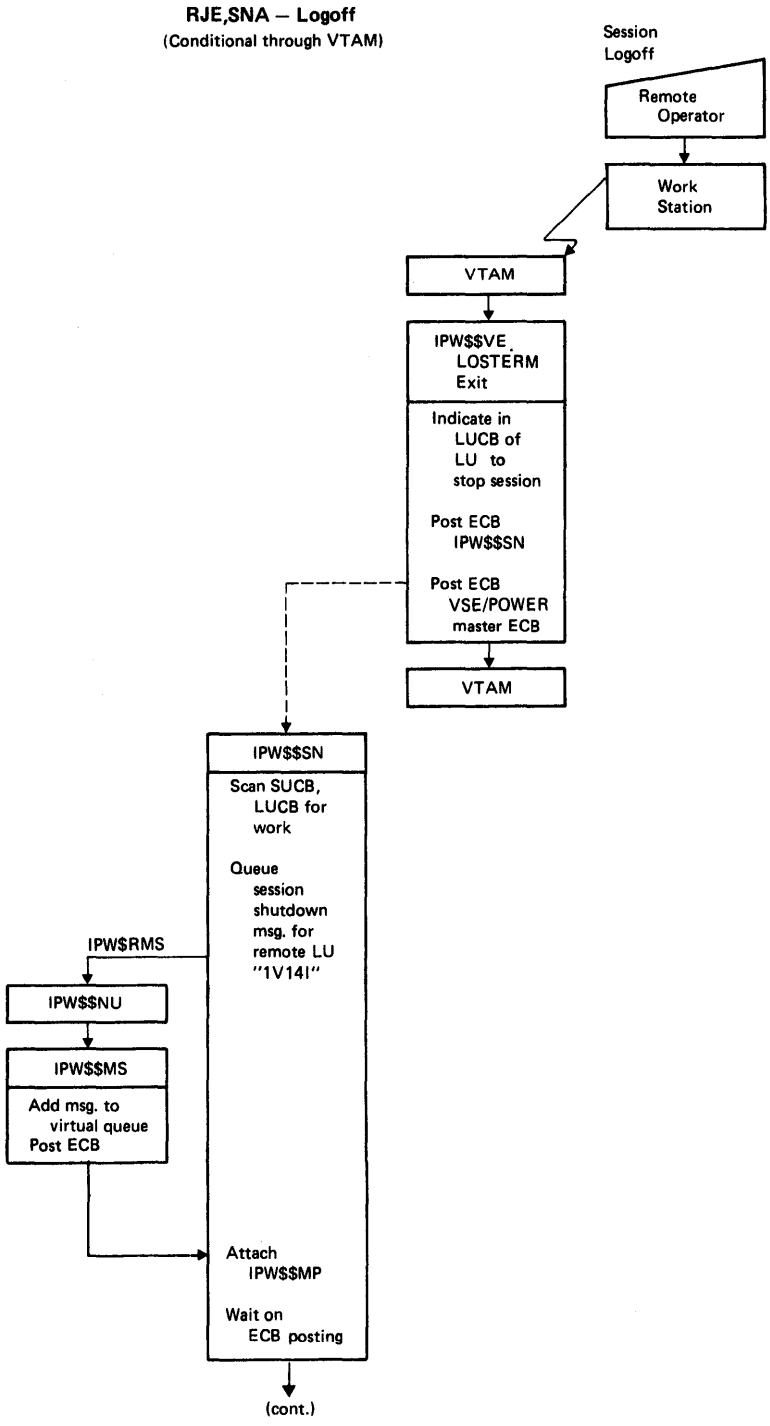


Figure 3-73 (Part 8 of 9). RJE,SNA Execution Flow

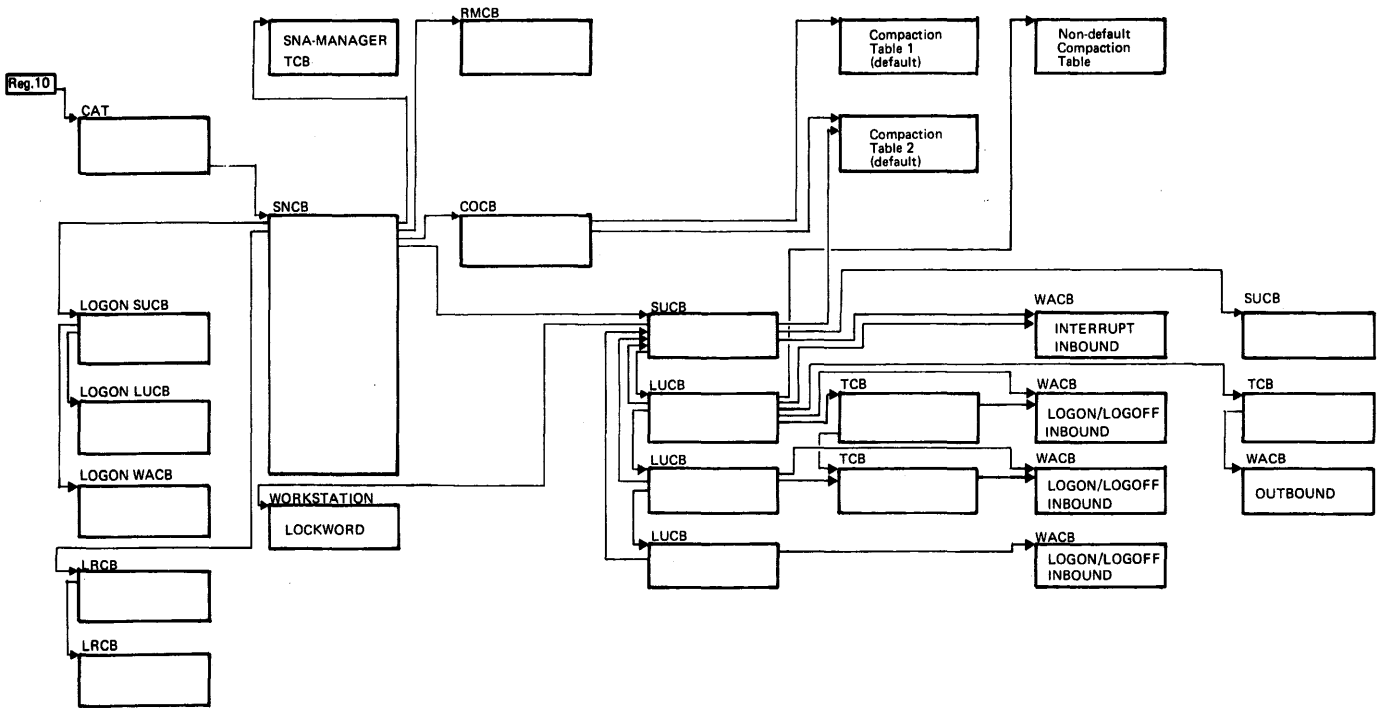


Figure 3-74. RJE,SNA Control Block and Work Area Chaining

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 Task" on page 3-65.

RJE,BSC and PNET,BSC Channel End Appendage

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

The appendage routine gets control from the VSE/Advanced Functions I/O interrupt handler whenever an interrupt is received from a BSC line. It then performs the following functions:

- If the line is a RJE line:

  1. It queues the LCB to an LCB chain that will be processed by the line manager.

  2. It posts the line manager ECB, the VSE/POWER master ECB, and sets the VSE/POWER partition dispatchable.

- If the line is a PNET line:

  1. It queues the input/output buffer to the buffer queue anchored to the TCB of the PNET driver.

  2. It posts the PNET driver ECB, the VSE/POWER master ECB, and sets the VSE/POWER partition dispatchable.

Control is then returned to the next sequential instruction in the VSE/Advanced Functions supervisor.

Hot Reader Appendage

The supervisor passes control to this appendage whenever an unsolicited 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 VSE/POWER partition is set dispatchable. In all other cases, no action is taken.

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
Unsolicited device end	Hot Reader	Set RDR task dispatchable, activate partition.	TCB
BSC channel end	Channel END	Set RJE,BSC line manager (LM)/ PNET driver (LD) dispatchable, activate VSE/POWER partition	TCB,LCB NCB
SVC 0 intercepted	SVC 0	Set XR/XW task dispatchable, activate partition.	TCB,PDB
SVC 90/SVC 91 interrupt	SVC 90/SVC 91	Set XR task dispatchable, activate partition.	TCB,PDB
Expiry of interval timer	Interval Timer	Set VSE/POWER partition dispatchable	CAT

Figure 3-75. Appendage Summary

- Another system could be saving the account file, which must also complete before control can be released

- Another processor that has locked the queue file may have a job that is both in a higher priority partition than VSE/POWER and in a loop; thus it cannot issue an UNLOCK

If one of these cases occurs and the time interval T4 expires, a warning message will be sent to the operator and VSE/POWER will set up time interval T4 again.

The timer task performs its time slicing and LOCK/UNLOCK within a VSE subtask. This ensures that the VSE/POWER main task and its internal subtasks do not enter wait state.

If PNET is used in conjunction with shared spooling, then an additional table, called node attached table (NAT), is initialized in the DMB. This table is used to communicate between sharing systems those adjacent nodes that are currently attached for networking. Whenever a node is signed on an entry is made in the NAT. When the node is terminated then the entry is removed.

Whenever a job or output is destined for a node which is not directly attached to this sharing system, the add-to-queue function checks the NAT for the required node name. If an entry is found then the job or output available bit is set to inform any other sharing systems that something has been placed in the transmission queue.

The NAT is part of the master record and thus is written to disk and read from disk by the timer task (IPW$$TI) during every time slice.

There is also another table in the DMB, called the 'Shared Remote-Id Table (SRT), which is used in a similar way to the node attached table. Every time output becomes available for a remote work station which is not attached to the system producing the output, a bit is set on for the corresponding remote-id in the SRT. This informs the other sharing systems that there is output in the queue for them. This table is also controlled by the timer task (IPW$$TI).

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 also among these names, 5, "Data Areas" of this manual, the charts in VSE/POWER Program Logic Manual Part 2, and VSE/POWER Program Logic Manual Part 3.

The names listed in the column "Module/Microfiche Name" are the names of the respective microfiche cards.

- 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.
- If the name is a phase, locate the appropriate chart in VSE/POWER Program Logic Manual Part 2, or VSE/POWER Program Logic Manual Part 3,
- If the name is a linkage macro, determine the invoked phase and its chart in VSE/POWER Program Logic Manual Part 2, or VSE/POWER Program Logic Manual Part 3.
- If the name is a definition macro (control block, or data block), locate the matching data area by using Figure 3-5 as a reference.

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

INCS	CSECT	IPW$$IN	IPW$$IN	IN
IPCS	CSECT	IPW$$IP	IPW$$IP	IP
IPW$xxx	see macro list below			
IPW$$AM	MODULE			
IPW$$AQ	PHASE	IPW$$AQ	IPW$$QM	AQ
IPW$$AS	PHASE	IPW$$AS	IPW$$DM	AS
IPW$$AT	PHASE	IPW$$AT	IPW$$AT	AT
IPW$$BM	PHASE	IPW$$BM	IPW$$BM	BM
IPW$$BR	PHASE	IPW$$BR	IPW$$BR	BR
IPW$$BS	PHASE	IPW$$BS	IPW$$BS	BS
IPW$$BW	PHASE	IPW$$BW	IPW$$BW	BW
IPW$$CA	PHASE	IPW$$CA	IPW$$CA	CA
IPW$$CAC	PHASE	IPW$$CAC	IPW$$CAC	CAC
IPW$$CB	PHASE	IPW$$CB	IPW$$CB	CB
IPW$$CC	PHASE	IPW$$CC	IPW$$CC	CC
IPW$$CD	PHASE	IPW$$CD	IPW$$CD	CD
IPW$$CE	PHASE	IPW$$CE	IPW$$CE	CE
IPW$$CF	PHASE	IPW$$CF	IPW$$CF	CF
IPW$$CG	PHASE	IPW$$CG	IPW$$CG	CG
IPW$$CH	PHASE	IPW$$CH	IPW$$CH	CH
IPW$$CI	PHASE	IPW$$CI	IPW$$CI	CI
IPW$$CJ	PHASE	IPW$$CJ	IPW$$CJ	CJ
IPW$$CL	PHASE	IPW$$CL	IPW$$CL	CL
IPW$$CLD	PHASE	IPW$$CLD	IPW$$CLD	CLD
IPW$$CM	PHASE	IPW$$CM	IPW$$CM	CM
IPW$$CN	PHASE	IPW$$CN	IPW$$CN	CN
IPW$$CO	PHASE	IPW$$CO	IPW$$CO	CO
IPW$$CP	PHASE	IPW$$CP	IPW$$CP	CP
IPW$$CPF	PHASE	IPW$$CPF	IPW$$CPF	CPF
IPW$$CPS	PHASE	IPW$$CPS	IPW$$CPS	CPS
IPW$$CR	PHASE	IPW$$CR	IPW$$CR	CR
IPW$$CRE	PHASE	IPW$$CRE	IPW$$CRE	CRE
IPW$$CS	PHASE	IPW$$CS	IPW$$CS	CS
IPW$$CT	PHASE	IPW$$CT	IPW$$CT	CT
IPW$$CU	PHASE	IPW$$CU	IPW$$CU	CU
IPW$$CX	PHASE	IPW$$CX	IPW$$CX	CX
IPW$$DD	PHASE	IPW$$DD	IPW$$DD	DD
IPW$$DM	MODULE			-
IPW$$DQ	PHASE	IPW$$DQ	IPW$$QM	DQ
IPW$$ER	PHASE	IPW$$ER	IPW$$ER	ER
IPW$$FQ	PHASE	IPW$$FQ	IPW$$QM	FQ
IPW$$GA	PHASE	IPW$$GA	IPW$$AM	GA
IPW$$GD	PHASE	IPW$$GD	IPW$$DM	GD
IPW$$GF	PHASE	IPW$$GF	IPW$$GF	GF
IPW$$IB	PHASE	IPW$$IB	IPW$$IB	IB
IPW$$IC	PHASE	IPW$$IC	IPW$$DM	IC
IPW$$IP	PHASE	IPW$$IP	IPW$$IP	IP
IPW$$I1	PHASE	IPW$$I1	IPW$$I1	I1
IPW$$I2	PHASE	IPW$$I2	IPW$$I2	I2
IPW$$I3	PHASE	IPW$$I3	IPW$$I3	I3
IPW$$I4	PHASE	IPW$$I4	IPW$$I4	I4
IPW$$I5	PHASE	IPW$$I5	IPW$$I5	I5
IPW$$I7	PHASE	IPW$$I7	IPW$$I7	I7
IPW$$IN	PHASE	IPW$$IN	IPW$$IN	IN

IPW$$S3	PHASE	IPW$$S3	IPW$$S3	S3
IPW$$TI	PHASE	IPW$$TI	IPW$$TI	TI
IPW$$TR	PHASE	IPW$$TR	IPW$$TR	TR
IPW$$T1	PHASE	IPW$$T1	IPW$$T1	T1
IPW$$VE	PHASE	IPW$$VE	IPW$$VE	VE
IPW$$XJ	PHASE	IPW$$XJ	IPW$$XJ	XJ
IPW$$XR	PHASE	IPW$$XR	IPW$$XR	XR
IPW$$XW	PHASE	IPW$$XW	IPW$$XW	XW
I1CS	CSECT	IPW$$I1	IPW$$I1	I1
I2CS	CSECT	IPW$$I2	IPW$$I2	I2
I3CS	CSECT	IPW$$I3	IPW$$I3	I3
I4CS	CSECT	IPW$$I4	IPW$$I4	I4
I5CS	CSECT	IPW$$I5	IPW$$I5	I5
I7CS	CSECT	IPW$$I7	IPW$$I7	I7
LDCS	CSECT	IPW$$LD	IPW$$LD	LD
LDCS1	CSECT	IPW$$LD1	IPW$$LD1	LD1
LDCS2	CSECT	IPW$$LD2	IPW$$LD2	LD2
LDCS3	CSECT	IPW$$LD3	IPW$$LD3	LD3
LDCS4	CSECT	IPW$$LD4	IPW$$LD4	LD4
LDCS5	CSECT	IPW$$LD5	IPW$$LD5	LD5
LFCS	CSECT	IPW$$LF	IPW$$LF	LF
LHCS	CSECT	IPW$$LH	IPW$$LH	LH
LMCS	CSECT	IPW$$LM	IPW$$LM	LM
LNCS	CSECT	IPW$$SN	IPW$$LN	LN
LRCB	Storage descriptor of control block.			
LRCS	CSECT	IPW$$LR	IPW$$LR	LR
LUCB	Storage descriptor of control block.			
LUCS	CSECT	IPW$$LU	IPW$$LU	LU
LWCS	CSECT	IPW$$LW	IPW$$LW	LW
MCB	Storage descriptor of control block.			
MDCS	CSECT	IPW$$MD	IPW$$MD	-
MMCS	CSECT	IPW$$MM	IPW$$MM	-
MMB	Storage descriptor of control block.			
MPCS	CSECT	IPW$$MP	IPW$$MP	MP
MSCB	Storage descriptor of control block.			
MSCS	CSECT	IPW$$MS	IPW$$MS	MS
MXCS	CSECT	IPW$$MX	IPW$$MX	MX
NCB	Storage descriptor of control block.			
NCCS	CSECT	IPW$$NC	IPW$$NC	NC
NDT	Storage descriptor of control block.			
NKCS	CSECT	IPW$$NK	IPW$$NK	NK
NMCS	CSECT	IPW$$NM	IPW$$NM	NM
NPCS	CSECT	IPW$$NP	IPW$$NP	NP
NQCS	CSECT	IPW$$NQ	IPW$$QM	NQ
NRCS	CSECT	IPW$$NR	IPW$$NR	NR
NSCS	CSECT	IPW$$NS	IPW$$NS	NS
NTCS	CSECT	IPW$$NT	IPW$$NT	NT
OBCS	CSECT	IPW$$OB	IPW$$OB	OB
OCCS	CSECT	IPW$$OC	IPW$$OC	OC
OECS	CSECT	IPW$$OE	IPW$$OE	OE
OFCS	CSECT	IPW$$OF	IPW$$OF	OF
OTCS	CSECT	IPW$$OT	IPW$$OT	OT
PACS	CSECT	IPW$$PA	IPW$$AM	PA
PDB	Storage descriptor of control block.			

MACRO LIST

Macro	Type	Phase	Chart
IPW$AQS	LINKAGE	IPW$$AQ	AQ
IPW$ATT	LINKAGE	IPW$$NU	NU
IPW$BUF	LINKAGE	IPW$$BS	BS
IPW$CAF	LINKAGE	IPW$$GA	GA/GF
IPW$CLI	LINKAGE	Note 1	
IPW$CNC	LINKAGE		
IPW$CTT	LINKAGE	IPW$$NU	NU
IPW$DAB	DEFINITION		
IPW$DAC	DEFINITION		
IPW$DBA	DEFINITION		
IPW$DBC	DEFINITION		
IPW$DCB	DEFINITION		
IPW$DCI	DEFINITION		
IPW$DCM	DEFINITION		
IPW$DCO	DEFINITION		
IPW$DCP	DEFINITION		
IPW$DCW	DEFINITION		
IPW$DCT	DEFINITION		
IPW$DDE	DEFINITION		
IPW$DDR	DEFINITION		
IPW$DEF	DEFINITION		
IPW$DET	LINKAGE	IPW$$NU	NU
IPW$DFC	DEFINITION		
IPW$DGN	DEFINITION		
IPW$DJK	DEFINITION		
IPW$DKA	DEFINITION		
IPW$DLC	DEFINITION		
IPW$DLR	DEFINITION		
IPW$DLU	DEFINITION		
IPW$DMC	DEFINITION		
IPW$DMD	DEFINITION		
IPW$DMM	DEFINITION		
IPW$DMS	DEFINITION		
IPW$DNC	DEFINITION		
IPW$DNR	DEFINITION		
IPW$DPA	DEFINITION		
IPW$DPC	DEFINITION		
IPW$DPD	DEFINITION		
IPW$DPN	DEFINITION		
IPW$DPW	DEFINITION		
IPW$DQC	DEFINITION		
IPW$DQR	DEFINITION		
IPW$DQS	LINKAGE	IPW$$DQ	DQ
IPW$DRM	DEFINITION		
IPW$DRQ	DEFINITION		
IPW$DSA	DEFINITION		
IPW$DSC	DEFINITION		
IPW$DSD	DEFINITION		
IPW$DSL	DEFINITION		
IPW$DSN	DEFINITION		

IPW$RSW	LINKAGE	IPW$$NU	NU
IPW$SAV	LINKAGE	Note 2	
IPW$SRJ	LINKAGE	IPW$$SC	SC
IPW$STM	LINKAGE	IPW$$NU	NU
IPW$SXJ	LINKAGE	IPW$$XJ	XJ
IPW$ULP	LINKAGE	IPW$$LU	LU
IPW$UNV	LINKAGE	IPW$$NU	NU
IPW$VCA	LINKAGE	IPW$$CM	CM
IPW$VDA	LINKAGE	IPW$$NU	NU
IPW$WFB	LINKAGE		
IPW$WFC	LINKAGE	IPW$$NU	NU
IPW$WFD	LINKAGE	IPW$$NU	NU
IPW$WFI	LINKAGE	IPW$$NU	NU
IPW$WFL	LINKAGE	IPW$$NU	NU
IPW$WFM	LINKAGE	IPW$$NU	NU
IPW$WFO	LINKAGE	IPW$$NU	NU
IPW$WFQ	LINKAGE	IPW$$NU	NU
IPW$WFS	LINKAGE	IPW$$NU	NU
IPW$WTD	LINKAGE	IPW$$NU	NU
IPW$WTO	LINKAGE	IPW$$NU	NU
IPW$WTQ	LINKAGE	IPW$$NU	NU
IPW$WTR	LINKAGE	IPW$$NU	NU
IPW$WTT	LINKAGE	IPW$$NU	NU

Notes:

1. Refer to 'Interface Linkage' under 'Linkage Conventions' in VSE/POWER Program Logic Manual Part 2.

2. Refer to 'Function Linkage' under 'Linkage Conventions' in VSE/POWER Program Logic Manual Part 2. For linkage conventions and register saving conventions, refer to the appropriate sections of the TCB, which is described in 5, "Data Areas" on page 5-1 of this book.

		IPW$$I4
		IPW$$I5
1Q24I	QUEUE FILE CHAIN ERROR	IPW$$RY
1Q25A	partition-id IN STOP STATE	IPW$$T1
1Q26I	GETVIS AREA TOO SMALL	IPW$$I1
		IPW$$I5
1Q27I	UNABLE TO INITIALIZE SPOOL MANAGEMENT	IPW$$I7
1Q28I	E O V ON cuu	IPW$$OT
1Q29I	END OF INPUT ON cuu	IPW$$SY
1Q2AI	POFFLOAD SUCCESSFULLY COMPLETED, cuu	IPW$$OF
1Q2BI	NOTHING TO SAVE ON cuu	IPW$$OF
1Q2CI	PSW=XXXXXXXXXXXXXXXX, CC=yy -progr. check desc.	IPW$$AT
1Q2DI	VSE/POWER CANCELED DUE TO PROGRAM REQUEST	IPW$$AT
1Q30D	ABNORMAL VSE/POWER TERMINATION, PRINTER=	IPW$$AT
1Q30D	INVALID PRINTER TYPE, RE-ENTER=	IPW$$AT
1Q31I	MORE THAN 80% FULL ACCOUNT FILE (IJAFILE)	IPW$$AM
		IPW$$PF
1Q32I	NO MORE ACCOUNT FILE (IJAFILE) SPACE FOR task,cuu	IPW$$PF
		IPW$$AM
1Q33I	STOPPED task,cuu	IPW$$XR
	partition-id	IPW$$OF
		IPW$$TR
1Q34I	partition-id WAITING FOR WORK	IPW$$XR
1Q34I	task WAITING FOR WORK ON cuu	IPW$$LW
		IPW$$PR
1Q35A	EOV ON TAPE, MOUNT NEW TAPE FOR task,cuu	IPW$$XW
1Q35A	EOF ON cuu	IPW$$PR
1Q36I	task CANCELED DUE TO I/O ERROR ON cuu	
1Q37I	JECL STATEMENT INCORRECT NEAR COLUMN n	IPW$$LR
1Q38I	NO DASD SPACE AVAILABLE FOR task,cuu	IPW$$DM
		IPW$$QM
1Q39I	JOB jobname FLUSHED BY THE OPERATOR OR BY VSE/POWER	IPW$$LW
1Q3AI	ERROR WHILE PROCESSING ACCOUNT RECORD, RC=xxx	IPW$$PF
1Q3CI	INVALID BLOCKSIZE FOR filename	IPW$$SF
1Q3DI	INVALID CI-SIZE FOR filename	IPW$$SF
1Q40A	ON cuu taskid FORMS ffff NEEDED FOR jobname jobnumber	IPW$$LW
		IPW$$OB
1Q41I	WRONG PRINTER/PUNCH FOR jobname jobnumber, cuu	IPW$$LW
1Q42I	PAGE COUNT EXCEEDS END OF QUEUE ENTRY FOR cuu.	IPW$$LW
1Q43I	END-OF-FILE ON TAPE FOR task,cuu	IPW$$QM
1Q44I	BOOK lib.book NOT FOUND	IPW$$SL
1Q45I	SLI STATEMENT NOT SUPPORTED partition-id	IPW$$XJ
1Q46I	DISP FORCED TO D FOR jobname jobnumber	IPW$$XJ
1Q47I	partition-id jobname jobnumber FROM remid user TIME=	IPW$$XJ
		IPW$$XR
1Q48D	NO MATCHING SPOOL DEVICE partition-id	IPW$$XJ
1Q49I	INVALID DELIMITER partition-id	IPW$$XJ
1Q4AI	MESSAGE DISCARDED, RC=nnnn	IPW$$NS
1Q4BI	NOTIFY SUPPORT CANCELED, RC=nnnn	IPW$$NS
1Q50I	UNKNOWN KEYWORD partition-id	IPW$$XJ
1Q51I	INVALID paraname PARAMETER partition-id	IPW$$XJ
1Q52I	OUTPUT LIMIT EXCEEDED FOR jobname jobnumber part-id,cuu	IPW$$XW
1Q53I	OUTPUT SEGMENTED FOR jobname jobnumber part-id,cuu	IPW$$XW
1Q54I	Buffer ERROR FOR jobname jobnumber task,cuu	IPW$$AS

```
                                                                  IPW$$SF
1Q84I  ACCOUNT SUPPORT CANCELED                                   IPW$$AM
                                                                  IPW$$PF
1Q85I  task,cuu WAITING FOR VIRTUAL STORAGE                       IPW$$NU
1Q86A  DISKETTE REQUIRED ON cuu, FOR jobname jobnumber, HDR1..    IPW$$ER
1Q87I  cuu EOJ ADDED jobname, jobnumber                           IPW$$SY
                                                                  IPW$$ER
1Q88I  INVALID 3540 UNIT FOR partition-id                         IPW$$XR
1Q89I  PROGRAM OUT OF SEQUENCE IN partition-id                    IPW$$XR
1Q90I  * $$ RDR STATEMENT NOT ALLOWED, JOB FLUSHED                IPW$$LR
       * $$ RDR STATEMENT NOT PROCESSED, JOB FLUSHED              IPW$$ER
1Q91D  cuu NON-COMPATIBLE DISKETTE FOR RDR,cuu                    IPW$$OE
1Q92D  cuu NO HDR1 FOR fileid, RDR,cuu R=                         IPW$$OE
1Q93D  cuu SECURED VOLUME/FILE FOR RDR,cuu R=                     IPW$$OE
1Q94D  cuu EXPECT VOL nn, NOT mm, RDR,cuu R=                      IPW$$OE
1Q95D  cuu NON-VERIFIED fileid, RDR,cuu R=                        IPW$$OE
1Q96I  cuu fileid IS EMPTY FILE FOR RDR,cuu                       IPW$$OE
1Q97D  cuu PREMATURE LAST VOL FOR RDR,cuu R=                      IPW$$OE
1Q98D  cuu fileid TOO MANY VOLS RDR,cuu R=                        IPW$$OE
1Q9nD  INVALID RESPONSE R=                                        IPW$$OE
1Q9nD  NO PRECEEDING VOL, INCONSIST RESP R=                       IPW$$OE
1QA0I  NO SUBTASK AVAILABLE FOR task,cuu                          IPW$$AS
                                                                  IPW$$LD4
                                                                  IPW$$TI
1QA1I  SETPRT ROUTINE NOT FOUND IN SVA  task,cuu                  IPW$$AS
1QA3I  SETPRT ERROR FOR jobname jobnumber task,cuu                IPW$$AS
1QA4I  OUTPUT PROCESSING STOPPED jobname jobnumber task,cuu       IPW$$AS
1QA5A  cuu SETUP REQUIRED jobname FORM=ffff FLASH=hhhh THREAD..   IPW$$LW
1QA6I  NO STORAGE AVAILABLE FOR TTTTT, cuu                        IPW$$AS
                                                                  IPW$$I7
1QA7A  MOUNT TRAIN FOR UCS=uuuuuuuu jobname jobnumber task,cuu    IPW$$PL
1QA9A  task,cuu WAITING FOR OPERATOR REACTIVATION                 IPW$$LW
1QB0I  SUPERVISOR WITHOUT DASD SHARING FEATURE                    IPW$$I2
1QB1I  filename NOT ON SHARED DEVICE                              IPW$$I2
                                                                  IPW$$I4
1QB2D  IS ANY OTHER VSE/POWER SYSTEM ALREADY INITIALIZED ?        IPW$$I2
1QB4I  LOCK TABLE SPACE EXHAUSTED                                 IPW$$TI
1QB5I  INTERAL MACRO CALL FAILED, RC=rrmm                         IPW$$I2
                                                                  IPW$$I4
                                                                  IPW$$I5
                                                                  IPW$$I7
                                                                  IPW$$SM
                                                                  IPW$$TI
                                                                  IPW$$T1
                                                                  IPW$$XW
1QB6I  QUEUE FILE LOCKED BY ANOTHER SYSTEM                        IPW$$TI
1QB7I  QUEUE FILE RECOVERY IN PROGRESS [FOR SYSID n1,n2...]       IPW$$RY
1QB8I  QUEUE FILE RECOVERY COMPLETED                              IPW$$RY
1R02I  LINE cuu STOPPED, TIME=hh:mm:ss                            IPW$$LM
1R03I  TRANSM xxxxx, TIMEOUTS xxxxx, ERRORS xxxxx                 IPW$$LD3
                                                                  IPW$$LM
1R04I  LINE cuu FORCED TO STOP BY PSTOP KILL COMMAND, TIME=....   IPW$$LM
1R05I  SENT xxxxx, RECEIVED xxxxx                                 IPW$$LD3
1R06I  LINE cuu NOT TRANSPARENT                                   IPW$$CPS
```

		IPW$$CT
1R52I	commandcode LAST OPERAND INVALID	IPW$$CM
		IPW$$CG
		IPW$$CPF
		IPW$$CS
		IPW$$CX
1R52I	commandcode OPERAND ## NEITHER DECIMAL NOR OMITTED	IPW$$CT
1R52I	commandcode OPERAND ## INVALID	IPW$$CT
1R52I	commandcode OPERAND ## MISSING OR INVALID	IPW$$CA
		IPW$$CAC
		IPW$$CB
		IPW$$CC
		IPW$$CD
		IPW$$CF
		IPW$$CG
		IPW$$CH
		IPW$$CI
		IPW$$CJ
		IPW$$CL
		IPW$$CN
		IPW$$CO
		IPW$$CP
		IPW$$CPS
		IPW$$CR
		IPW$$CRE
		IPW$$CS
		IPW$$CT
		IPW$$CX
1R52I	commandcode OPERAND ## NO VALID QUEUE	IPW$$CH
		IPW$$CL
		IPW$$CO
		IPW$$CR
1R52I	commandcode INVALID SPECIFICATION FOR KEYWORD	IPW$$CA
		IPW$$CM
1R52I	commandcode OPERAND ## NOT SPECIFIED AS VALID KEYWORD	IPW$$CA
		IPW$$CLD
		IPW$$CM
1R52I	commandcode INVALID BUFFER SPECIFICATION	IPW$$CS
1R52I	commandcode OPERAND ## NO DEVICE ADDRESS	IPW$$CG
1R52I	commandcode INVALID DESTINATION SPECIFIED	IPW$$CB
1R53I	commandcode INVALID DENSITY	IPW$$CJ
1R54I	commandcode CLASS X INVALID	IPW$$CM
1R55I	commandcode INVALID FILENAME	IPW$$CJ
1R56I	NO LOGICAL UNIT LOGGED ON	IPW$$CI
1R56I	lineaddr INACTIVE	IPW$$CI
1R56I	lineaddr NOT INITIATED	IPW$$CI
1R56I	lineaddr PROCESSING remid	IPW$$CI
1R56I	luname LOGGED ON	IPW$$CI
1R56I	luname LOGGING ON	IPW$$CI
1R56I	luname NOT LOGGED ON	IPW$$CI
1R56I	luname PROCESSING remid	IPW$$CI
1R56I	cuu PROCESSING NODE nodeid	IPW$$CI
1R56I	cuu NODE nodeid SESSION PENDING	IPW$$CI
1R56I	JOB-TRANSMITTER 1=A 2=I.....	IPW$$CI

```
                                                                    IPW$$CG
1R79I  commandcode ERRONEOUS AUTOSTART CARD(S) READ'                IPW$$CS
1R80I  commandcode WARNING: CLASS SPECIFICATION IGNORED             IPW$$CS
1R81I  commandcode MESSAGE OPERAND DOES NOT START WITH QUOTE        IPW$$CM
1R81I  commandcode OPERAND TOO LONG OR NO CLOSING QUOTE             IPW$$CM
1R81I  commandcode MESSAGE TEXT WILL BE TRUNCATED                   IPW$$CB
1R82I  commandcode 'PSETUP' OR 'PRESTART' IN PROGRESS               IPW$$CT
1R83I  PINQUIRE OPERAND NEITHER ALL LINE ADDRESS                    IPW$$CI
1R84I  commandcode DELETION NOT ALLOWED OR IMPOSSIBLE               IPW$$CL
1R85I  commandcode COMMAND NOT ALLOWED FOR REMOTE OPERATOR          IPW$$CM
1R85I  commandcode COMMAND NOT ALLOWED FOR X-PARTITION USER         IPW$$CM
1R86I  PLEASE SPECIFIY DEVICES TO BE SPOOLED                        IPW$$CS
1R87I  PSTART TOO MANY CLASSES, FIRST n PROCESSED                   IPW$$CM
                                                                    IPW$$CS
1R88I  JOB jobname jobnumber CANNOT BE ALTERED                      IPW$$CA
1R88I  NOTHING TO ALTER                                             IPW$$CA
1R88I  NOTHING TO CANCEL                                            IPW$$CC
1R88I  NOTHING TO DELETE                                            IPW$$CL
1R88I  NOTHING TO HOLD                                              IPW$$CH
1R88I  NOTHING TO RELEASE                                           IPW$$CR
1R88I  OK                                                           IPW$$CA
                                                                    IPW$$CH
                                                                    IPW$$CL
                                                                    IPW$$CR
1R89I  PEND VSE/POWER INITIATION NOT COMPLETE                       IPW$$CE
1R90I  commandcode INVALID TASK SPECIFICATION operand               IPW$$CS
1R91I  commandcode TOO MANY OPERANDS, COMMAND REJECTED              IPW$$CM
1R91I  commandcode TOO MANY OPERANDS, FIRST n PROCESSED             IPW$$CS
1R92I  ALLUSER MESSAGE QUEUE IS FULL                                IPW$$CB
1R93I  commandcode REMOTE remid CURRENTLY NOT SIGNED ON             IPW$$CB
                                                                    IPW$$CX
1R93I  commandcode NO SESSION ESTABLISHED FOR lunamexx              IPW$$CP
1R94I  INVALID DEVICE DUPLICATION                                   IPW$$CS
1R95I  LINE cuu NOT SUPPORTED                                       IPW$$CI
                                                                    IPW$$CJ
                                                                    IPW$$CM
1R97I  commandcode COMMAND INVALID DURING SHUTDOWN                  IPW$$CM
1R98I  commandcode INVALID VSE/POWER COMMAND                        IPW$$CM
1R99I  VSE/POWER HAS BEEN TERMINATED                                IPW$$CE
1R99I  VSE/POWER IS IN SHUTDOWN                                     IPW$$CE
1RA0I  job/output jobname jobnum TRANSMITTED TO NODE nodeid...      IPW$$NT
1RA1I  job/output jobname jobnum NODE nodeid UNKNOWN                IPW$$QM
1RA2I  COMMAND FOR NODE node1, NODE node2 NOT CONNECTED             IPW$$MX
1RA2I  NODE nodeid UNKNOWN                                          IPW$$MX
1RA3I  commandcode VSE/POWER NETWORKING NOT SUPPORTED               IPW$$CAC
                                                                    IPW$$CB
                                                                    IPW$$CD
                                                                    IPW$$CF
                                                                    IPW$$CLD
                                                                    IPW$$CN
                                                                    IPW$$CP
                                                                    IPW$$CS
                                                                    IPW$$CX
1RA4I  [commandcode] INVALID NODEID nodeid                          IPW$$CA
```

```
                                                                  IPW$$SE
                                                                  IPW$$LD2
1RE0I  ACF/VTAM NOT STARTED OR INACTIVE                           IPW$$LD4
1RE1I  ACF/VTAM INTERFACE CLOSED FOR NETWORKING                   IPW$$S1
1RE2I  REMOTE SESSION REQUEST FOR NODE nodeid REJECTED, RC=nnnnIPW$$S2
                                                                  IPW$$SE
1RE3I  APPLID FOR NODE nodeid ALREADY DEFINED IN NDT              IPW$$CLD
1V01I  NO SUBTASK AVAILABLE FOR RJE/SNA                           IPW$$SN
1V02I  VTAM OPEN FAILURE RTNCD=nnnn                               IPW$$SN
1V03I  ERROR ON rplrequest RTNCD,FDB2=nn,nn SENSE=yyyy            IPW$$SN
1V04I  RJE,SNA STARTED                                            IPW$$SN
1V05I  RJE,SNA TERMINATED                                         IPW$$SN
1V06I  UNABLE TO LOGON luname RC=nnnn                             IPW$$LH
                                                                  IPW$$LN
                                                                  IPW$$VE
1V07I  ERROR ON rplrequest RTNCD,FDB2=nn,nn SENSE=xxx ON lunameIPW$$IB
                                                                  IPW$$LH
                                                                  IPW$$LN
                                                                  IPW$$MP
                                                                  IPW$$OB
                                                                  IPW$$SN
1V08I  luname BIND PARAMETERS INVALID                             IPW$$LH
1V09I  REMOTE remid LOGGED ON TO applid ON luname                 IPW$$LN
1V10I  RJE,SNA IS IN SHUTDOWN                                     IPW$$SN
1V11I  REMOTE remid LOGGED OFF FROM applid ON luname              IPW$$LF
1V12I  LOGOFF COMPLETED FOR luname                                IPW$$LF
1V13I  LOGOFF FORCED FOR luname                                   IPW$$LF
1V14I  SESSION IS IN SHUTDOWN                                     IPW$$SN
1V15I  NO STORAGE AVAILABLE FOR task                              IPW$$OB
1V16I  NO STORAGE AVAILABLE FOR task FOR luname,remid             IPW$$OB
1V17A  task SUSPENDED FOR FORMS MOUNT                             IPW$$OB
1V18A  REPLY WITH RESTART ON INTERVENTION REQUIRED task           IPW$$OB
1V22I  INVALID command COMMAND                                    IPW$$IB
1V23I  command OUT OF SEQUENCE                                    IPW$$IB
1V24I  task TERMINATED REASON=nnnn FOR luname                     IPW$$IB
                                                                  IPW$$OB
1V25I  EOJ ADDED FOR jobname jobnumber                            IPW$$IB
1V26I  INVALID REMOTE-ID, PASSWORD, OR LUNAME RC=yy               IPW$$LH
1V27I  REMID remid EXCEEDS SESSLIM                                IPW$$LH
1V28I  JOB jobname GETVIS FOR COCB FAILED                         IPW$$OC
1V29I  JOB jobname GETVIS FOR COMPACT TABLE FAILED                IPW$$OC
1V30I  JOB jobname COMPACTION TABEL NOT FOUND                     IPW$$OC
1V31I  JOB jobname NO SPACE AVAIL. IN COMPACT POOL                IPW$$OC
1V32I  JOB jobname INVALID COMPACTION TABLE                       IPW$$OC
1V33I  REMOTE remid OUTPUT FOR NON WRITER WORK STATION            IPW$$SN
1V34I  display of bind parameters                                 IPW$$LH
```

5. DATA AREAS

This section describes the control blocks, buffer areas, save areas and work spaces required by VSE/POWER in addition to the storage layout of the VSE/POWER 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 5-26 to Figure 5-36 in the form of a series of examples.

Most VSE/POWER control blocks and many sections of VSE/POWER 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 VSE/POWER DATA AREAS

The VSE/POWER Partition Storage Layout

The layout of the VSE/POWER partition storage is illustrated in Figure 5-1 .

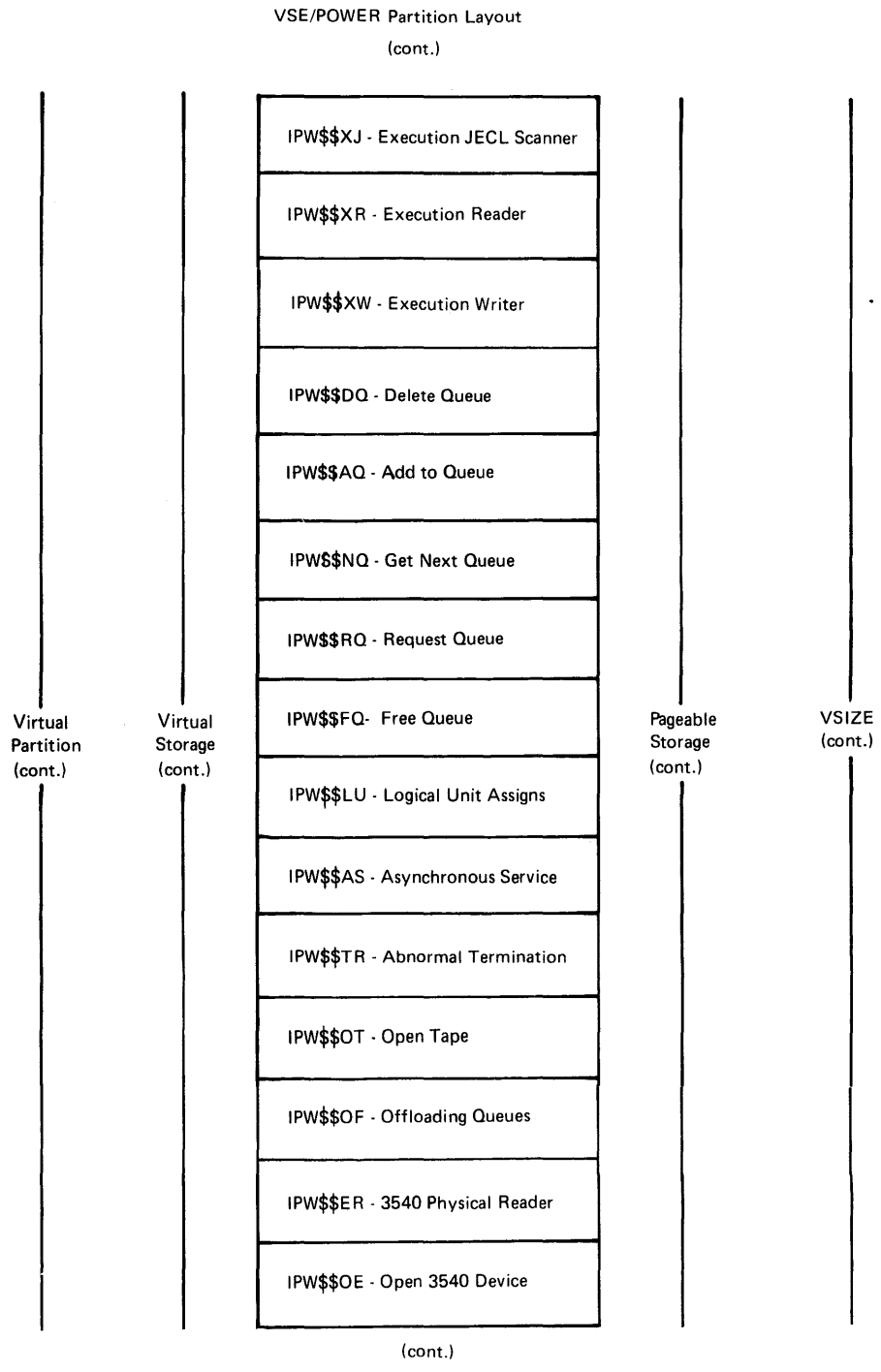


Figure 5-1 (Part 2 of 4). VSE/POWER Partition Storage Layout

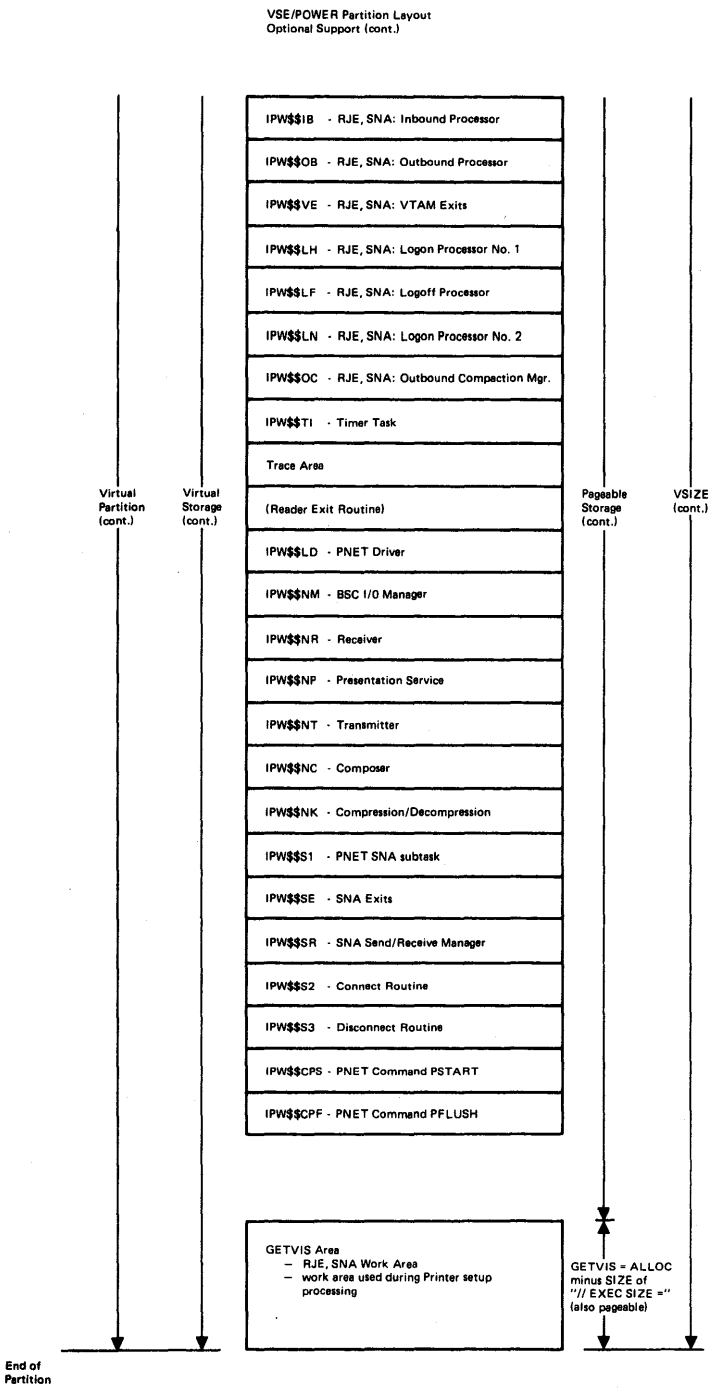


Figure 5-1 (Part 4 of 4). VSE/POWER Partition Storage Layout

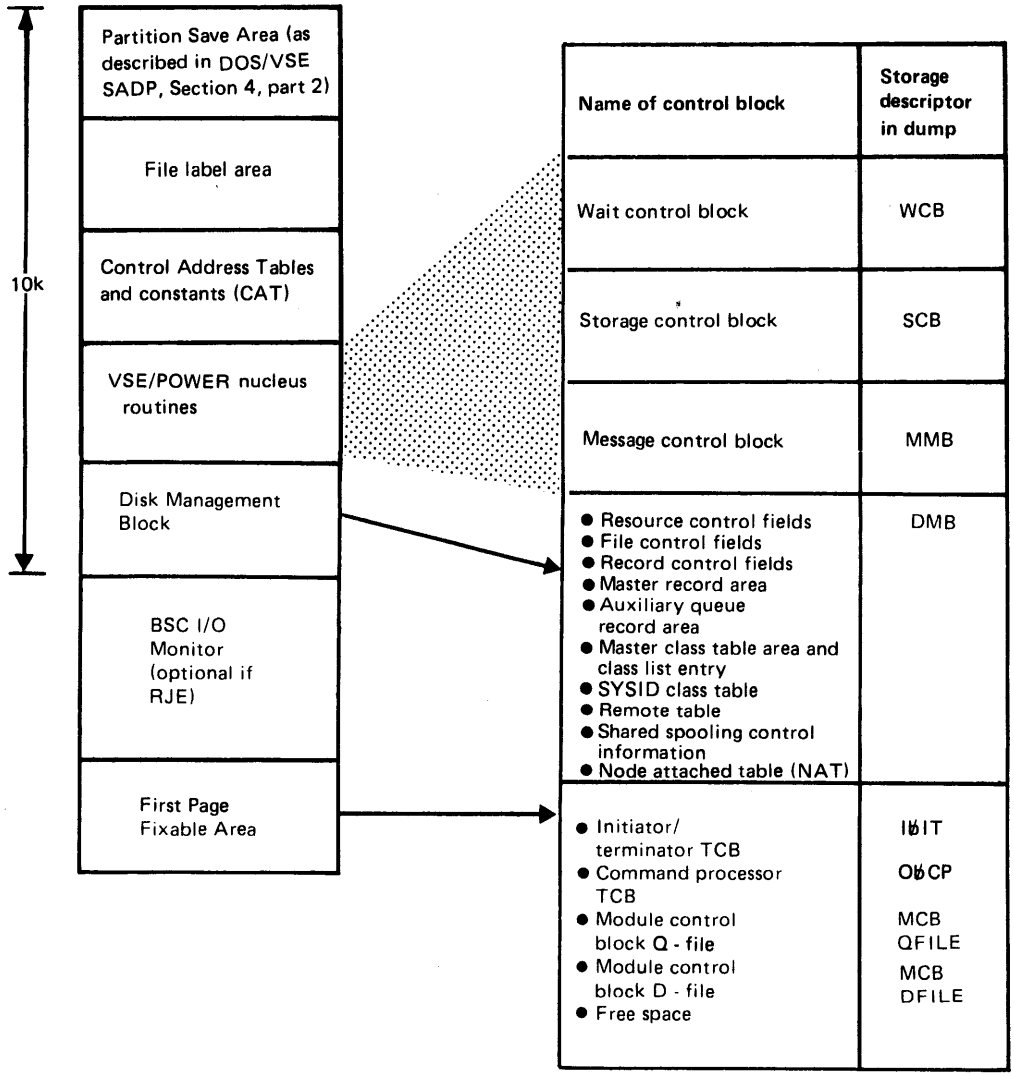


Figure 5-3. Organization of the VSE/POWER Permanent Area with Fixed Control Blocks

Control Blocks Dynamically Allocated in the Fixable Area

These blocks (Figure 5-4) 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 that are eligible to be in the fixable area.

Description of Use	Storage Descriptor
Session Account Record	None
RJE/BSC Line Control Block	None
RJE/BSC Line Manager TCB Fields	None
RJE/SNA Control Block	SNCB
RJE/SNA Message Control Block	MSCB
RJE/SNA Manager TCB Fields	None
Generation Table	GNB
CCB	None
CCW	None
Separator Line Area	None
Diskette Work Space	OEWS
Asynchronous Service Anchor Block	ASWS
Service Request Block	None
Assign/Unassign Work Space	LUWS
TCB Extension Area	None
Print Status Work Area	None
Virtual Storage Control Block	VSCB
PNET Master Control Block	PNCB
Node Control Block	NCB
Trace Information Block	TIB

Figure 5-4 (Part 2 of 2). Control Blocks Dynamically Allocated in the Fixable Area

Description of Area	Storage Descriptor
RJE,SNA Session Control BLock	SUCB
RJE,SNA Logical Unit Control Block	LUCB
RJE,SNA Logon Request Control Block	LRCB
RJE,SNA Work area	WACB
RJE,SNA Remote Control Block	RMCB
Network Definition Table	NDT
Network Receiver Work Area	None
Receiver presentation workarea	NPWA
Network Transmitter Work Area	None
Composer workarea	NCWA
Start-up Account record	None
Transmitter Account record	None
Receiver Account record	None
PNET Account record	None
ACF/VTAM Driver Control Block	VDCB
SNA Session Control Block	SSCB
SNA Request Queue Element	SRQE
Command Processor Work Area	----
SL - Work Area	SLWA
SL - member element	None

Figure 5-5. Control Blocks Dynamically Allocated in the GETVIS Area

64-67	ACEC	20% limit residual capacity
68-6B	ACAC	Current residual capacity
6C-6F	ACMT	Maximum track capacity
70-73	ACLC	Residual capacity on current track
74-77	AC#T	Number of tracks per cylinder
78-79	ACSE	Sector values
7A-7B	ACUH	Upper Head
7C-7D	ACDL	Total account record length
7E-7F	ACPRL	Length of block field plus prefix

- Channel Program (referred to by label ACCH)

80-87	ACSK	Seek CCW
88-8F	ACSS	Set sector or TIC CCW
90-97	ACSH	Search CCW
98-9F	ACTI	TIC CCW
A0-A7	ACWC	Write count CCW
A8-AF	ACWD	Write account data CCW
B0-B7	ACRS	Read sector or not used
B8-C7	ACPM	Chan prog modifiers RDATA and RCOUNT CCW's
C8-CB	ACWA	Virt addr of workspace buffer
CC-D7		Not used

Account File on an FBA Device

Bytes Hex.	Label of Field	Description/Function
00-0F	AFSDF	Storage descriptor (ACB)
10-13	AFEBF	Event control block This ECB is posted when the account file is emptied.
14-17	AFLOF	Extent lower limit
18-1B	AFHIF	Extent upper limit
1C-1F	AFLWF	Lockword

- Command Control Block (referred to by label AFCBF)

20-21	AFCTF	Residual count
22-23	AFCMF	Communication bytes
24-25	AFSTF	Device status
26-27	AFLUF	Logical unit
28		Reserved for LIOCS
29-2B	AFCAF	CCW real address
2C		Reserved for PIOCS
2D-2F	AFLAF	CCW address in CSW
30-33	AFTCB	Save account TCB address
34	AFPBF	PUB device type code
35	AFDTF	DTFPH device type code F = FBA device
36	AFSTATUS	Processing status byte X'80' = process current CI X'40' = IPW$$PF wait on full file

- Extent description block (referred to by label AFEDT)

98	AFMBF	Mask byte
	AFPWF	X'C0' = permit all write commands
	AFPDF	X'04' = permit all diagnostic commands
	AFIWC	X'40' = inhibit all unit commands
99-9B		Reserved, must be zero.
9C-9F	AFBBF	Physical address of first block
A0-A3	AFFBF	Relative displacement of first block (=0)
A4-A7	AFLBF	Relative displacement of last block
A8-AF		Reserved for future use

- Channel Program (referred to by label AFCHF)

B0-B7	AFDFF	Define extent CCW
B8-BF	AFLCF	Locate CCW
C0-C7	AFRWF	Read or write CCW
C0	AFRWFO	Operation code
	AFWRITE	X'41' = CCW write command
	AFREAD	X'42' = CCW read command
C1-C3	AFRWFA	Data address
C4	AFRWFF	Flags
C5		Not used
C6-C7	AFRWFL	Data length
C8-D7		Not used

HOW TO LOCATE: Refer to Figure 6-2 in 6, "Diagnostic Aids"

Field Name	Description	Field Type & Length[3]
EXSIO	Length of SIO table.	BL2
EXTAC	Length of total account record.	BL2
	Reserved.	CL6
EXDJOB	VSE job name from // JOB card.	CL8
EXDUSER	16 bytes user information from // JOB card.	CL16
EXPID	Partition ID in EBCDIC format.	CL2
EXDCANC	VSE cancel code.	BL1
EXTYPE	Type of record; S =job step, L =last step.	CL1
	Reserved.	CL4
EXPHASE	Phase name, taken from // EXEC card.	CL8
EXENDAD	End addr. of active program phase, COMREG.	BL4
EXCPUTM	Processor time in 300ths of a second (see Note).	BL4
EXOVHTM	Overhead time in 300ths of a second (see Note).	BL4
EXALLTM	All-bound time in 300ths of a second (see Note).	BL4
EXSIOTB	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.[4]	
EXSIOTB+m	Overflow byte: normally X'20', but X'30' if more devices are used within a partition than specified by SYSGEN options.[5]	
EXSIOTB +m+1	User account information (provided via user PUTACCT macro).[6]	

[1] Stop time may be higher than the time logged on console to account for VSE/POWER job termination.
[2] Contains the constant 'VSE/POWER-E.A.R' (execution account record) for the VSE/POWER partition execution account record (written during normal shutdown).
[3] C: alphameric, B: binary, P: packed decimal
[4] VSE/POWER will update the SIO tables in the execution account record with the number of I/Os it has intercepted for spooling purposes.
[5] m = Total length of SIO tables (EXSIO).
[6] Maximum length of execution account record = 2008 bytes. (If the account file resides on an FBA device, the maximum length is 1995 bytes.)
[7] The date format may change if the // DATE statement is used.

Figure 5-6 (Part 2 of 2). Execution Account Record

Note:

CPU time: This is the real time used by a job or a job step in the system.

Overhead Time: This is the time for various activities that cannot be charged to a specific program or partition. For example, the time for the calling rou-

ACCOUNT RECORD - LIST

A list account record (Figure 5-7) 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 DSECT for this account record may be obtained from the PACCNT macro by specifying either LIST=YES or ALL=YES.

ACCOUNT RECORD - NETWORK

A network account record (Figure 5-8) is created for every PNET connection or session which had been established, when that connection or session is normally terminated. It contains information relating to all activity on this connection or session. The DSECT for the record may be obtained from the PACCNT macro by specifying PNET=YES or ALL as a parameter.

Field Name	Description	Field Type & Length[2]
NETDTE	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy [1]	CL8
NETSGN	Signon Time (0HHMMSSF; F = sign)	PL4
NETSGF	Signoff Time (0HHMMSSF; F = sign)	PL4
NETNODE	Nodeid of connected node	CL8
NETNPAS	Node Password	CL8
NETPSW	Line Password	CL8
NETICNT	Invalid responses per session	BL2
NETIDEN	Record Identifier (N)	CL1
NETCANC	Cancel Code	BL1
	X'80' Due to operator command (PSTOP, PSTOP EOJ, PEND)	
	X'40' Due to remote SIGNOFF	
	X'20' Due to time-out	
	X'10' Due to line/error session termination	
	X'08' Node stopped due to internal error	
	X'04' ACF/VTAM abnormal termination	
	X'02' ACF/VTAM normal shutdown	
NETLAD	Line Address or SNA	CL3
NETTRAN	Transmission count per session or buffers sent during session	BL4
NETTCNT	Timeout count per session	BL2
NETERR	Error count per session	BL2
ORG NETTCNT	For SNA only	
NETRCVE	Buffers received during session	BL4
NETSOD	Signoff date in same format as NETDTE	CL8
NETLNG	Length of account record	

[1] The date format may change if the // DATE statement is used.
[2] C: alphameric; B: binary; P: packed decimal

Figure 5-8. Network Account Record

ACCOUNT RECORD - READER

A reader account record (Figure 5-10) is created for each read queue entry that is entered into the VSE/POWER system. Whether or not the queue entry has actually been placed in the queue file is indicated by the VSE/POWER cancel code. The record is copied from the first 58 bytes of the corresponding queue record. Reader account records are not created for a writer-only partition.

The DSECT for this account record may be obtained by specifiying READER=YES or ALL=YES in the PACCNT macro.

Note: The TO remote ID is a dummy entry in the Reader Account Record and in the Execution Account Record. It is copied from a FROM remote ID.

Field Name	Description	Field Type & Length[1]
ACDATE	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy). [2]	CL8
ACSTRT	Start time of read (0HHMMSSF; F = sign).	PL4
ACSTOP	Stop time of read (0HHMMSSF; F = sign).	PL4
ACUSER	16 bytes of user information from * $$ JOB statmnt.	CL16
ACNAME	VSE/POWER job name from * $$ JOB or // JOB card.	CL8
ACNUMB	Job number assigned by VSE/POWER.	BL2
ACIDEN	Record identifier (R).	CL1
ACCANC	VSE/POWER cancel code.	BL1
	Reserved.	
RDRADD	Reader device address, or SNA, or PSP, or line address (cuu). With PUTSPOOL.	CL3
RDRFRM	FROM remote ID.	BL1
	Reserved	BL1
RDRICL	Input class.	CL1
RDRIPR	Input priority number.	CL1
RDRNUM	Number of records read (including record added or deleted by a reader exit routine).	BL4
RDRTRK	Number of tracks for input storage.	BL2
RDRLNG	Length of account record	

[1] C: alphameric; B: binary; P: packed decimal
[2] The date format may change if the // DATE statement is used.

Figure 5-10. Reader Account Record

ACCOUNT RECORD - RJE/BSC

A line account record (Figure 5-12) is created for each RJE,BSC user session when signoff or line stop is processed. It is a copy of the first 62 bytes of the line control block.

A copy of the DSECT for this account record may be obtained by specifying BSC=YES or ALL=YES in the PACCNT macro.

Field Name	Description	Field Type & Length[1]
BSCDTE	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy). [2]	CL8
BSCSGN	SIGNON time (0MMMMSSF; F = sign).	PL4
BSCSGF	SIGNOFF time (0MMMMSSF; F = sign).	PL4
BSCUSE	16 bytes user information from the SIGNON command.	CL16
BSCPAS	Line password.	CL8
BSCIRS	Number of invalid responses during transmission(4).	BL2
BSCIDN	Record identifier (T).	CL1
BSCSFC	SIGNOFF code [3]	BL1
	X'01' - Normal SIGNOFF	
	X'02' - SIGNOFF forced due to PSTOP cuu	
	X'04' " - excessive idle time	
	X'08' " - unrecov. I/O error	
	X'10' " - PEND or PSTOP cuu,EOJ	
	X'20' " - real storage shortage	
	X'40' " - PSTOP cuu,KILL	
	X'80' " - line stop at last I/O	
BSCTEC	Terminal error count.	BL1
BSCLAD	Line address.	CL3
BSCRID	Remote identifier.	BL1
	Reserved.	
BSCTRAN	Transmission count per session (1).	CL1
BSCTCNT	Timeout count per session (2).	BL2
BSCERR	Error count per session (3).	BL2
BSCSOD	SIGNOFF date (mmddyy)	CL6
BSCLNG	Length of account record	

[1] C: alphameric; B: binary; P: packed decimal
[2] The date format may change if the // DATE statement is used.
[3] A combination of SIGNOFF codes is possible.

Figure 5-12. RJE,BSC Line Account Record

Note:

Comparing (1) to (2) gives an indication of idle time per session. Comparing (1), (2), and (3) gives an indication of line quality. (1), (2), and (3) are also printed locally at SIGNOFF.

ACCOUNT RECORD - SYSTEM START-UP

A system start-up account record (Figure 5-14) is written whenever VSE/POWER initialization is complete.

The DSECT for this account record may be obtained by specifying SYS=YES or ALL=YES in the PACCNT macro.

Field Name	Description	Field Type & Length[1]
PWRDTE	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy). [2]	CL8
PWRSGN	Startup time	PL4
	Reserved	BL4
PWRVER	Version/modification level	CL4
PWRLEV	Level identifier	CL4
PWRPARSZ	Partition size	BL4
PWRGETSZ	GETVIS size	BL4
PWRRELSZ	Real storage size	BL4
PWRPART	Partition id	CL2
PWRFLAG	Feature flags	CL4
PWRIDEN	Record identifier (U)	CL1
PWRDXTN	Number of data file extents	BL1
PWRDTRK	Number of tracks/blocks - data file	BL4
PWRQTRK	Number of tracks/blocks - queue file	BL4
PWRATRK	Number of tracks/blocks - account file	BL4
PWRLNG	Length of account record	

[1] C: alphameric; B: binary; P: packed decimal
[2] The date format may change if the // DATE statement is used.

Figure 5-14. System Start-up Account Record

ASSIGN/UNASSIGN WORK SPACE

DSECTname: LUDS

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.

Bytes Hex.	Label of Field	Description/Function
00-0F		Storage descriptor
10-47	LUSV	Temporary register save area for the interface between functions.
48-4F	LUGR	Save area for registers 14-15. Used when another function is invoked.
50-93	LUSP	SETPRT parameter list
94-97		Not used

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 3, "Program Organization" of this manual and VSE/POWER Program Logic Manual Part 2. Its format as it is printed in a dump is shown below. An example of a page containing buffer control words is shown in Figure 5-16.

Bytes Hex.	Label of Field	Description/Function
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	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	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 5-16 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.

BUFFER LAYOUT

The control of buffers for PNET is done by buffer management. The layout of the buffers as provided by the function is shown below.

Definition Macro: IPW$DVD BUF=YES

Bytes Hex.	Label of Field	Description/Function
• Buffer header common part		
000-003	BUFNEXT	Next buffer in chain
004-007	BUFREAL	Real (pfixed) address of buffer (BSC)
008-00B	BUFOWN	Address of related TCB
00C-00F	BUFNCBA	Address of related NCB
010-011	BUFDATL	Count of bytes to send or received
012-013	BUFSZ	Buffer size (excluding header)
014	BUFSTAT	Status flag
	BUFFREE	X'80' - Release buffer on send complete
	BUFPOST	X'40' - Post task when buffer sent
	BUFRRPL	X'20' - Buffer contains response RPL (SNA)
	BUFBCBI	X'10' - Send 'ignore BCB' (BSC)
	BUFTERM	X'08' - Terminating buffer (BSC)
	BUFSTCH	X'04' - Status change requested
015	BUFEST1	Data stream status
	BUFRIF	X'80' - RIF sent/received
	BUFPGR	X'40' - PGR sent/received or receiver cancel sent/received
	BUFPRJ	X'20' - NPGR sent/received
	BUFEOF	X'10' - EOF sent/received
	BUFADS	X'08' - Abort transmission
	BUFCMC	X'04' - EOT sent/received
016-017		Reserved
• Data portion BSC		
018-020	BUFDATA	Data portion of BSC TP buffer
018-019	BUFSTRT	Transmission control bytes
01A	BUFBCB	Block control byte
	BUFMLIC	X'80' - MLI control bit
	BUFBRES	X'20' - Reset expected block sequence CNT
	BUFBBYB	X'10' - Bypass block sequence validation
01B-01C	BUFFCS	Function control sequence
01D	BUFRCB	Record control byte
01E	BUFSRCB	Subrecord control byte
01F	BUFSCB	String control byte
020	BUFEOB	End-of-block RCB
016-017		Reserved
• Data portion SNA		
018-01A	BUFRIDD	Decompressed RID of 1st record
01B		Unused

CANCEL CODES OF VSE/POWER

Figure 5-17 shows the VSE/POWER cancel codes that appear in several VSE/POWER records.

Cancel Code	Condition
X'10'	Normal end of VSE/POWER job or task (see Note 1).
X'20'	PCANCEL was issued.
X'30'	PSTOP command was issued (see Note 2).
X'40'	PFLUSH command was issued.
X'50'	PDELETE was issued.
X'60'	VSE/POWER job was flushed via RDREXIT.
X'70'	VSE/POWER job canceled due to I/O error.

Figure 5-17. Cancel Codes of VSE/POWER

Note:

1. Although no abnormal VSE/POWER termination occurred, the VSE/Advanced Functions jobs associated with the queue entry could have been canceled via VSE/Advanced Functions.

2. The PSTOP cancel code is not stored in an account record if the EOJ option was specified in the PSTOP command.

COMMAND CONTROL BLOCK (CCB)

Definition Macro: IPW$DCB

The layout of a Command Control Block is shown below.

Bytes Hex.	Label of Field	Description/Function
00-01	CBCT	Residual count
02	CBC1	First communication byte (see flags A)
03	CBC2	Second communication byte (see flags B)
04	CBSD	Device status byte (see flags C)
05	CBSC	Channel status byte
06	CBLC	LUB class (see flags D)
07	CBLN	LUB number within class
08	CBLI	LIOCS communication byte
09-0B	CBCA	CCW address
0C	CBPI	PIOCS communication byte
0D-0F	CBCS	CCW address in CSW
10	CBNX	First entry outside CCB
Flags B	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
Flags C	UE	X'01' = unit exception
	UNCK	X'02' = unit check
Flags D	EXR	X'80' = EXCP real
	PRU	X'01' = programmer unit
	SID	X'20' = sense information desired

		should be offloaded
65	SWFLAG2	Flag byte 2
	SWREPLY	X'80' turned on by caller of message routine to indicate answer required
	SWAUTOST	X'40' turned on to indicate that the initiator task must be prompted to supply address of device to be spooled
	SWERROR	X'20' turned on when an error message should be issued to central operator even when AUTOSTART is in progress
66	SWFLAG3	Flag byte 3
	SWINDEV	X'80' indicates that the device type designated in the PSTART command is inconsistent with the task type
	SWDELAY	X'40' indicates that a warning message must be delayed, to avoid the disappearance of the appropriate TCB
	SWDEL	X'20' turned on by PALTER processor when the queue set to be altered must be deleted and then added in its correct place in the class chain, rather than rewriting the queue record in the current place
	SWNOCCO	X'10' turned on by PALTER processor when attributes other than #-of-copies should be changed
	SWUSER5	X'08' turned on by calling routine to indicate indicate that reg 5 should be used as TCB pointer
67	SWFLAG4	Flag byte 4
	SWMSG	X'80' turned on by PDISPLAY processor if messages are to be suppressed
	SWPBCST	X'40' turned on by PBRDCST processor if messages have to be truncated
	SWF4all	X'20' turned on by PINQUIRE processor if 'All' is specified as 1st operand
	SWF4NOTH	X'10' turned on by PINQUIRE processor if nothing to display
68-6B		Reserved for future use

- Register save areas

6C-9B	CPWRRS	Register save area for root phase
9C-CB	CPWCRS	Register save area for command proc
CC-FB	CPWSRS	Register save area for subroutines

- General work area

  The following 64 bytes are used as general purpose work area, which may be broken down into fields as it is required.

FC-11B	CPWGW1	Work area 1
11C-13B	CPWGW2	Work area 2

- Message output area

228		Flag byte 2
229		reserved for future use
22A-239		Operand
23A-23B	OP4HEX	Hexadecimal value of operand
23C-23D	OP4DEC	Decimal value of operand
23E-255	OP5	Operand 5
23E	OPLEN5	Length of operand contents
23F	OPSW5	Flag byte 1
240		Flag byte 2
241		reserved for future use
242-251		Operand
252-253	OP5HEX	Hexadecimal value of operand
254-255	OP5DEC	Decimal value of operand
256-26D	OP6	Operand 6
256	OPLEN6	Length of operand contents
257	OPSW6	Flag byte 1
258		Flag byte 2
259		reserved for future use
25A-269		Operand
26A-26B	OP6HEX	Hexadecimal value of operand
26C-26D	OP6DEC	Decimal value of operand
26E-285	OP7	Operand 7
26E	OPLEN7	Length of operand contents
26F	OPSW7	Flag byte 1
270		Flag byte 2
271		reserved for future use
272-281		Operand
282-283	OP7HEX	Hexadecimal value of operand
284-285	OP7DEC	Decimal value of operand
286-29D	OP8	Operand 8
286	OPLEN8	Length of operand contents
287	OPSW8	Flag byte 1
288		Flag byte 2
289		reserved for future use
28A-299		Operand
29A-29B	OP8HEX	Hexadecimal value of operand
29C-29D	OP8DEC	Decimal value of operand
29E-2B5	OP9	Operand 9
29E	OPLEN9	Length of operand contents
29F	OPSW9	Flag byte 1
2A0		Flag byte 2
2A1		reserved for future use
2A2-2B1		Operand
2B2-2B3	OP9HEX	Hexadecimal value of operand
2B4-2B5	OP9DEC	Decimal value of operand

- Dummy TCB definition

  Bytes 2B8-42F are reserved for the dummy TCB definition. The details are shown in the TCB definition.

474-47F	CPWCLAR	Argument list
474-479	CPWCLAS	Class(es) up to 4 classes
47A	CPWCLIX	Used for translation of class char
47B	CPWCL#C	Max number of valid classes - 4 for execution-reader tasks - 4 for physical-writer tasks - 1 for physical-reader tasks
47C-47E	CPWCLTI	Task type ('LST', 'RDR' or 'PUN')
47F	CPWCLDF	Default class

- Variables used for subroutine 'QRINSPCT'

  The following arguments are set up by the calling routine in order to determine if a queue set meets all applicable criteria. When an argument contains hex zeros it is assumed to be not important and will not be checked.

480-4BF	CPWQARG	Argument list
480-487	CPWQAJN	Job name
488-489	CPWQAJ#	Job number
48A	CPWQACL	Class associated to queue set
48B	CPWQAF1	Flag byte 1
	CPWQAF1S	X'80' - Job suffix number must be present
48C	CPWQABIN	Binary RJE user-id (0 for central)
48D-48F	CPWQADEC	Printable decimal RJE user-id
490	CPWQAGLN	Length of generic-supplied job name
491-498	CPWQAGJN	Generic job name
499-49F		Reserved
4A0	CPWQACDP	Current disposition
4A1	CPWQACPY	Current priority
4A2	CPWQACSY	Current sysid
4A3		Reserved for future use
4A4-4AB	CPWQACNN	Current target node name
4AC-4B3	CPWQACUS	Current user id
4B4-4B7	CPWQACFI	Current forms id (FNO)
4B8-4BF		Reserved for future use

- Variables used for subroutine 'VQEUEID'

4C0-4C3	CLASSPTR	Points to start of class table
4C4-4C5	CLASSLC	Number of scans to be performed
4C6	CLASSQID	Queue-record identifier
4C7	CLASSPCB	Queue processing flag

- Variables used by operand formatting routine

4C8-4CB	CMNDPTR	Points to command-table entry
4CC-4CF	DELIMPTR	Address of current delimiter
4D0-5CF	TRTTAB	Translate-and-test table
5D0-5D1	MAXOP	Max number of operands allowed
5D2-5D3	OPNUM	Current no. of operands in process
5D4-5DB	SVEOP	Operand save field

- Variables used for queue manipulation

The following fields represent the argument list which will be passed to the print status task to perform the appropriate display function.

5F4-643	CPWDARGL	Argument list
5F4	CPWDID	Identification field
		C'P' for PNET display
		C'D' for default display
5F5	CPWDTOID	Destination of report (RJE remid)
5F6-5F7	CPWDLU	Programmer logical unit of the printer
5F8-5FB	CPWDECB	Address of ECB to be posted at completion
5FC-604	CPWDFNM	From node name and qualifier
605-60C	CPWDUID	From user id / remote id
60D	CPWDNMRF	Flag byte from NMR copied
60E-60F		reserved for future use
610-613	CPWDBS	Begin scan indicator (start of queue scan)
614	CPWDQID	Queue processing flags
615-616	CPWDJN	Job number
617	CPWDBIN	Remote id (binary)
618-61A		reserved
61B	CPWDFLG	Flag byte
	CPWDREM	X'80' PDISPLAY RJE
	CPWDHLD	X'40' PDISPLAY HOLD
	CPWDFRE	X'20' PDISPLAY FREE
	CPWDLOC	X'10' PDISPLAY LOCAL
61C	CPWDGJL	Length of generic job name
61D-624	CPWDGJN	Generic job name
625-62C	CPWDJOB	Job name
62D-634	CPWDTNN	Target node name
635	CPWDCDP	Current disposition
636	CPWDCPY	Current priority
637	CPWDCSY	Current SYSID
638-63B	CPWDCFI	Current forms id (FNO)
63C-643	CPWDCUS	Current user id
644	CPWDCLS	Job class
645-647		Reserved

- Variables used by PDISPLAY PNET command processor

The following fields together with the common part of the argument list are passed to the print status task to perform the PDISPLAY PNET function.

610-613	CPWDPPTR	Pointer to specified nodeid
614	CPWDFLG1	Flags
	CPWDPOWN	X'80' Own node display
	CPWDPLIN	X'40'Link display request
	CPWDPNID	X'20' Specific node display request
	CPWDPALL	X'10' Display all nodes request
615		unused
616-61D	CPWNODID	Node name

- Variables used by PALTER command processor

COMMUNICATOR INFORMATION BLOCK (CIB)

The communciator information block controls all access to the Notify message queue. It is created by the VSE/POWER Notify processor at initialization time if 'NTFYMSG=' had been specified in the POWER generation macro.

It is addressed by field CACI in the CAT.

Definition Macro: IPW$DCI

Bytes Hex.	Label of Field	Description/Function
000-010	CIBSD	Storage descriptor
010-017	CIBWW	Work area
018-01B		Reserved
01C-020	CIBLW	Lockword
020-04F	CIBSV	Register save area (Register 14 - Register 9)
• General section		
050-053	CIBTCB	Address of the TCB of the notify processor
054-055	CIBMM#	Maximum number of messages allowed in queue
056-057	CIBLMC	Lost message count
058	CIBFLAG	Flag byte 1
	CIBIDNT	X'80' successful identification done
059	CIBACT1	Action byte
	CIBICCF	X'80' - set up connection to VSE/ICCF
	CIBDSNX	X'40' - set up connection to DSNX
05A-05B		Reserved
• VSE/ICCF subsystem section		
05C-05F	CIBIMSG	Address of first message in queue
060-063	CIBIMTL	Àddress of tail pointer for message queue
064-067	CIBIMBS	Address of message being sent
068-06B	CIBICM#	Current number of messages in the queue
06C	CIBIFLG	Flag Byte 1
	CIBISIP	X'80' - Send in progress
	CIBICON	X'40' - Connection completed
	CIBIQIH	X'02' - Queueing of new messages inhibited
	CIBIQNE	X'01' - Message added to queue
06D-06F		Reserved
070-073	CIBIXPCC	Address of XPCCB
• VSE/DSNX section		
074-077	CIBDMSG	Address of first message in queue
078-07B		Address of tail pointer for message queue
07C-07F	CIBDMBS	Address of message being sent
080	CIBDFLG	Flag Byte 1
	CIBDSIP	X'80' - Send in progress
	CIBDCON	X'40' - Connection completed
	CIBDQIH	X'02' - Queueing of new messages inhibited

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 VSE/POWER partition, used to link the component routines of the VSE/POWER subsystem during execution. The format of this table as it is printed in a dump is shown below.

Register 10 always points to the beginning of the VSE/POWER partition. The displacements shown in this DSECT are relative to the beginning of the VSE/POWER partition, i.e. the fields may be found by using register 10 as a base register.

Bytes Hex.	Label of Field	Description/Function
140-14F	PASD	Storage descriptor (CAT)
150-153	PAEB	VSE/POWER master ECB (refer to Appendix B)
154-157	PAPA	Start address VSE/POWER partition
158-15B	PAFA	Start address fixable area
15C-15F	PAVA	Start address pageable area
160-163	PAEN	End address VSE/POWER partition +1
164-167	PALS	Start address LTA
168-16B	PALE	End address LTA
16C-16F	CAPB	Address of VSE/POWER PIB
170-171	CATI	VSE/POWER task id
172-173	CATS	Task id of timer subtask

- External interface address

174-177	CAAI	Attention interface
178-17B	CAPF	Page fault appendage
17C-17F	CAHR	Hot reader routine
180-183	CACE	RJE CE appendage
184-187	CA00	SVC 0 appendage
188-18B	CA90	SVC 90/91 appendage

- External address for BSC nucleus functions

18C-18F	CABM	RJE,BSC monitor; nucleus function entry point

- In-core reader cross-partition XECB information

190-193	ICXP	Internal reader XECB
194		Not used
195-197	ICTA	XECBTAB ADDR of ICR XECB

- Spool/command manager cross-partition XECB

198-19B	SMXP	Spool/command MGR XECB
19C		Not used
19D-19F	SMTA	XECTAB ADDR of SPM XECB selecting the internal RDR task and/or the spool/command

- Spool/command manager LST task

210-213	CAD6	MCB data file module 5
214-217	CAL7	MCB private SSL
218-21B	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).

21C-21F	TMCI	The task is inactive, task not selected. Branch to TM10
220-223	TMCP	Page fault in process, task not selected. Branch to TM10
224-227	TMCO	Wait for operator, task not selected. Branch to TM10
228-22B	TMCL	Wait on locked resource, test lockword. Branch to TM30
22C-22F		Not used
230-233	TMCM	Wait on multiple posting, test control blocks. Branch to TM50
234-237	TMCQ	Wait on class table posting, test control blocks. Branch to TM50
238-23B	TMCC	Wait on single posting, test control block. Branch to TM80.
23C-23F	TMCS	Wait on space posting, test control blocks. Branch to TM80
240-243	TMCD	Immediate dispatch, dispatch the task. Branch to TM90
244-247	TMCW	Wait state. Used for WCB only wait routine Branch to TM20.
248-24B	TMCR	The task is running, re-selection address
24C-24F	TMCB	Wait on RJE,BSC or PNET event, test control block. Branch to TMB0.
250-253	TMCT	Reserved

- Permanent TCB address

254-257	TATM	Wait control block
258-25B	TAOC	Command processor TCB
25C-25F	TAIT	Initialization/termination TCB
260-263	TALM	Line manager TCB
264-267	TASP	Spool manager TCB address

- Task Control Address Table.

  The following table is used by the task initiation and termination routines to determine the position in the task list at which a new task of a given type is to be inserted. The first byte of each entry contains an alphameric character identifying the type of task to which the entry relates. The remaining three bytes contain the address of the TCB for the most recently attached task of that type. If no such task exists the address contained in the entry is that of the TCB of the currently attached task which must precede any new task of the designated type.

- Writer task phases

29C-29F	CAPP	Physical punch
2A0-2A3	CAPL	Physical list
2A4-2A7	CAGD	Get data record function
2A8-2AB	CALW	Logical writer

- Execution processor phases

2AC-2AF	CAXJ	JECL analysis
2B0-2B3	CAXR	Execution reader
2B4-2B7	CAXW	Execution writer

- Queue management phases

2B8-2BB	CADQ	Delete from queue function
2BC-2BF	CAAQ	Add to queue function
2C0-2C3	CANQ	Get next-from-queue
2C4-2C7	CARQ	Reserve queue function
2C8-2CB	CAFQ	Free queue function

- Miscellaneous phases

2CC-2CF	CALU	LUB/PUB update function
2D0-2D3	CAAS	Asynchronous service routine
2D4-2D7	CATR	Task terminator
2D8-2DB	CAOT	Open tape routine
2DC-2DF	CAOF	Offload queue routine
2E0-2E3	CAER	3540 physical reader
2E4-2E7	CAOE	3540 open routine
2E8-2EB	CASY	SYSIN tape support
2EC-2EF	CAPS	Print status report
2F0-2F3	CAIC	Invoke command processor function
2F4-2F7	CARY	Queue-file recovery
2F8-2FB	CAAT	Abnormal termination program
2FC-2FF	CAMS	Message handler
300-303	CAMX	Message distributor
304-307	CA$M	Message definition module
308-30F		Reserved

- Spool management option

310-313	CASF	Spool manager

- Accounting phases

314-317	CAPA	Put account function
318-31B	CAGA	Get account function
31C-31F	CASA	Save account function

- Notify and source library include support

320-323	CANY	Notify support
324-327	CASL	Get SSL function

- Spool management used fields

420-421	IPIK	In-core reader user's PIK
422-423	SPIK	Spool managers user's PIK

- Service Routine Branch Table (continued).

  The branch instructions are used to transfer control from service routine macro instructions to the appropriate service code.

424-42B	VS50	Release virtual storage
42C-433	VS90	Unchain element
434-43B	SR00	Set remote mask routine
43C-443	TZ00	Get trace entry routine
444-45B		Reserved for future use

- 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.

45C-45F	BLBF	Data buffer - set by INIT (amount of storage required to accommodate the data block)
460-463	BLDB	Data block - set by INIT (size of record written to disk)

- Address of RJE,BSC Control Blocks

464-467	CALC	Line control block address
468-46B	CART	RJE,BSC remote table address
46C-46F	CALT	Line table address
470-47F		Reserved for future use

- Statistical information (refer to Appendix E)

480-481	NRRE	Highest BSC remote ID
482-483	NRLI	Number of BSC lines
484-487	NRTR/NBLK	Total number of tracks/blocks data file for C-K-D/FBA devices
488-48B	NRTW	Number of times waiting for storage
48C-48F	NRTV	Number of times waiting for virtual storage
490-493	NRPG	Total number of pages allocated
494-497	NRPC	Current number of pages allocated
498-49B	NRPM	Maximum number of pages allocated
49C-49F	NRTC	Current number of tasks
4A0-4A3	NRTH	Maximum number of tasks
4A4-4AB		Reserved for future use

DATA SET CONTROL BLOCK

The data set control block is created by the network receiver for each queue set to be allocated. Its contents are:

1. Spool control information
2. Queue set characteristics

Bytes Hex.	Label of Field	Description/Function
000-00F	DSDESCR	Storage descriptor
010-013	DSNEXT	Address next DSCB entry
014-02B	DSTCDAST	Data file status field
014-01B	DSTCBDW	• Data file seek address (MBBCCHHR)
01C-01F	DSTCBDA	• Real address logical data buffer
020-023	DSTCBDV	• Address logical data buffer
024-027	DSTCBBC	• Residual block count
028-02B	DSTCBPR	• Previous record address
02C-03B	DSTCQFST	Queue file status fields
02C-033	DSTCBQW	• Queue file seek address (MBBCCHHR)
034-037	DSTCBQA	• Real queue space address
038-03B	DSTCBQV	• Queue space address
03C-068	DSQREC	DSCB characteristics field
03C-03D		• Not used
03E	DSQRPY	• Job priority
03F	DSQRQI	• Queue record identifier
040	DSQRDP	• Disposition
041	DSQRCL	• Class
042	DSQRNC	• Number of copies
043-046	DSQRFI	• Forms ID
047-04A	DSQRCP	• Compaction table name
04B-058	DSQR3800	• 3800 characteristics
04B-04E	DSQRFL	• Forms-overlay identifier
04F-056	DSQRCG	• 8 copy groups
057	DSQRGI	• Copy group index
058	DSQRPS	• Burst mode indicator
059-060	DSQRTN	• Target node name
061-068	DSQRTU	• Target user ID

- Record control fields.

  They contain information used to read and write records to and from the master record area and auxiliary queue record area.

  The address XXXXXXXX contains for C-K-D device(s): MBBCCHHR and for FBA device(s): M0O0NNNN.

78-7F	QCMW	Master record address (XXXXXXXX)
80-83	QCMA	Real master area address
84-87	QCMV	Virtual master area address
88-8F	QCQW	Queue record address (XXXXXXXX)
90-93	QCQA	Real auxiliary queue area address
94-97	QCQV	Virtual queue record area address

- VSE/POWER communication area.

  The master record is written as the lastt physical record within the queue file extent. During VSE/POWER 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.

98-9F	MRDY	Date These eight bytes contain the date of VSE/POWER execution in the format chosen at system generation (dd/mm/yy or mm/dd/yy).
A0-A3	MRST	VSE/POWER start time

- Switch bytes

  The following 6 switch bytes preserve the options established by the VSE/POWER user at the time he generated his version.

A4	MRSL	Source library switch This byte contains a single alphabetic character representing the source sublibrary to be searched.
A5	MRJA	Job accounting switch This byte contains a single alphabetic character The character A indicates that VSE/POWER job accounting is required; a blank character indicates that VSE/POWER accounting is not required.
A6	MRPP	Pause punch switch
A7	MRLG	LOG option switch (set to character L if JLOG=YES and blank if JLOG=NO)
A8		Reserved for future use
A9	MROP	Option byte X'01' - 3540 feed option X'02' - Multiple channel 12 option (see POWER macro)

DC-DF	PVBS	Records before segmentation
E0-E3	PVBM	Records before message
E4-E7	PVBN	Records before next message

- The next 16 bytes contain the timer task values for shared spooling:

E8-EB	MREB	Shared spooling subtask ECB
EC-ED	MRT1	Interval T1 (time slice)
EE-EF	MRT2	Interval T2
F0-F1	MRT3	Interval T3 (polling time)
F2-F3	MRT4	Interval T4
F4-FC		Reserved
FD	MRSY	SYSID of own system
FE	MRSO	Shared spooling option byte
FF		Reserved

- Auxiliary queue record area (216 bytes). This area is required as a work space for an additional queue record, see the description of the Queue Record Area (QRA). For example, for updating class chain addresses during the add to queue function. The first part (136 bytes) of the Q record contains body fields (information pertinent to this particular queue entry and the user job which created it).

100-187	QCBF	Body fields of queue records
100-107	QCDY	Date in format specified at SYSGEN (mm/dd/yy or dd/mm/yy)
108-10B	QCST	Operation start time, in packed decimal (0HHMMSSF; F = sign)
10C-10F	QCET	Operation end time (0HHMMSSF; F = sign)
110-11F	QCUI	16 bytes user information
120-127	QCNM	Job name Job name associated with this particular VSE/POWER or VSE/Advanced Functions job. If no job name is provided by the user the default value AUTONAME is set into this field.
128-129	QCNO	Job number contains a binary number assigned to the job upon its entry into the system and thereafter available for further identification of jobs with a common job name.
12A	QCQI	Queue record identifier R = read queue record L = list queue record P = punch queue record F = free queue record D = dummy queue record
12B	QCCN	VSE/POWER cancel codes

Cancel Code	Condition
X'10'	Normal end of VSE/POWER job or task
X'20'	PCANCEL has been issued
X'30'	PSTOP has been issued [4]
X'40'	PFLUSH has been issued

		within input-related queue records.
154-157	QCBS	Number of records before segmentation (count driven segmentation)
158-15B	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.
15C-15F	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.
160-161	QCER	Physical 3540 device address (packed)
162-163	QCJ#	Saved job number for accounting
164-167	QCCP	Compaction table name

- 3800 printer control information

168-16B	QCFL	Forms overlay identifier
16C-173	QCCG	Copy groups
174	QCTC	Total number of transmission
175	QCCI	Current copy group index (restart purposes)
176	QCPS	Paper status (3800 only)

- Continuation of general part

177	QCOP	Option byte X'20' - No separator pages between copies
178-17F	QCPW	Password
180-181	QCOJ#	Original job number
182	QCSID	Sysid of target processor
183		Reserved for future use
184-185	QCRL	maximum record length
186-187		Not used

- The second portion (48 bytes) of the queue record contains control fields (information relating to the status of the queue record and to its position within the VSE/POWER queues).

188-1B7	QCCF	Control portion of the queue record
188	QCXS	Execution switch X = job in execution b = job not in execution
189	QCFS	First in set switch
18A	QCSG	Segmentation type (not used)
18B	QCSY	Sysid of system processing the queue record
18C	QCS1	Control flag byte 1
18D-197		Reserved
198-19F	QCNS	Next record in set. (XXXXXXXX) M = index in module control block address

1F4-1F7	NRQR	Total number of queue records
1F8-1FB	NRQF	Number of free queue records
1FC-1FF	NRQM	Maximum number of queue records used

- Account-file values.

200-207	MRAS	Last record address account file (XXXXXXXX)
208-209	MRSE	Sector values
20A-20B		Reserved
20C-20F	MRCF	Free space in current control interval or residual capacity of current track
210-213	MRCC	Current residual capacity

- Free queue pointers.

214-21B	MRQF	First record in free queue (XXXXXXXX)

- Master class table area. Defines the status of the VSE/POWER queues.

21C-3D7	QCCT	Master class table (without XMT queue entries)
21C-2AF	CTRT	Reader class area (37 entries, that is, 1 dummy entry and 36 entries 0-9 and A-Z)
2B0-343	CTLT	List class area (37 entries, that is, 11 dummy entries and 36 entries A-Z)
344-3D7	CTPT	Punch class area (37 entries, that is, 11 dummy entries and 36 entries A-Z)
3D8-3DF	CTXT	XMT class area (2 entries, that is 1 entry for RDR and 1 entry for LST and PUN)

Each area entry in the DMB 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 queue set in this class chain.

- Relative record number of first queue record in last 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 5-26 to Figure 5-36 for an illustration of this relationship.) Entries in the table also act as ECBs for the class chains (refer to Appendix B, "Summary of ECB Usage" on page B-1).

- Shared spooling Control Words

3E0	SSID	System-id owning queue file
3E1	SSF1	Flag byte 1 X'80' - Queue file shared X'40' - Account file shared
3E2	SSF2	Flag byte 2
3E3	SSAC	System-ID owning account file
3E4-3E7	SSWK	Work-to-do ECB
3E8-3EB	SSNW	No-work-to-do ECB

DSECTS FOR ACCOUNTING (A-FILE ON FBA)

Definition Macro: IPW$DJK

This macro maps 4 DSECTs:

- Sequential file header
- Control interval definition field (CIDF)
- Record definition field (RDF)
- I/O area for read device characteristics

The formats are as follows:

Bytes Hex.	Label of Field	Description/Function

- Sequential file header

00-01	HEADBL	Block descriptor. Contains length of the physical block, including its own length.
02-03		Reserved
04-05	HEADRL	Record descriptor. Contains length of the logical record, including its own length
06-07		Reserved

- Control interval definition field

00-01	CIDFA	Start address of the free space in the control interval
02-03	CIDFL	Length of free space in the control interval

- Record definition field

00	RDFF	Flags (always zero)
01-02	RDFL	Contains the length of the corresponding record

- I/O-area for Read device characteristics

00		Operation mode
01		Features
02		Device class
03		Unit type
04-05	RDCBLOCK	Physical record size
06-09		No. of blocks
0A-1F		Reserved

A0-A7	GNRE	RDREXIT name
A8-A9	GN#M	Maximum number of messages in Notify message queue
AA-B1	GNNT	Network Definition Table name
B2-B9	GNPR	PNET Readerexit name
BA	GNRJ	RJEBSC specification
BB-C2	GNMT	Default member type
C3		Reserved

- Work area for EXTRACT macro. Referred to by label GNWE. The EXTRACT macro (SVC98) will be issued from the VSE/POWER load routine which loads IPW$$IP. Information about the VSE/POWER partition will be saved in following fields.

C4-C7	GNPB	Partition begin address
C8-CB	GNPE	Virtual partition end address excluding GETVIS
CC-CF	GNPG	Virtual partition end address including GETVIS
D0-D3	GNFX	Real allocated storage in K-bytes
D4-D7	GNFC	Real allocated storage in number of pages
D8-DB	GNGB	GETVIS area begin address
DC-DF	GNGE	GETVIS area end address
E0-E7	GNIN	IPW$$I1 phase name

- During initialization the IPW$$IP phase name is overlaid by the following fields:

E0-E3	GNRM	Address of remote control block (RMCB)
E4-E7	GNSS	Address of SNA unit control block (SUCB)
E8-F4	GNLT	LTAB
F5-F7	GNLU	Length of LU table. Accumulated length of LU= in PRMT macro.

- SNA information

F8-F9	GNTT	Table length plus BSC plus SNA
FA		Reserved
FB	GNAL	Length of ACB password
FC-103	GNAP	ACB password
104	GNSU	Maximum number of SNA logical units
105	GNSR	Number of SNA remotes
106	GNFR	First SNA remote ID
107	GNHR	Last SNA remote ID
108	GNVA	Length of APPLID for VTAM
109-110		APPLID for VTAM
111-113		Reserved
114-117	GNLA	Address of first line block

Variable (depending on number of lines (GNNL))	BSC line table entries of 16 bytes each
Variable (depending on number of remotes (GNNR))	BSC remote block entries of 28 bytes each

INITIALIZATION PROCESSOR WORK AREA (IP)

Definition Macro: IPW$DEF IWK=MAP

This area contains addresses and information that are used in communication among the initialization root phase and various initialization phases. It is located in the initialization-processor work phase (IPW$$IP).

Bytes Hex.	Label of Field	Description/Function
00-2F	IWKRSA	Register save area
30-33	IWIPEA	End address of root phase
34-37	IWPPEA	End address of last phase
38-3B	IWMAXA	Maximum address for IPLOAD
3C-3F		Reserved
40-41	IWLALN	Length of overlay area
42-89	IWMSGAR	Message I/O area
8A-D9	IWCDIN	Card read-in area
DA	IWABSW	Abnormal-warm-start switch X'C1' A - abnormal-warm-start indicator
DB	IWFLG1	IP flag byte 1 X'80' - Other-system-up indicator X'01' - Request termination X'02' - Single load affected X'04' - Load routine cancelled X'08' - Area too small (IPLOAD) X'10' - Feature to be loaded
DC-DF	IWGENA	Address of original gen table
E0-E3		Reserved

- DTF's Address Table

E4-E7	IWQFILE	Address of queue-file DTF
E8-EB	IWDFILE	Address of data-file DTF
EC-EF	IWAFILE	Address of account-file DTF
F0-F3	IWINPTF	Address of SYSIPT DTF
F4-F7	IWPWRLK	Address of queue-file DTL control block

Subroutine Address Table

F8-FB	IWUPDT	Address of update-Q-record address
FC-FF	IWSUPH	Address of set-up-DTFPH routine
100-103	IWPRST	Address of print-status-report routine
104-107	IWFMCB	Address of format-MCB routine
108-10B	IWCOLD	Address of common load routine
10C-10F		Reserved
110-113	IWGENT	Address of VSE/POWER generation table
114-117	IWMCBC	Address of skeleton MCB (C-K-D)
118-11B	IWMCBF	Address of skeleton MCB (FBA)

- Local directory list for CIL

11C-143	IWGENL	Directory list for core image library

LOGICAL DATA RECORD AREA (LDA)

Definition Macro: IPW$DDR

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 LDA 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.

The format of a logical data record is as follows.

Bytes Hex.	Label of Field	Description/Function
	DRDS	Definition of this dummy section
00-01	DRRL	Logical record length. This field contains the length of the data-record text without preceding fields like DRRL, DRGP, DRGP2, DRGP3, and DRCC, that is in fact the length of DRDT field.
02	DRGP	General purpose byte: X'00' normal record X'01' line print/card move data 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 X'40' extended record
03	DRCC	Command code. Indicates command code for output list/punch device or 00 when input record or spooled-account record.
04	DRG2	General purpose byte 2 X'80' job header record X'40' data set header record X'20' job trailer record
05	DRG3	General purpose byte 3 X'80' extended record begin X'40' extended record middle X'20' extended record end
06-07	DREL	Extended record residual length not including DRDL
	DRDL	Length of descriptor = addr(* - DRRL)
	DRDT	Text of data record

HOW TO LOCATE: Refer to Figure 6-2 in 6, "Diagnostic Aids"

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.)

The format of this block as it is printed in a dump is as follows.

Bytes Hex.	Label of Field	Description/Function
00-0F	MMSD	Storage descriptor (MMB)
10-17	MMWW	Work area
18-1B		Reserved
1C-1F	MMLK	Lockword
20-23	MMRE	Saved register 14
24-27	MMRF	Saved register 15
28-2B	MMR0	Saved register 0
2C-2F	MMR1	Saved register 1
30-33	MMR2	Saved register 2
34-37	MMR3	Saved register 3
38-3B	MMR4	Saved register 4
3C-3F	MMR5	Saved register 5
40-43	MMR6	Saved register 6
44-47	MMR7	Saved register 7
48-4B	MMR8	Saved register 8
4C-4F		Reserved
	MMCB	CCB
50-51	MMCT	Residual count
52-53	MMCM	Communication bytes
54-55	MMST	Status bytes
56-57	MMLU	LUB identifier
58	MMCA	Flags
59-5B		Channel program address
5C-5F		VSE internal use
	MMCH	Channel program
60-67	MMWT	Write CCW
68-6F	MMRD	Read CCW
70-F3	MMMA	Message output area
F4-13B	MMMI	Reply input area
13C-147		Reserved

HOW TO LOCATE: Refer to Figure 6-1 in 6, "Diagnostic Aids."

Module Control Block for FBA Devices

Bytes Hex.	Label of Field	Description/Function
00-0F	MFSD[1]	Storage descriptor MCB
10-17	MFSA	Module address (M000NNNN)
18	MFEA	Address of extension area
18	MFDT	Device type of file F = FBA device
19-1B		Extension area address
1C-1F	MFLK	Lockword
	MFCB	Command control block
20-21	MFCT	Residual count
22-23	MFCM	Communication bytes
24-26	MFST	Device status
26-27	MFLU	EXCP real plus LUB index (logical unit)
28-2B	MFCA	CCW address
2C-2F		CCW address in CSW
	MFXT	Extent information
30-33	MFLO	Low limit (starting physical block number of extent)
34-37	MFHI	High limit (entering physical block number of extent)
38-3B	MFSI	Default blocksize for FBA devices
3C-3F		Reserved
	MFLW	Locate control word
40	MFOB	Operation byte X'01' = write data X'06' = read data
41	MFRC	Replication count
42-43	MF#B	Number of blocks to be processed
44-47	MFRD	Relative displacement (begin of extent-block processor)
	MFCH	Channel program
48-4F	MFDF	Define extent CCW
50-57	MFLC	Locate CCW
58-5B	MFTI	TIC CCW
5C-5F	MFTV	Virtual address of buffer
60-67	MFRW	Read or write CCW
68-6B	MF$T	Owner of this I/O request
6C-6F	MF$1	Saved register 1 of current request
	MFED	Extent description block
70	MFMB	Mask byte C0 = Permit all write commands 04 = Permit all diagnostic commands 40 = Inhibit all write commands 44 = Inhibit all write commands and permit all diagnostic commands
71-73		Reserved, must be zero
74-77	MFBB	Physical address of first block of extent
78-7B	MFFB	Relative displacement of first block of extent
7C-7F	MFLB	Relative displacement of last block of extent
80-97		Reserved for alignment

NETWORK COMPOSER WORK AREA

Definition Macro: IPW$DWC

This work area is used by the composer to build records which it has received from the transmitter, into a transmission block.

Bytes Hex.	Label of Field	Description/Function
000-07F		PLS dynamic data area
080-083	NCWAHDR	Work area header
084-087	NCWTCB	Points to task control block
088-095	NCWAPUT	Parameter area for PUT macro
088	NCWACOCO	TCB comand code for current REC
089-08B	NCWARA	Address of record
08C	NCWAGP	Copied TCB gen. purpose byte
08D	NCWAG2	Copied TCB gen. purpose byte 2
	NCWAGJHR	X'80' - Job header record
	NCWAGJTR	X'40' - Job trailer record
	NCWAGDHR	X'20' - Dataset header record
08E-08F	NCWARL	Length of record
090-091	NCWARLM	Maximum record length
092	NCWART	Record type passed over
093	NCWAMLI	MLI request
094	NCWRCB	RCB of task
095	NCWAIND	Various indications
	NCWAFXD	X'80' - Fixed format record indication
	NCWANOCC	X'40' - Record without carriage control
096	NCWFLAGS	Composer flag bits
	NCWLSEG	X'80' - Last segment indicator
	NCWSYSIN	X'40' - Input record indicator
	NCWSYSOU	X'20' - Output record indicator
	NCWNOCMP	X'10' - Do not compress indicator
	NCWASPR	- Spanned record indicators (indicates first/middle/last segment or UNSP)
	NCWFDSG	X'02' - First data segment indicator
	NCWEXP	X'01' - Blank expansion indicator
097		Reserved
098-09B	NCWWPTR	Points to first free byte in segment area work field
09C-09F	NCWNCROF	Offset pointer into NCR
0A0-0A3	NCWNCREA	Points to last byte of NCR
0A4-0A7	NCWSGPTR	Points to segment area NCWSGAR (initialized by transmitter)
0A8-0AB	NCWSGC	Segment counter
0AC-0AD	NCWSGLEN	Length of segment area (initialized by transmitter)
0AE	NCWRLTB	Length of blank string to be added to current record
0AF		Reserved
0B0-0B1	NCW#BLNK	Nr. of blanks to be expanded
0B2-0B3	NCWTBLEN	Length of NJE transmission block

NETWORK COMPRESSION WORK AREA

Definition Macro: IPW$DKA

This work area is used by the compression routine when it is compressing records before building them into a transmission buffer. It is also used by the decompression routine when it decompresses buffers which it has received from the PNET driver.

Bytes Hex.	Label of Field	Description/Function
00-03	DKAINFA	Address of input field
04-07	DKAINFEA	Input field end address
08-0B	DKAOUFA	Address of output field
0C-0D	DKAOUFLN	Length of output field
0E-0F	DKAOULN	Output string length
10-13	DKAOUFEA	Output field end address + 1
14	DKAREQ	Request byte
	DKABCR	X'00' - Compression BSC mode
	DKABDCR	X'04' - Decompression BSC mode
	DKASCR	X'10' - Compression SNA mode
	DKASDCR	X'14' - Decompression SNA mode
	DKASNEEK	X'18' - Decompression SNA (sneek-a-peek)
15	DKASTAT	Status byte
	DKASTAT0	X'00' - No error occurred
	DKAERR01	X'01' - Output string exceeds output area
	DKAERR02	X'02' - SCB error, length exceeds input area
	DKAERR03	X'03' - Invalid request code
	DKAERR04	X'04' - Length of input string <= 0
	DKAERR05	X'05' - Length of output string <= 0
	DKAERR06	X'06' - Invalid SCB encountered
	DKAERR07	X'07' - SCB count = 0
	DKAERR09	X'09' - Output length exceeds output area
	DKAERR0A	X'0A' - Decompression output area to small
16-17		Reserved
18-1B	DKANSCB	Pointer to start SCB(SNA obly)
1C-1F	DKAINPOS	Points to next byte to be processed
20-21	DKAINFLN	Length of input field (string)
22-23	DKAREQL	SNA output buffer length

NETWORK DATA SET HEADER RECORD (DSH)

Definition Macro: IPW$DNR DHR=YES

The data set header record is a control record which is normally only present on output data sets (list or punch). It contains information relevant to the output, e.g. forms number, output class. A short form of the record, the record characteristics change section, may also be present on input jobs, if the record length of the data set is not 80 bytes.

The layout of the first four bytes of every header record is identical. Bytes 0 and 1 are the length of the entire block. Individual records must not be greater than 256 bytes long, so if the total record is longer it must be segmented. Byte 2 is a flag byte that is zero. Byte 3 is the transmission sequence indicator and is used to indicate that a header has been segmented. The high order bit (X'1... ....') indicates that there are more parts to come, and the other bits are a sequence counter of the blocks for this record. For example, if the header had to be split into three parts then the sequence indicators in the three blocks would be as follows:- X'80', X'81', and X'02'.

The layout of the first four bytes of all sections is always identical. The section flags are to be found in the description of the job header record (refer to "Network Job Header Record (JHR)" on page 5-86).

Bytes Hex.	Label of Field	Description/Function
00-01	NDHLEN	Length of entire block
02	NDHFLAGS	Flags
03	NDHSEQ	Transmission sequence indicator
	NDHLBCI	Length of block control information

- General section

00-01	NDHGLEN	Length of general section
02-03	NDHGFLGS	Section type flags
02	NDHGTYPE	X'00' - Identifier for general section
03	NDHGMOD	X'00' - Section modifier
04-0B	NDHGNODE	Destination node name
0C-13	NDHGRMT	Destination remote name
14-1B	NDHGPROC	Procedure invocation name
1C-23	NDHGSTEP	Step name
24-2B	NDHGDD	DD name
2C-2D	NDHGDSNO	Data set number
2E	NDHGSEC	Security level
32F	NDHGCLAS	Output class (default 'A')
30-33	NDHGNREC	Record count
34	NDHGFLG1	Flags
	NDHGF1SP	X'80' - Spin data set (segmented)
	NDHGF1HD	X'40' - Hold data set at final destination
	NDHGF1LG	X'20' - Job log indicator
	NDHGF1OV	X'10' - Page overflow indicator
	NDHGF1IN	X'08' - Punch interpret indicator
35	NDHGRCFM	Record format

19	NDHPSYID	System identifier
1A	NDHPNSEP	Number of job separator pages / cards
1B	NDHPOPTN	Option byte
	NDHPCSUP	X'20' - No separators between copies
1C-1D	NDHPPART	VSE/Advanced Functions partition identifier
1E-1F	NDHPCARD	Number of cards or pages
20-23		Reserved
24-27	NDHPCOMP	Compaction table name
28-2F	NDHPPASS	Data set password
30-73	NDHPSETP	SETPRT parameter list
	NDHPLLEN	Length of VSE/POWER section

- VSE/POWER subsystem section (short form). This form is used only on input, whenever data is read from a IBM 3540 diskette device.

05		Reserved
06-07	NDHPCUU	3540 cuu address
	NDHPCLEN	Length of short version

- Record Characteristics Change Section. This section is only used within an input data stream when the record length is not 80 bytes.

00-01	NDHCLEN	Length of Characteristics change section
02-03	NDHCFLGS	Section flags
02	NDHCTYPE	X'00' - Type for general section
03	NDHCMOD	X'40' - Modifier for characteristics change
04	NDHCFLG1	Flags
05	NDHCRCFM	Record format
06-07	NDHCLREC	Maximum record length
	NDHCLLEN	Length of section

40-47	NJHGORGN	Origin node name
48-4F	NJHGORGR	Origin remote name
50-57	NJHGXEQN	Execution node name
58-5F	NJHGXEQU	Execution userid (VM/370)
60-67	NJHGPRTN	Default print node name
68-6F	NJHGPRTR	Default print remote name
70-77	NJHGPUNN	Default punch node name
78-7F	NJHGPUNR	Default punch remote name
80-87	NJHGFORM	Job forms name
88-8B	NJHGICRD	Input card count
8C-8F	NJHGETIM	Estimated execution time
90-93	NJHGELIN	Estimated output lines
94-97	NJHGECRD	Estimated output cards
98-AB	NJHGPRGN	Programmers name
AC-B3	NJHGROOM	Programmers room number
B4-BB	NJHGDEPT	Programmers department number
BC-C3	NJHGBLDG	Programmers building number
C4-C7	NJHGNREC	Record count on output transmission
	NJHGLLEN	Length of general section

- VSE/POWER subsystem section

00-01	NJHPLEN	Length of VSE/POWER subsystem section
02-03	NJHPFLGS	Section type flags
02	NJHPTYPE	X'86' - Id for VSE/POWER section
03	NJHPMOD	X'00' - Modifier
04	NJHPFLG1	Flags
05	NJHPDISP	Job disposition (default D)
06		Reserved
07	NJHPSYID	Target system identifier
08-17	NJHPUSER	User information
	NJHPLLEN	Length of VSE/POWER section

- Section type flags

  These flags are used by other operating systems which could be present within the network. The layouts of the sections used by these systems can be found in the appropriate operating system manual. Because a subsystem appears in this list of section type flags does not mean that IBM supports this subsystem as part of the network.

NTYPGEN	X'00' - General section
NTYPSUB	X'80' - Subsystem section
NTYPASP	X'81' - ASP subsystem section
NTYPHASP	X'82' - HASP sybsystem section
NTYPJES1	X'83' - JES/RES subsystem section
NTYPJES2	X'84' - JES2 subsystem section
NTYPJES3	X'85' - JES3 subsystem section
NTYPPOWR	X'86' - VSE/POWER subsystem section
NTYPVNET	X'87' - RSCS subsystem section
NTYPUSER	X'C0' - User section

NETWORK PRESENTATION WORK AREA

Definition Macro: IPW$DWP

This work area is used by the receiver. Presentation services is responsible for taking a transmission block, de-compressing it and passing individual records to the receiver.

Bytes Hex.	Label of Field	Description/Function
000-07F	NPWDYNA	PLS dynamic data area
080-083	NPWDSD	Little storage descriptor
084	NPWRC	Return code presentation service
085-087		Reserved
088-08B	NPWBUPRO	Pointer to process buffer
08C-08F	NPWRCDPT	Address of physical record in normal or alternate pres. buffer
090-093	NPWHDRPT	Pointer to head accumulated RCD
094-097	NPWPRBUF	Pointer to presentation buffer
098-09B	NPWALBUF	Pointer to alternate pres. buffer
09C-09D	NPWTRCL	Accumulated length of segments
09E-09F	NPWFSGTL	Total length of spanned record as indicated in first segment
0A0-0A1	NPWPRBLN	Length of presentation buffer
0A2-0A3	NPWALBLN	Length of alternate pres. buffer
0A4-0A5	NPWALUSE	Use count alternate buffer
0A6	NPWLPREQ	Reserved
	NPWSPAN	X'80' - REQ accumulate record segments
	NPWNXTSG	X'40' - REQ accumulate header segments
0A7		Reserved
0A8-0DF	NPWFSVE	Function save area (implicit length definition)
0E0-103	NPWDKA	Decompression work area (impl. length def.)

154-157	NRUSRRCD	Pointer to user record
158-159	NRUSRLEN	Length user record
15A-15B	NRUSRBLN	Length intermediate data buffer
15C-15F	NRUSRBUF	Pointer to intermediate data buffer
160-163	NRRCDSAV	Pointer to saved current record
164-167	NRSPLRTN	Pointer to specific put data record routine (job input (SPLINP), job output (SPLOUT))
168	NRPRSRCB	SRCB char preceding record
169-16B		Reserved

• Function work areas

16C-1DB	NRACOUNT	Receiver account area
1DC-2DF	NRPWA	Presentation work area
2E0-423	NRCWA	Composer work area
424	NRWAEND	Start of presentation buffer

NODAL MESSAGE RECORD (NMR)

Definition Macro: IPW$DNR NMR=YES

The Nodal message record (NMR) is the record format used to transmit all messages and commands throughout the network.

Bytes Hex.	Label of Field	Description/Function
00	NMRFLAG	Flag byte
	NMRFLAGC	X'80' - NMRMSG contains a command
	NMRFLAGW	X'40' - NMROUT has VSE/POWER remote number
	NMRFLAGT	X'20' - NMROUT has a ICCF/CMS userid
	NMRFLAGU	X'10' - NMROUT has UCMID information
		* The next four flag settings are
		* not used by VSE/POWER.
	NMRFLAGR	X'08' - Console is only remote authorized
	NMRFLAGJ	X'04' - Console is not job authorized
	NMRFLAGD	X'02' - Console is not device authorized
	NMRFLAGS	X'01' - Console is not system authorized
01	NMRLEVEL	Importance level (high 4 bits)
	NMRPRIO	Output priority (low 4 bits)
02	NMRTYPE	Type byte
	NMRTYPEX	X'F0' - Reserved
	NMRTYPEF	X'02' - Formatted command in NMRMSG
	NMRTYPET	X'04' - Message text only in NMRMSG
	NMRTYPE4	X'08' - Message text contains control information
03	NMRML	Message length
04-0C	NMRTO	Target node
04-0B	NMRTONOD	Target node name
0C	NMRTOQUL	Target node qualifier
0D-14	NMROUT	Local output information
15-1D	NMRFM	Originator node
15-1C	NMRFMNOD	Originator node name
1D	NMRFMQUL	Originator node qualifier
1E-A2	NMRMSG	Message

029	NCBTTSTQ	Stop qualifier for NCBTTCS/NCBTTCE
	NCBPHALT	X'80' - PNET halts node due to error
	NCBVHALT	X'40' - VTAM requested node stop at EOJ (Z NET)
	NCBRHALT	X'20' - Request by remote to terminate at EOJ
02A-02B	NCBCNRV	Current number of active receivers
02C-02D	NCBCNTR	Current number of active transmitters
	NCBMNRV	7 - Maximum number of receivers
	NCBMNTR	7 - Maximum number of transmitters
02E-02F		Reserved

- The following table defines the task entries for the command/message transmitter/receiver.

030-037	NCBCONST	Console transmitter task
038-03F	NCBCONSR	Console receiver task

- The following table defines the task entries for the transmitters

040-077	NCBJTTB	Job transmitter table. This table has entries for upto seven transmitters, each entry being eight bytes long.
078-0AF	NCBOTTB	Output transmitter table. This table has entries for upto seven transmitters, each entry being eight bytes long.

- The following table defines the task entries for the receivers

0B0-0E7	NCBJRTB	Job receiver table
0E8-11F	NCBORTB	Output receiver table
120-123	NCBMSGA	Pointer to first message/command in queue
124-127	NCBMSGT	Tail pointer for message/command queue
128-12B	NCBNBFRQ	Pointer buffer queue signon-on event

- Buffer control fields

12C-12F	NCBIFRE	Address of free input buffer queue
130-133	NCBOTBS	Address of to-be-sent output buffer queue
134-137	NCBOBTL	Tail pointer of to-be-sent output buffer queue
138-13B	NCBLBFI	Line driver buffer (BSC)
13C-13F	NCBLBFO	Line driver buffer (BSC)
140-141	NCBBFSZ	Buffer size
142	NCBMNIB	Maximum number of input buffers
143	NCBMNJB	Maximum number of job/output transmission buffers
144-145	NCBNIBU	Number of acquired input buffers
146-147		Reserved

- I/O manager and SEND/RECEIVE manager fields

148-14B	NCBIBUF	Address of non-line input buffer (BSC)
14C-14F	NCBOBUF	Address of non-line output buffer (BSC)
150-153	NCBCBFI	Address of buffer for actual receive
154-157	NCBCBFO	Address of buffer for actual send
158-159	NCBRFCS	Remote held/released stream status
15A-15B	NCBTFCS	New held/released stream status
15C-15F	NCBPBFRQ	Head pointer to suspended buffer queue

170-173	NCBCTCB1	Address of the TCB for connect task initiated by the local operator
174-177	NCBCTCB2	Address of the TCB for connect task initiated by the remote operator
178-17B	NCBDTCB	Address of disconnect session task TCB
17C-17F	NCBDECB	ECB of disconnect task
180-183	NCBIFREX	Address of the 'receive-ahead' input buffer queue
184-187	NCBOTBSX	Address of the 'send-ahead' output buffer queue
188-18B	NCBOBTLX	Tail pointer to the 'send-ahead' output buffer queue
18C-193		Reserved
194-197	NCBWRKA	Address of the work area for compress/de-compress
198	NCBSGTE	Gate for send
199	NCBRGTE	Gate for receive
	NCBGTIPR	X'FF' - Gate is in progress

- Session status

19A	NCBSFL1	Primary application program
	NCBF11	X'80' - Primary in progress
	NCBF12	X'40' - Primary permit given
	NCBF13	X'20' - OPNDST in progress
	NCBF14	X'10' - OPNDST complete
	NCBF15	X'08' - Primary complete
	NCBF16	X'01' - Primary APPLID error
19B	NCBSFL2	Secondary application program
	NCBF21	X'80' - Secondary in progress
	NCBF22	X'40' - Secondary permit given
	NCBF23	X'20' - OPNSEC in progress
	NCBF24	X'10' - OPNSEC complete
	NCBF25	X'08' - Secondary complete
	NCBF26	X'04' - Session in progress
	NCBSEOK	X'02' - Session establishment complete
	NCBF28	X'01' - Secondary application program error
19C	NCBSFL3	Type of session
		X'00' - Primary half session
		X'FF' - Secondary half session
19D	NCBSFL4	Flag byte 4
	NCBRSHTS	X'80' - RSHUTD sent
	NCBRSHTR	X'40' - RSHUTD received
19E	NCBSEST	Session status byte
	NCBSSUB	X'80' - UNBIND received
	NCBSSTS	X'40' - TERMSESS is waiting
	NCBWSDT	X'20' - Wait for SDT
	NCBSSSD	X'10' - SDT received
	NCBSSCL	X'08' - Clear received
	NCBSHTC	X'04' - Wait for SHUTC

- Continuation of common part

1D4-1DB	NCBCPWD	Password for local node (outgoing)
1DC-1E3	NCBCLPW	Password for line (outgoing)

- Account record

NODE CONTROL BLOCK TASK ENTRY

Bytes Hex.	Label of Field	Description/Function
00	NCBERCB	RCB of task concerned
01	NCBEST1	Status/action byte
	NCBTDRN	X'80' - Task drained
	NCBTLVE	X'40' - Task live
	NCBDETE	X'20' - Dequeue & delete NCB task entry
	NCBTCRE	X'10' - Task creation requested
02	NCBESTS	Task stop state (duplicate to TCB field)
03	NCBETYP	Task type
	NCBETYPT	X'80' - Transmitter task
	NCBETYPR	X'40' - Receiver task
	NCBETYPC	X'20' - Console task
	NCBETYPJ	X'08' - Job processing
	NCBETYPO	X'04' - Output processing
04-07	NCBETCB	TCB address of task concerned

	WESI	Sequence ID (copy of PESI)
14C	WEMI	Multivolume identification (copy of PEMI)
14D	WESN	Volume sequence number (copy of PESN)
14E	WEOD	Number of opened diskettes (copy of PEOD)
14F	WEND	Number of diskettes to be read (copy of PEND)
150-157		Not used

- 3540 Volume 1 label layout in label test area (OELB)

	VOLL	Diskette volume 1 label
80-83	VLID	Volume label ID and number
84-89	VLSN	Volume serial number
8A	VLAI	Volume access indicator
8B-A4		Reserved
A5-B2	VLDI	Volume owner identity
B3-CA		Reserved
CB	VLPL	Physical record length
CC-CD	VLRS	Physical record sequence code
CE		Reserved
CF	VLST	Label standard version (W)

- 3540 Header 1 label layout in label test area (OELB)

	HDRL	Diskette header 1 label
80-83	HDID	Header label ID and number
84		Reserved
85-8C	HDFI	File identifier
8D-95		Reserved
96-9A	HDBL	Block length of data record
9B		Reserved
9C-A0	HDLO	Begin of extent (CCHRR)
A1		Reserved
A2-A6	HDHI	End of extent (CCHRR)
A7		Reserved
A8	HDBI	Bypass indicator (B)
A9	HDFS	File security indicator (S)
AA	HDWP	File write protection indicator (P)
AB	HDEI	Basic exchange indicator (,E)
AC	HDMV	Multivolume indicator (,C,L)
AD-AE	HDSN	Volume sequence number
AF-B4	HDCR	Creation date
B5-C1		Reserved
C2-C7	HDEX	Expiration date
C8	HDVI	Verify indicator (,V)
C9		Reserved
CA-CE	HDED	End of data address (CCHRR)
CF		Reserved

PARTITION CONTROL BLOCK (PDB)

Definition Macro: IPW$DPD

A partition control block is dynamically created for each partition to be controlled by VSE/POWER. 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$DDE macro instruction.

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:

1R86I PLEASE SPECIFY SPOOL DEVICES

after entering the PSTART command to start a partition.

Bytes Hex.	Label of Field	Description/Function
00-0F	PSSD	Storage descriptor (PDB)
10-11		Reserved
12-13	PDPI	Partition identifier
14-17	PDNE	Number of entries
18-1B	PDCM	Partition comreg address
1C-1F	PDPB	PIB address
20-23	PDPA	First entry address
24-27	PDBA	Virtual begin address of related partition
28-2B	PDEA	Virtual end address of related partition
2C-2F	PDRL	Real begin address of related partition
30-33	PDRH	Real end address of related partition
34-35	PDJN	LST/PUN job number indicators
36	PDTT	Termination code
37		Not used
38-3B	PDJH	Pointer to job header record
3C-3F	PDJT	Pointer to job trailer record

- Statistical information. This information is destined for the execution account record and there is a pointer to the SLI work area.

40-43	PDSL	Pointer to SLI work area
44-47	PD#L	Number of lines spooled
48-4B	PD#C	Number of cards spooled
4C-4D	PD#P	Number of pages spooled
4E	PDOC	Default output class
4F	PDMT	Multitasking indicator

- 3540 spool device entry (same format as RDR device entry)

50-5F	PDER	3540 spool entry

- RDR device entry (maximum = 1)

60-63	PDPU	Address of entry in the VSE/Advanced Functions

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 areas that contain the input or output data records.

Note: For an RJE task the CCB and the channel program is in the LCB. 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 5-26 to Figure 5-36 illustrate this relationship for an RJE task and for a local task.

HOW TO LOCATE: Refer to Figure 6-2 in 6, "Diagnostic Aids."

3540 PHYSICAL WORK SPACE

Definition Macro: IPW$DPW E3540=YES

The 3540 physical work space is used to address and save the information necessary for diskette processing. The workspace is either reserved by the physical routine (IPW$$PR) in case of alternate diskette processing, by the process diskette record routine (IPW$$ER) in case of primary diskette processing or by the logical reader routine (IPW$$LR) for dynamic * $$ RDR processing. The address of the 3540 physical workspace currently in use is stored in the TCB field 'TC3W'.

Bytes Hex.	Label of Field	Description/Function
00-03	PERA	Real address of the physical work space
04-07	PEDI	Device type indication.
04		Reserved.
05	PEDT	Device type.
06-07	PELU	Programmer logical unit.
08-0B	PECU	device address of diskette unit (' cuu')
0C-0F	PEHA	address of higher level 3540 PWS
10-1F	PEDP	Diskette parameters from PSTART.
10-17	PEFI	File identification.
18-1B	PEPS	PSTART parameters.
18	PEOP	Option byte feed for 3540.
	PEFD	X'01' - Feed 3540
19	PEND	Number of diskettes to be read.
1A	PESC	Sequence check required.
1B	PEVE	Verify requested.
1C-1F	PECD	Displacement between real and virtual CCB addresses.
20-23	PECV	Address of 3540 CCB or physical data area
24-27	PEDV	Virtual address of first 3540 data buffer
28-2B	PEDA	Real address of first 3540 data buffer
2C-2F	PEVN	Virtual address of second data buffer.
30-33	PERN	Real address of second data buffer.
34-37	PEBS	Real address of forced pre-SEEK CCW.
38-3B	PESK	Seek address (00CCHHRR).
3C-3F	PESO	Overlap seek address (00CCHHRR).
40-43	PELO	Extent lower limit (00CCHHRR).
44-47	PEED	Next sector address (00CCHHRR).
48-49	PERL	Record length.
4A-4B	PENN	No. of buffers allocated in 2nd data buffer.
4C-4D	PESI	Sequence identification.
4C	PEMI	Multivolume identification.
4D	PESN	Volume sequence number.
4E	PEOC	Open return code.
4F	PEOD	Number of opened diskettes.
50-57	PEDW	Double word for conversion purposes.
58	PESD	Copy of device status byte from CCB

HOW TO LOCATE: Refer to Figure 6-2 in 6, "Diagnostic Aids."

- Own node information

84-8B	PNCBONN	Node name of own system
8C	PNCBONQ	Qualifier for own system
8D-94	PNCBNDTN	Network definition table phase name
95	PNCBAPPL	Length of APPLID
96-9D	PNCBAPID	APPLID of own node
9E-B7		Reserved

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 as follows.

Refer to the Disk Management Block (DMB) auxiliary queue record area for a fuller description of the individual entries.

Bytes Hex.	Label of Field	Description/Function

- 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	QRDY	Date
08-0B	QRST	Operation start time
0C-0F	QRET	Operation end time
10-1F	QRUI	User information
20-27	QRNM	Job name
28-29	QRNO	Job number
2A	QRQI	Queue record identifier
2B	QRCN	VSE/POWER cancel code
2C	QRRJ	Line identifier or device type
2D-2F	QRCU	Channel and unit (line address)
30	QRFJ	From terminal identifier
31	QRTJ	To terminal identifier
32	QRCL	Class
33	QRPY	Priority
34-37	QRNR	Record count
38-39	QRNT	Number of tracks
3A	QRSN	Job suffix number
3B	QRNC	Number of copies
3C-3F	QRFI	Forms identifier
40-43	QRNA	Number of additional records
44-45	QRNP	Number of pages
46-47	QRNE	Number of extra pages
48-4B	QRLC	Line/card counter
4C-4F	QRRR	Restart page count
50	QRCR	Copies remaining
51	QRDI	Default disposition/purge flag 'P'
52	QRDP	Disposition
53	QRSP	Number of separators
54-57	QRBS	Number of records before split
58-5B	QRBM	Maximum value of count
5C-5F	QRBN	Additional count value
60-61	QRER	Physical 3540 device address (packed)
62-63	QRJ#	Save job number for accounting
64-67	QRCP	Compaction table name

This determines the meaning of the fields QRNS, QRQP, and QRQN as shown in Figure 5-18 and Figure 5-19.

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

Figure 5-18. Queue Record is NOT First in Queue Set (QRFS = X'00')

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

Figure 5-19. Queue Record IS First in Queue Set (QRFS = X'01')

HOW TO LOCATE: Refer to Figure 6-2 in 6, "Diagnostic Aids."

46	LCBLREQ	Last sent request
47	LCBREQF	Request code for I/O monitor
48-4B	LCBNEXT	LCB chain pointer, last = zero
4C-4F	LCBRCPT	Pointer to next record in buffer
50-51	LCBPUBA	Pointer to related PUB entry
52-53	LCBLRCL	Length of logical record
54-55	LCBTIMC	Time out count. This field contains 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.
56		Not used
57	LCBTOCT	Count for:- 5 retries of Enable, Nop, Read response CCW sequence if line is in control state before trying to send ENQ. or:- 5 retries of Enable, Nop, Read response CCW sequence if line is in switched mode, not signed on, control state and timed out.
58	LCBRTRY	Retry count for unit check (max 30). If line is in data transfer mode and received unit check, it is reset at SYSREC writing.
59	LCBMSGI	Message index in virtual message queue
5A	LCBLIMO	Line mode
	LCBMREC	X'80' - Receive mode
	LCBMXMT	X'40' - Transmit mode
	LCBDOUT	X'20' - Discontinued output mode
	LCBTRDR	X'08' - Reader task indicator
	LCBTOUT	X'07' - Output task indicator
	LCBTLST	X'04' - List task indicator
	LCBTPUN	X'02' - Punch task indicator
	LCBTMSG	X'01' - Message task indicator
5B	LCBOUSW	LST/PUN output indicator
	LCBOUL1	X'80' - List output queued for LST1
	LCBOUP1	X'08' - Punch output queued for PUN1
5C	LCBACT	Activity control byte
	LCBACTR	X'80' - Task creation
	LCBASHD	X'40' - Shutdown
	LCBATSTP	X'20' - Task stop
	LCBASGF	X'10' - Final signoff
	LCBALSTP	X'08' - Line stop
	LCBALSTR	X'04' - Line start
	LCBALIN	X'02' - Line initialization
	LCBAKILL	X'01' - Line stop (PSTOP cuu,KILL)
5D	LCBTRACE	Line trace indication X'FF' - trace recording requested
5E-5F	LCBNORC	Number of records in the buffer
60-61	LCBMSG#	Number of messages queued
62	LCBSTMSG	3741 status message
63		Not used
64-7B	LCBTQE	Timer queue element

	F5TNL	X'80' - New line character on end of HT
	F5EJECT	X'20' - Eject before messages
	F5SPACE	X'10' - Terminal requests space 3 after record.
	F5T3741	X'08' - Translate table with 3741 control
	F5L2780	X'04' - Terminal like 2780
9A	LCBFLG1	LCB flag byte 1
	LF1CHEND	X'80' - LCBSCAN called from channel end
	LF1TERM	X'40' - Terminate session
	LF1RVIS	X'20' - RVI sent
	LF1SIGN	X'10' - Remote is signed on
	LF1SOMQ	X'04' - Signoff message queued
	LF1TOMSG	X'02' - Time-out message queued
	LF1EOTR	X'01' - EOT has been received for writer
9B	LCBFLG2	LCB flag byte 2
	LF2TIME	X'80' - Timer is set
	LF2MSOK	X'20' - Sending of messages disabled
	LF2SENQ	X'10' - ENQ to be sent
	LF2BUSY	X'08' - Line busy indication
	LF2TOMSG	X'02' - Time-out message to be sent
	LF2SFRC	X'01' - Forms change in progress
9C	LCBFLG3	LCB flag byte 3
	LF3LSCN	X'80' - Trace entry for activity
	LF3SOFF	X'40' - Signoff read by BSC reader
	LF3MR17	X'20' - Message 1R17I already queued
	LF3EOTR	X'10' - EOT has been received for reader
	LF3EOTS	X'08' - EOT to send in order to reset terminal
	LF3TCRF	X'04' - Task creation failed
	LF3EODR	X'02' - Write end-of-day record
	LF3TEOJ	X'01' - Turnaround indication
9D	LCBRCNT	Retry count

- BSC characters

9E	MBSCSOH	SOH BSC character
9F	MBSCENQ	ENQ BSC character
A0	MDLESTX	Start of text sequence
A1	MBSCSTX	STX BSC character
A2	METBSEQ	End of block sequence
A3	MBSCETB	ETB BSC character
A4	METXSEQ	End of text sequence
A5	MBSCETX	ETX BSC character
A6	MACK0SEQ	Even acknowledgement sequence
A7	MBSCACK0	Even acknowledgement character
A8	MACK1SEQ	Odd acknowledgement sequence
A9	MBSCACK1	Odd acknowledgement character
AA	MNAKSEQ	Negative acknowledgement sequence
AB	MBSCNAK	Negative acknowledgement character
AC	MBSCACKX	Acknowledgement conversion character
AD	MBSCCWCH	CCW chaining character
AE	MWACKSEQ	WACK sequence
AF	MBSCWACK	WACK BSC character
B0	MEOTSEQ	DLE-EOT sequence
B1	MBSCEOT	EOT BSC character
B2-B3	MDLEETB	DLE-ETB sequence

128-12B	DCT1LID	List task identifier (1LST)
12C-12F	DCT1LTCB	TCB address of task
130	DCT1LST1	Status byte 1 (see Input DCT)
131	DCT1LST2	Status byte 2 (see Input DCT)
132	DCT1LST3	Status byte 3 (see Input DCT)
133	DCT1LIDI	Hex identifier (as LCBOUSW)
134-137	DCT1LCLS	List classes
138-139	DCT1LPLN	Length of one print line
13A	DCT1LNRC	Number of records / buffer
13B	DCT1LIDH	Hex identifier (inv. LCBOUSW)
13C	DCT1LSCT	Space 3 line count
13D	DCT1LCSC	Component select character
13E-141	DCT1LFRM	Forms identifier
142	DCT1LPRC	Previous command code
143		Reserved

- Punch DCT

144-147	DCT1PID	PUNCH task identifier (1PUN)
148-14B	DCT1PTCB	TCB address of task
14C	DCT1PST1	Status byte 1 (see Input DCT)
14D	DCT1PST2	Status byte 2 (see Input DCT)
14E	DCT1PST3	Status byte 3 (see Input DCT)
14F	DCT1PIDI	Hex identifier (as LCBOUSW)
150-153	DCT1PCLS	Punch classes
154-155	DCT1PPLN	Length of one punch line
156	DCT1PNRC	Number of records / buffer
157	DCT1PIDH	Hex identifier (inv. LCBOUSW)
158	DCT1PSCT	Space 3 line count
159	DCT1PCSC	Component select character
15A-15D	DCT1PFRM	Forms identifier
15E	DCT1PPRC	Previous command code
15F		Reserved

- Message task DCT

160-163	DCT1MID	MSG task identifier (1MSG)
164-167	DCT1MTCB	TCB address of task
168	DCT1MST1	Status byte 1 (see Input DCT)
169	DCT1MST2	Status byte 2 (see Input DCT)
16A	DCT1MST3	Status byte 3 (see Input DCT)
16B		Reserved
16C-16F	DCT1MCLS	N/A
170-171	DCT1MPLN	Length of one print line
172	DCT1MNRC	Number of records / buffer
173		N/A
174	DCT1MSCT	Space 3 line count
175	DCT1MCSC	Component select character
176-179		Not used
17A	DCT1MPRC	Previous command code
17B		Reserved

- Reader DCT

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 Hex.	Label of Field	Description/Function
00-03	SRBNEXT	Address of next SRB in chain
04	SRBREQ	Request type
	SRBSPR	C'S' - SETPRT request (SRBSPR)
	SRBLD	C'L' - Load request
	SRBFT	C'F' - Fetch and invoke B transient
	SRBFCB	C'B' - Load FCB request (SRBFT)
	SRBIDR	C'I' - Trace logging request
	SRBFND	C'C' - Find member request(LIB)
	SRBNTE	C'N' - Note member request(LIB)
	SRBPNT	C'P' - Point request(LIB)
	SRBGRC	C'R' - Get record request(LIB)
	SRBDIS	C'D' - Disconnect member request(LIB)
05	SRBTRC	Return code (low-order byte of R15)
	SRBPNF	X'04' - Phase not found
	SRBPTL	X'08' - Phase too large
06-07		Reserved
08-0B	SRBECB	Event control block (VSE/POWER task, issuing service request, waits on this ECB, which is posted by the service subtask).
0C-17	SRBPARM	Request parameter list: This field is broken down differently for each service request type:

- SETPRT request

0C-0F		Address of SETPRT parameter list
10-17		Reserved for future use

- Load request

0C-0F	SRBAPN	Address of the phase name
10-13	SRBLDA	Address where to load the phase
14-17	SRBPSZ	Size of phase in bytes

- Trace logging services

0C-0F	SRBTSA	Trace start address
10-13	SRBTEA	Trace end address
14-17		Reserved for future use

SNA SESSION CONTROL BLOCK FOR PNET (SSCB)

Definition Macro: IPW$DSS

A SNA session control block is created is a SNA session is established to another node in the network. The SSCB is anchored to the appropriate node control block (NCB).

Bytes Hex.	Label of Field	Description/Function
00-0F	SSCBSD	Storage descriptor
• Function save areas		
10-3F	SSCBFUSS	Function save area for SEND exit
40-6F	SSCBFUSR	Function save area for RECEIVE exit
• Save areas for VTAM macro calls		
70-73	SSCBLS13	Reg 13 save area on SEND, PNET Driver
74-BB	SSCBVSS1	Save area used for SEND macro
BC-BF	SSCBES13	Reg 13 save area on SEND, SEND EXIT
C0-107	SSCBVSS2	Save area used for CHECK/SEND, SEND EXIT
108-10B	SSCBLR13	Caller's reg 13 save area if RECEIVE
10C-153	SSCBVSR1	Save area used for RECEIVE macro
154-157	SSCBER13	Caller's reg 13 save area, RECEIVE EXIT
158-19F	SSCBVSR2	Save area used for RECEIVE macro, RECEIVE exit
1A0-1E7	SSCBVDSA	Save area used by disconnect task
1E8-1EB	SSCBR14S	VTAM return address, SEND exit
1EC-1EF	SSCBR14R	VTAM return address, RECEIVE exit
• Save area for individual addresses/pointers		
1F0-1F3	SSCBSRQE	Connect SRQE address
1F4-1F7	SSCBNCB	Connect Node control block address
1F8-1FB	SSCBS2P	Connect save area PAP
1FC-1FF	SSCBS2S	Connect save area SAP
200-203	SSCBS3	Disconnect save area
204-267	SSCBSRPL	RPL skeleton
268-2A7	SSCBNIB	NIB skeleton
• BIND image and FM-Headers		
2A8-2CB	SSCBBIND	Bind RU
2CC-2D3	SSCBFMHO	FM header (output)
2D4-2DB	SSCBFMHI	FM header (input)

- Restart information

5C-5F	LURS	Restart
5C	LURX	Restart function index
5D-5F	LURP	Restart page count

- List and punch characteristics

60-63	LUPH	Pointer to device in SUCB
64	LULO	List output support X'80' - ASCII X'40' - Compression X'20' - Transparency X'10' - Spanning X'08' - Inter-record separator X'04' - Xlation of characters below blank X'01' - Compaction
65	LUPO	Punch output support X'80' - ASCII X'40' - Compression X'20' - Transparency X'10' - Spanning X'08' - Inter-record separator X'01' - Compaction
66	LUPD	PDIR information byte X'80' - PDIR outbound allowed
67	LUAD	Card/document flow X'80' - Card inbound allowed X'40' - Card outbound allowed X'20' - Basic exchange media allowed X'08' - Document inbound allowed X'04' - Document outbound allowed
	LUOC	Current compaction table in use by outbound processor
68-6B	LU01	Compaction table name
6C-6F	LU02	Pointer to COCB entry

- Process control section. Referred by label LUTC.

70-73	LURT	RDR, LGN, LGF TCB address
74-77	LULT	LST, PUN TCB address
78-7B	LUMT	Message TCB address
7C-7F	LUTI	RDR2 TCB address
80-83	LUTH	LGH TCB address
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

SNA LOGON REQUEST CONTROL BLOCK (LRCB)

Definition Macro: IPW$DLR

A LOGON request control block contains information for 6 LOGON requests to the VSE/POWER 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.

The format of a LOGON request control block is as follows.

Bytes Hex.	Label of Field	Description/Function
00-0F	LRSD	Storage descriptor (LRCB)
10-13	LRNX	Pointer to next LRCB
14-17		Reserved
18	LRLC	Length of one LRCB
19	LRLB	Length of one LRUB
1A	LRAL	No. of total LRUBs in LRCB
1B	LRUS	No. of active LRUBs in LRCB
1C-1F		Reserved
20-7F	LRAU	Space for six LRUBs
20-2F		First LRUB entry
20-23	LRAC	ACB address
24-2B	LRLU	LU-name
2C	LRST	Status (X'01' indicates active entry)
2D-2F	LRLM	Length of LOGON message

SNA SESSION REQUEST QUEUE (SRQE)

Definition Macro: IPW$DRQ

Bytes Hex.	Label of Field	Description/Function
000-00F	SRQESD	Section descriptor
010-011	SRQETLGF	SRQE length (multiple 128)
012-013		Reserved
014-017	SRQENPTR	Next in chain pointer
018-01B	SRQETPTR	Task pointer belonging to
01C-01F	SRQEANCB	Address of node control block
020-021	SRQEALEN	Actual length of BIND-RU
• Status:		
022	SRQESTA	SRQE status byte
023	SRQERC	Reason code used in NSEXIT
024-025	SRQESSMO	Sense modifier
• This part will contain the BIND-RU		
026-049	SRQEBDRU	Reserved for BIND-RU
026-03F	SRQEBIND	BIND-RU area
040	SRQEPLUL	PLU-name length
041-048	SRQEPLU	PLU-name
049		
• This part contains the ACF/VTAM emergency save area		
04A-04B		Unused
04C-093	SRQESAVE	Reserved for VTAM save area
094-097	SRQESR13	Save reg. 13, if VTAM-macro
098-09B		Reserved
• This part contains the RPL after a BIND-RU has been received		
09C-0FF	SRQERPL	RPL area
• This part contains the NIB after a BIND-RU has been received		
100-13F	SRQENIB	NIB area

59-5B	SUP1L	Pointer to LUCB
5C-5F	SUP1F	Forms ID
60-63	SUP1C	Punch output classes
64-67	SUR1P	Reader - C'RDR1'
68	SUR1S	Device status
69-6B	SUR1L	Pointer to LUCB
6C-6F	SUR1F	Forms ID (not used by reader)
70-73	SUR1C	Reader class - C'A' (initialized by IPW$$IB)
74-77	SUX1P	Exchange media reader - 'RDR2'
78	SUX1S	Device status
79-7B	SUX1L	Pointer to LUCB
7C-7F	SUX1F	Forms ID (not used by reader)
80-83	SUX1C	Reader class - C'A' (initialized by IPW$$IB)
84-87	SUC1P	Console - C'CON1'
88	SUC1S	Device status
89-8B	SUC1L	Pointer to LUCB
8C-8F	SUC1C	Forms ID (not used by console)
90-93	SUC1C	Console class - C'A' (initialized by IPW$$IB)
94	SUHD	Device List delimiter
95-96		Reserved
97	SUDLS	Device select indicator

- Compaction table information for outbound (referred to bu SOOC)

98-9B	SUO1	Name of default table
9C-9F	SUO2	Address of default table virtual
A0	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

A1	SUMR	Message request status X'80' - Message processor for work station is active X'40' - Request to interrupt IPW$$OB for outbound message was issued
A2-A4	SUMRL	Pointer to the LUCB with the suspending IPW$$OB
A5		Unused
A6-A7	SUMN	No. of messages
A8	SUMC	Subchain index
A9	SUMD	Temporary delete chain index
AA	SUTY	Terminal type
AB	SUTF	Terminal features X'80' - Console specified

- Miscellaneous

SOURCE LIBRARY WORK AREA

Definition Macro: IPW$DSL SLWA=YES

This work space is reserved and used by phase IPW$$SL and provides storage to read records from a source statement library. The work space is anchored to the partition control block of the partition concerned.

Bytes Hex.	Label of Field	Description/Function
000-00F	SLWASD	Storage descriptor
		• Communication switches
010	SLRS	Read SSL switch C'R' - Read request
011	SLRR	Read RDR switch C'R' - Read request C'I' - Ignore RDR record
012	SLF1	Flag byte 1
	SLEOM	X'80' - End of member
013		Reserved for future use
		• Buffer control information
014-017	SLCREC	Current record address
018-01B	SLLREC	Address of last REC in buffer
	SLRLEN	80 - SSL record length
01C-33B	SLRBUF	Buffer area
		• Pointers and save areas
33C-33F	SLAPDB	Address of partition control block
340-343	SLASRB	Address of service request block
344-347	SLSLME	Address of current SL-member element
348-34B	SLIBUF	Address of ICCF process buffer
34C-383	SLSAVA	Save area used by ASYN SERV
384-387	SLDALN	Data name length save area
		• RPL and buffer used by librarian
388-407	SLRPL	Request parameter list area
408-	SLBUF	Process buffer area

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 storage control block is as follows.

Bytes Hex.	Label of Field	Description/Function
000-00F	SCSD	Storage descriptor
010-013	SCLP	Last permanent page
014-017	SCFP	First fixed page
018-01B	SCEB	Event control block
01C-01F	SCLK	Lockword
020-023	SCRE	Task register 14
024-027	SCRF	Task register 15
028-02B	SCR0	Task register 0
02C-02F	SCR1	Task register 1
030-033	SCR2	Task register 2
034-037	SCR3	Task register 3
038-03B	SCR4	Task register 4
03C-03F	SCR5	Task register 5
040-13F	SCBT	Storage assignment table
140-147	SCFB	Constant to initialize the first BCW in a new fixed page in the fixable area.
148-14F	SCLB	Constant to initialize the last BCW in a new fixed page in the fixable area.
150-15B	SCPF	Page fix/free work area
150-153		Page virtual address
154-157		Page length (-1)
158-15B		End-of-list indicator (X'FF000000')
15C-15F		Reserved

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 VSE/POWER 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 VSE/POWER start-up routines to reflect the amount of real storage available to the VSE/POWER partition at that time.

TAPE CONTROL BLOCK (TBB)

Definition Macro: IPW$DTB

This control block dynamically created to satisfy requirements of VSE/POWER tasks utilizing tape as intermediate storage. Its format as it is printed in a dump is as follows.

The IPW$DTB macro is issued by VSE/POWER phases IPW$$OF, IPW$$OT and IPW$$SY.

Bytes Hex.	Label of Field	Description/Function
	TBDS	Define dummy section
00-03	TBSD	Storage descriptor 'TBB'
04-07	TBPB	Physical unit block address
08	TBFU	Function control byte
	FOUT	X'01' Output processing
	FINP	X'02' Input processing
	FCON	X'10' Continuation requested
	FCLT	X'20' Close tape request
	FOPT	X'40' Open tape request
	FBTB	X'80' Build TBB request
09	TBFG	TBB flag byte:
	TTWA	X'01' temporary work area available
	TMFI	X'02' multi-file volume
	TMVF	X'04' multi-volume file
	TUNL	X'08' unlabeled tape
	TBLK	X'10' blocked data
	TDMD	X'20' data mode, no label
	TEOV	X'40' EOV indicated
	TEOF	X'80' EOF indicated
0A	TBDM	Default mode setting, loaded from PUB
0B	TBSM	Specified mode setting, given in operator command
0C	TBDT	Device identifier 'T' for 'tape'
0D-0F	TBCU	Physical unit number 'cuu', which is given in the operator command
10-1F	TBCB	Command control block
10-11	TBRS	Residual count
12-13	TBCM	Communication bytes
14	TBCS	Channel and device status
15	TBC1	Channel and device status
16-17	TBLU	EXCP real plus LUB index
18-1B	TBCA	CCW address if it is a SYSIN tape RDR task then this field points to a start of I/O area (TBAV)
1C-1F	None	CCW address in CSW
20-27	TBCW	Channed command word
20	TBCC	Write command code X'01'
21-23	TBRA	Address of data area
24-25	TBWS	CCW flag bytes
26-27	TBCT	Count field, initialized by '1'
	TLNC	CCW length = addr (TBCT-TBCW)
28-38	TBLA	File label save area in form 'Label.....'

SOURCE LIBRARY WORK AREA

Definition Macro: IPW$DSL SLWA=YES

This work space is reserved and used by phase IPW$$SL and provides storage to read records from a source statement library. The work space is anchored to the partition control block of the partition concerned.

Bytes Hex.	Label of Field	Description/Function
000-00F	SLWASD	Storage descriptor
		• Communication switches
010	SLRS	Read SSL switch C'R' - Read request
011	SLRR	Read RDR switch C'R' - Read request C'I' - Ignore RDR record
012	SLF1	Flag byte 1
	SLEOM	X'80' - End of member
013		Reserved for future use
		• Buffer control information
014-017	SLCREC	Current record address
018-01B	SLLREC	Address of last REC in buffer
	SLRLEN	80 - SSL record length
01C-33B	SLRBUF	Buffer area
		• Pointers and save areas
33C-33F	SLAPDB	Address of partition control block
340-343	SLASRB	Address of service request block
344-347	SLSLME	Address of current SL-member element
348-34B	SLIBUF	Address of ICCF process buffer
34C-383	SLSAVA	Save area used by ASYN SERV
384-387	SLDALN	Data name length save area
		• RPL and buffer used by librarian
388-407	SLRPL	Request parameter list area
408-	SLBUF	Process buffer area

	SPNR	X'00' - Normal return
	SPLR	X'08' - End-of-data on GETSPOOL
	SPFR	X'09' - Warning: Task had to wait for Queue/ Account File space.
1F	SPER2	Error-feedback byte 2
	SPAI	X'80' - Access inhibited (wrong password)
	SPME	X'01' - Multiple Queue entries found
20	SPR1	PUTSPOOL request type
	SPEJ	X'40' - The last data record for internal reader job is contained in this PUTSPOOL request
21	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
22	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
23	SPBG	X'10' - Buffered GETSPOOL
	SPR4	CTLSPOOL request-byte 2
	SPOO	X'80' - Spool queue display
	SPQR	X'20' - Queue lookup request
24-2B	SPXR	PUTSPOOL user's XECB name
2C-33	SPXL	GETSPOOL/CTLSPOOL user's XECB name
34-37	SPCB	Address current PUTSPOOL buffer area
38-3B	SPPB	Address user-supplied buffer area for VSE/POWER
	SPMO	X'1C' - Message displacement in buffer from byte 0
3C-3F	SPBL	Data buffer area length
40-43	SPRL	Data record length
	SPRS	Browse control
44	SPSN	Signed browse start control
45-47	SPCT	Browse start line number
48	SPCL	LST output class
49	SPDP	LST output disposition
4A	SPCC	Print/POWER control character
4B	SPSQ	Display job status return C'N' - Not on VSE/POWER queues C'R' - On RDR queue C'L' - On LST queue C'P' - On PUN queue C'X' - On XMT queue
4C	SPQD	Job disposition on RDR/LST queue

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 storage control block is as follows.

Bytes Hex.	Label of Field	Description/Function
000-00F	SCSD	Storage descriptor
010-013	SCLP	Last permanent page
014-017	SCFP	First fixed page
018-01B	SCEB	Event control block
01C-01F	SCLK	Lockword
020-023	SCRE	Task register 14
024-027	SCRF	Task register 15
028-02B	SCR0	Task register 0
02C-02F	SCR1	Task register 1
030-033	SCR2	Task register 2
034-037	SCR3	Task register 3
038-03B	SCR4	Task register 4
03C-03F	SCR5	Task register 5
040-13F	SCBT	Storage assignment table
140-147	SCFB	Constant to initialize the first BCW in a new fixed page in the fixable area.
148-14F	SCLB	Constant to initialize the last BCW in a new fixed page in the fixable area.
150-15B	SCPF	Page fix/free work area
150-153		Page virtual address
154-157		Page length (-1)
158-15B		End-of-list indicator (X'FF000000')
15C-15F		Reserved

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 VSE/POWER 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 VSE/POWER start-up routines to reflect the amount of real storage available to the VSE/POWER partition at that time.

TAPE CONTROL BLOCK (TBB)

Definition Macro: IPW$DTB

This control block dynamically created to satisfy requirements of VSE/POWER tasks utilizing tape as intermediate storage. Its format as it is printed in a dump is as follows.

The IPW$DTB macro is issued by VSE/POWER phases IPW$$OF, IPW$$OT and IPW$$SY.

Bytes Hex.	Label of Field	Description/Function
	TBDS	Define dummy section
00-03	TBSD	Storage descriptor 'TBB'
04-07	TBPB	Physical unit block address
08	TBFU	Function control byte
	FOUT	X'01' Output processing
	FINP	X'02' Input processing
	FCON	X'10' Continuation requested
	FCLT	X'20' Close tape request
	FOPT	X'40' Open tape request
	FBTB	X'80' Build TBB request
09	TBFG	TBB flag byte:
	TTWA	X'01' temporary work area available
	TMFI	X'02' multi-file volume
	TMVF	X'04' multi-volume file
	TUNL	X'08' unlabeled tape
	TBLK	X'10' blocked data
	TDMD	X'20' data mode, no label
	TEOV	X'40' EOV indicated
	TEOF	X'80' EOF indicated
0A	TBDM	Default mode setting, loaded from PUB
0B	TBSM	Specified mode setting, given in operator command
0C	TBDT	Device identifier 'T' for 'tape'
0D-0F	TBCU	Physical unit number 'cuu', which is given in the operator command
10-1F	TBCB	Command control block
10-11	TBRS	Residual count
12-13	TBCM	Communication bytes
14	TBCS	Channel and device status
15	TBC1	Channel and device status
16-17	TBLU	EXCP real plus LUB index
18-1B	TBCA	CCW address if it is a SYSIN tape RDR task then this field points to a start of I/O area (TBAV)
1C-1F	None	CCW address in CSW
20-27	TBCW	Channed command word
20	TBCC	Write command code X'01'
21-23	TBRA	Address of data area
24-25	TBWS	CCW flag bytes
26-27	TBCT	Count field, initialized by '1'
	TLNC	CCW length = addr (TBCT-TBCW)
28-38	TBLA	File label save area in form 'Label.....'

TASK CONTROL BLOCK (TCB)

Definition Macro: IPW$DTC

Each VSE/POWER 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 workarea.

When the TCB belongs to a command processor task, the general task work area is replaced by command processor control fields. When it belongs to a BSC line manager task, these fields are replaced by BSC line manager control fields. Refer to the relevant paragraphs in this section, "Command Processor Control Block" and "BSC Line Manager TCB Fields".

TCB Task Management Fields

Bytes Hex.	Label of Field	Description/Function
00-0F	TCSD	Storage descriptor
00-03	TCBI	Storage descriptor block ID C'TCB '
04-07	TCTI	Task ID C' TSK' C'O CP' - command processor task C'I IT' - initiator task C'T TT' - terminator task C'T TI' - timer task C'RRDR' - local reader task C'WLST' - local printer task C'WPUN' - local punch task C'E xX' - Execution processor task.xx specifies the partition requesting the task. C'1' - C'5 ' 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.) C'LRLM' - line manager task C'P PS' - print status task C' ACT' - account task C'J ' - spool manager task. The three remaining bytes indicating the type of task. (RDR, LST, or SPM.) C'LSNA' - SNA task C'NTFY' - notify task C'LLDR' - line driver C'NRVn' - network receiver task n (n=blank for console task)

Task States	Char.	Task Condition	Routine
Not dispatchable	I	Task is inactive	TM10
	P	Page fault in proces	TM10
	O	Waiting for operator response	TM10
Conditionally dispatchable	L	Waiting for locked resource	TM30
	M	Wait on multiple CCB or ECB posting	TM50
	Q	As for M state, except event may never occur	
	C	Wait on single CCB or ECB posting	
	S	As for C state, except event may never occur	
	B	Wait on RJE,BSC or networking event	TMB0
Immediately dispatchable	D	Dispatch task immediately	TM90
Running	R	Task is running	N/A
Partition wait	W	Waiting for dispatch from supervisor	TM20

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). The task class list is shown in Figure 5-20.
	TC#C	Number of class entries.

- Fields used by spool management task

24		One class for spool-mgmt-lst delimiter
25-27	TCEWA	Address of work space for EXTRACT
28	TCIQ	Spool-mgmt queue id
29		Not used
2A	TCSG	Spool-mgmt general-purpose byte
	TC1T	X'02' - 1st time buffered GETSPOOL X'01' - GETSPOOL DASD short on storage message
2B	TCSS	Spool mgmt switch
	TCIW	C'I' - logical writer initialized
	TCOW	C'O' - OPEN logical writer TCCW X'C' - CLOSE logical writer
2C-2F	TCER	Address of user cross-partition (XECB)

- Termination types and function trace indicators

31	TCTT	Termination type
	TT40	b - Normal - continue execution
	TTCU	U - Unrecoverable I/O error
	TTCX	X - TASK cancel condition
	TTCC	C - PCANCEL command issued

	TCNR	X'40' - Nodal message record
	TCKP	X'80' - Leave in queue
	TCNP	X'02' - Do not build virtual buffer control area in front of storage
	TCAE	X'01' - Task accepts I/O errors
36	TCEP	Event post byte
	TCEO	X'80' - Event post bit on
	TCBSCLV	X'40' - Event bit BSC-wait or PNET 'B'
37	TCSI	C'T' if spooling from or to tape

- Each VSE/POWER 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.

38-3B	TCMW	Message request word (see note 2 for message formats). Consists of four bytes. The first byte contains the hold flag and the R5 flag. The remaining three bytes contain the 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.
3C-3F	TCAW	Reply request word (see note 2 for message formats). Consists of four bytes. The first byte contains binary zero. The remaining three bytes contain the reply address. 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.

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. The format of a TCB is as follows.

Bytes Hex.	Label of Field	Description/Function
40-43	TCTR	Register 12 - asynchronous address register ('task PSW'). R12 contains the address of the first instruction to be executed when the task is dispatched. The first byte contains the condition code and the program mask bits in the form in which they

E8-EF	TCQW	Queue file seek address (MBBCCHHR) for C-K-D, or locate word (M0RRBBBB) for FBA. M = X'04'
F0-F3	TCQA	Real queue space address
F4-F7	TCQV	Virtual queue space address

- Tape spooling control words

	TCTS	Tape request word
F8	TCTF	Function byte for open/close tape function (refer to TBFU in tape control block)
F9-FB	TCTA	Address of tape control block
FC	TCTM	Tape unit density specified by operator
FD	TCTDT	Device type code of related tape unit
FE-FF	TCTU	Logical unit of assigned tape unit
100-103	TCPU	Physical tape unit or PUB address

- Various control fields

104-107	TCPL	Address of spool management parameter list
108	TCF2	Flag byte 2
	TCT2S	X'80' - 2nd time switch for IPW$$TR
	TCERT	X'40' - Execution reader task id
	TCWOP	X'20' - Writer-only partition
	TCOFF	X'10' - POFFLOAD task
	TCSLI	X'08' - SLI in process
	TCJBP	X'04' - JOB statement processing
	TCLTP	X'02' - LST statement processing
	TCPUP	X'01' - PUN statement processing
109	TCF3	Flag byte 3
	TCGDS	X'80' - Internally generated data set header record
	TCSGM	X'40' - Segmentation required
	TCDFT	X'20' - Default SETPRT required
	TCIDH	X'10' - Insert data set header record
	TCUNN	X'08' - Queue record with unknown nodeid
	TCOTV	X'04' - Old tape version
	TCNRW	X'02' - No rewind wanted
10A-113		Reserved
114-117	TC3E	Address of TCB extension area
118-11B	TCHD	Head pointer of virtual storage chain
11C-11F		Tail pointer of virtual storage chain
120-123	TC3W	Pointer to 3540 workarea
124-137		Reserved
138-157		General Task Workarea This area may be broken into fields in whatever way is required by a task (for example, logical reader and writer workareas).

- Logical reader re-definitions

138-13F	USCC	Save area for inserted record
140	LWPI	Parameter id byte 1
141	LWPI2	Parameter id byte 2
142	LWPI3	Parameter id byte 3
143	LWOC	Operation code identifier

160-163	TCSADV	Device address where the saved records art to be placed
164-167	TCSAPB	PUB entry address of the saving device
168-16B	TCSAR1	Device specific data passed from command processor task.
16C-16F	TCSART	Linkage register save field
170-173	TCSARN	Second linkage register save field
174-177	TCSADP	Reserved

- TCB expansion area for 2nd data block buffer. If a local printer task is started with the double buffering option, the task is equipped with an expanded TCB to save specific information required. The TCB is enlarged to the next multiple of 32 bytes.

158-15F	TC2DW	2nd data block seek address
160-163	TC2DA	Real address of second data buffer
164-167	TC2DV	Virtual address of second data buffer
168-177		Reserved Second linkage register save field

Notes:

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.

2. The high-order byte of this field will contain the command code of the current or last executed operation.

3. These labels refer to fields in a second LRSA described later in this section under "Second LRSA". The second LRSA has a format identical to that of the LRSA in the TCB.

4. 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).

Overlay for PNET Tasks

This overlay is used whenever the task is a PNET task. It overlays the general task workarea of the standard TCB.

Bytes Hex.	Label of Field	Description/Function
138-13B	TCENCB	Address of the node control block (NCB)
13C-13F	TCENTE	Address of the NCB task entry
140	TCERCB	RCB of the task
141	TCETTC	Termination condition
	TCETSO	X'80' - SIGNOFF record sent or received
	TCETLC	X'40' - line error stop
142-143	TCEFCS	FCS bytes
144-147	TCEWKA	Address of the workarea
148	TCEST1	Status byte 1
	TCERIF	X'80' - RIF sent/received
	TCEPGR	X'40' - Permission granted sent/received
	TCEPRJ	X'20' - Permission rejected sent/received
	TCERCS	X'20' - Receiver cancel sent/received
	TCEEOF	X'10' - End-of-file sent/received
	TCEADS	X'08' - Abort transmission sent/received
	TCECMC	X'04' - Transmission complete sent/received
149	TCEST2	Status byte 2
	TCEWIB	X'80' - Waiting for input buffer
	TCECRTL	X'40' - Compressed record too long
	TCENOP	X'20' - Do not post this task
	TCESPD	X'10' - Task suspended
	TCEPBO	X'08' - Post only after buffer sent
	TCERCA	X'04' - Receiver cancel after abort sent
	TCERAB	X'02' - Release all buffers requested
	TCESOB	X'01' - Short-on-buffer condition
14A-14B		Reserved

- Receiver task

14C-14F	TCERHD	Address of received-input-buffer queue
150-153	TCERTL	Tail ponter of received-input-buffer queue
154	TCENRB	Number of received buffers
155-157		Reserved

- Transmitter task

14C-14F	TCEFOB	Address of -free-output-buffer' queue
150	TCENAB	Number of acquired buffers
151-153	TCETL#	Total number of records/lines
154-157	TCECL#	Current line number

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.

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 5-22 and Figure 5-23 . Refer to VSE/POWER Program Logic Manual 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 the task is executing in the physical routines (PR, PL, PP), is shown in Figure 5-22.

Case 2, where the task is executing in the logical routines (LR, LW), is shown in Figure 5-23.

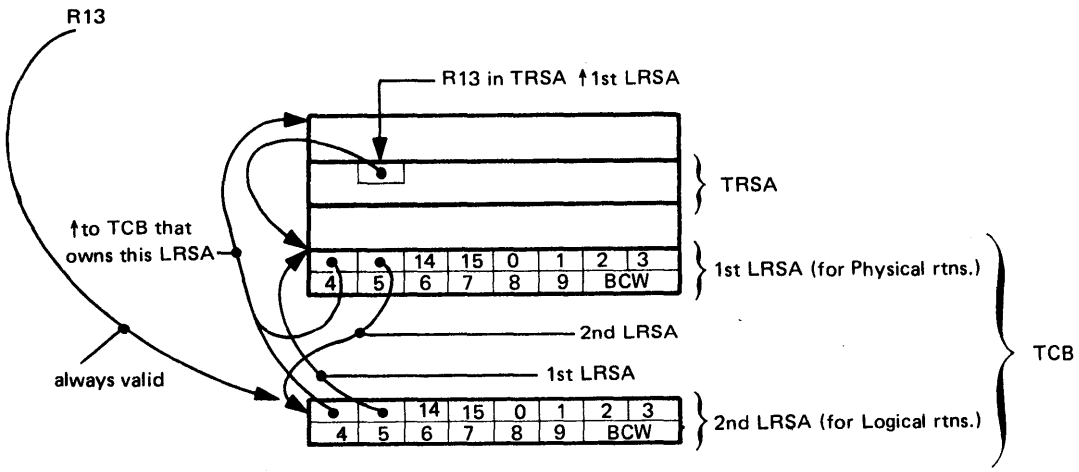


Figure 5-22. Linkage Between the Two LRSAs in a Double Linkage Register Save Area. (Case 1)

TIMER QUEUE ELEMENT (TQE)

This control block is used to control timer intervals set up by a VSE/POWER task. One Timer queue element (TQE) exists for each interval currently setup.

Definition Macro: IPW$DEF TQE=YES

Bytes Hex.	Label of Field	Description/Function
00-03	TQENE	Address of next TQE in chain
04-0B	TQETI	Interval end time (TOD units)
0C-0F	TQEECBP	ECB or address of ECB
10-13	TQEOTCB	Requestor's TCB address
14	TQEFLG	Flag byte
	TQEECB	X'80' - ECB within timer queue element
	TQECAN	X'40' - Cabncel of TQE requested
	TQEACT	X'20' - TQE active
15-17		Reserved for future use

VIRTUAL BUFFER CONTROL AREA (PREFIX)

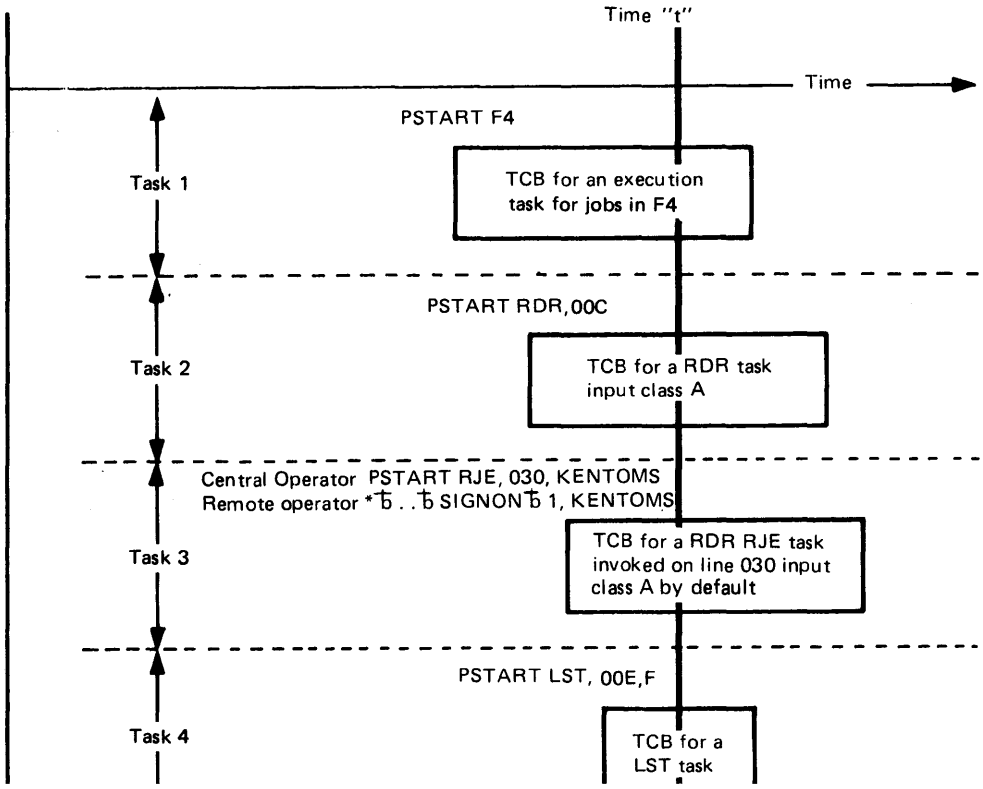
Definition Macro: IPW$DBA

When virtual storage (GETVIS) is required by a VSE/POWER task, storage management precedes each storage area with a buffer control area, which indicates to which pool the storage area belongs.

Bytes Hex.	Label of Field	Description/Function
00-01	BCABL	Length of storage area reserved
02	BCAPID	Pool identifier X'FF' - if freed
03		Reserved
04-07	BCAFWD	Pointer to next storage element
08-0B	BCABWD	Pointer to previos storage element
0C-0F		Reserved

RELATIONSHIPS AMONG VSE/POWER CONTROL BLOCKS AND DATA AREAS

Figure 5-26 to Figure 5-36 contain a set of examples which illustrate the interrelationships between the VSE/POWER 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.



160-163	TCSADV	Device address where the saved records art to be placed
164-167	TCSAPB	PUB entry address of the saving device
168-16B	TCSAR1	Device specific data passed from command processor task.
16C-16F	TCSART	Linkage register save field
170-173	TCSARN	Second linkage register save field
174-177	TCSADP	Reserved

- TCB expansion area for 2nd data block buffer. If a local printer task is started with the double buffering option, the task is equipped with an expanded TCB to save specific information required. The TCB is enlarged to the next multiple of 32 bytes.

158-15F	TC2DW	2nd data block seek address
160-163	TC2DA	Real address of second data buffer
164-167	TC2DV	Virtual address of second data buffer
168-177		Reserved Second linkage register save field

Notes:

1. The first characters of the labels in the control block vary according

Note:

[1] See Figure 5-29
[2] See Figure 5-30
[3] See Figure 5-31
[4] See Figure 5-32
[5] See Figure 5-34

Before proceeding it is necessary to understand the method of presentation of these examples. Each table or block 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 in Figure 5-28.

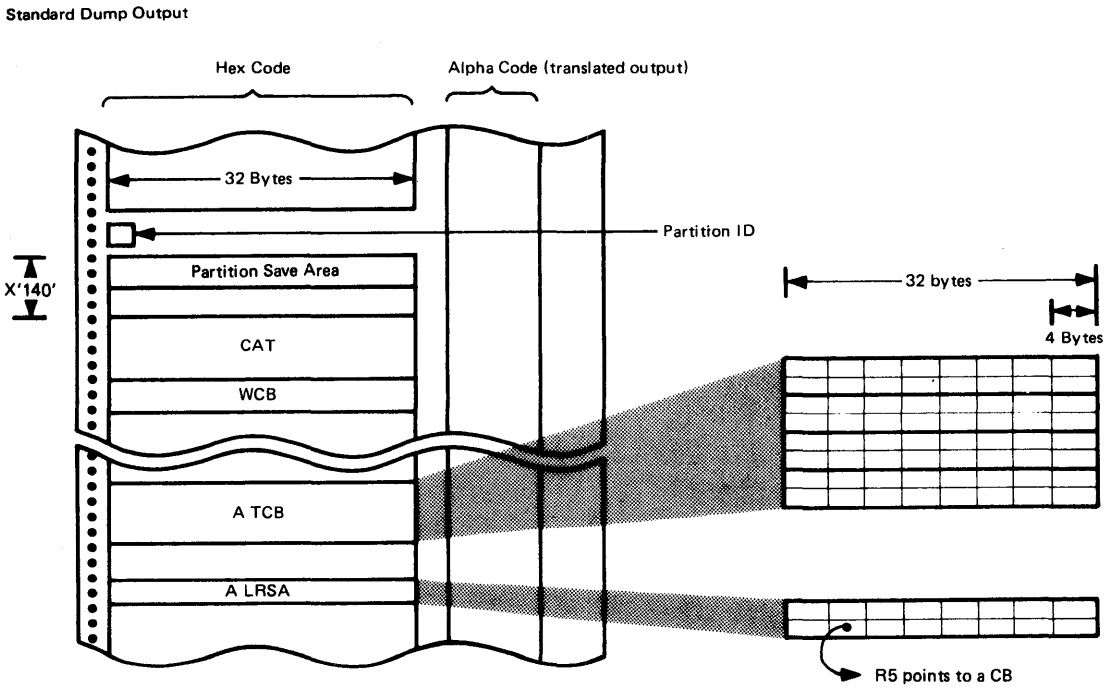


Figure 5-28. Method of Presentation and Task Conditions

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 Figure 5-29 to Figure 5-36.

Task 1

- The permanent operator command processor TCB in the permanent area is in I state waiting for the operator to enter a command.

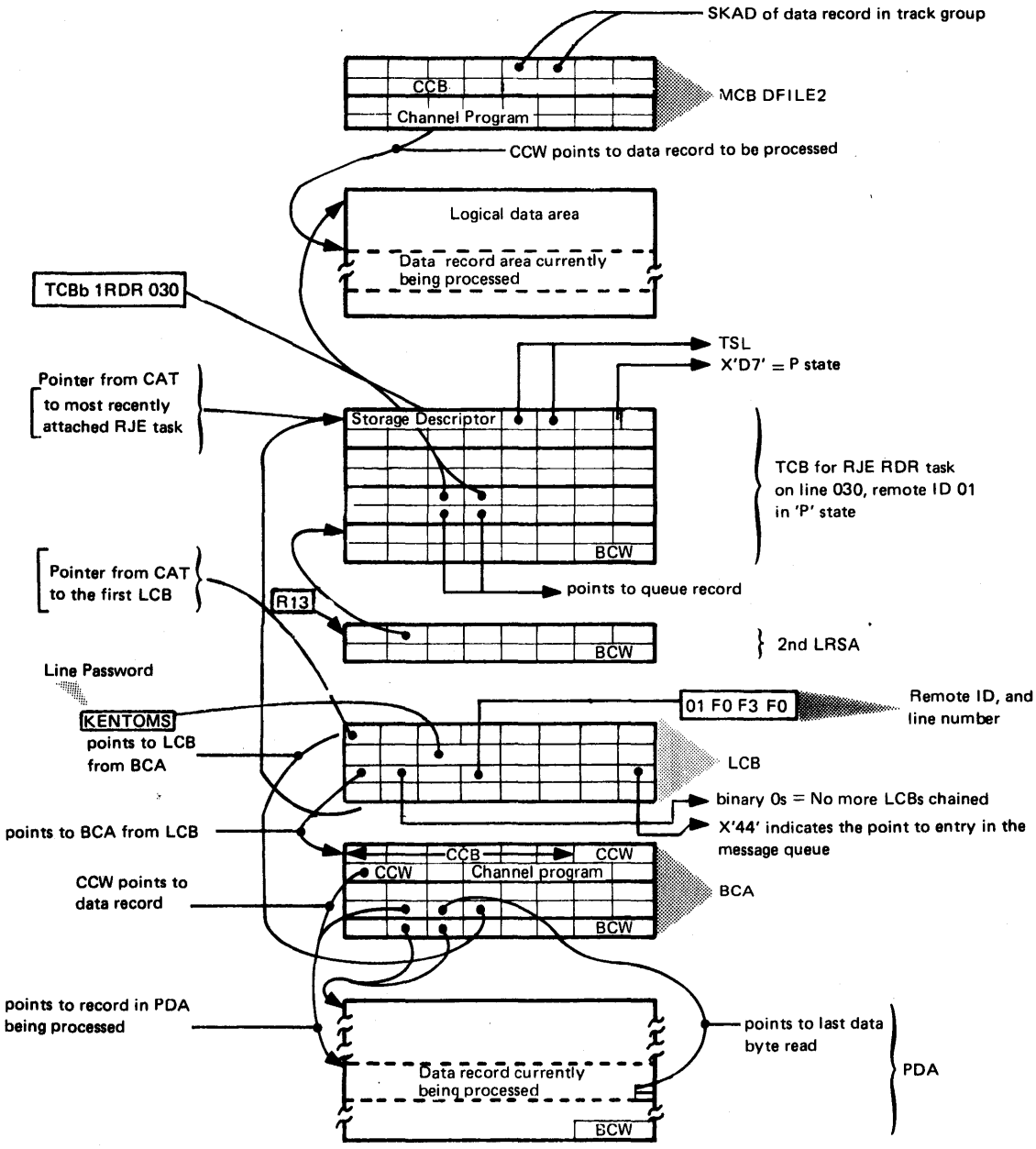
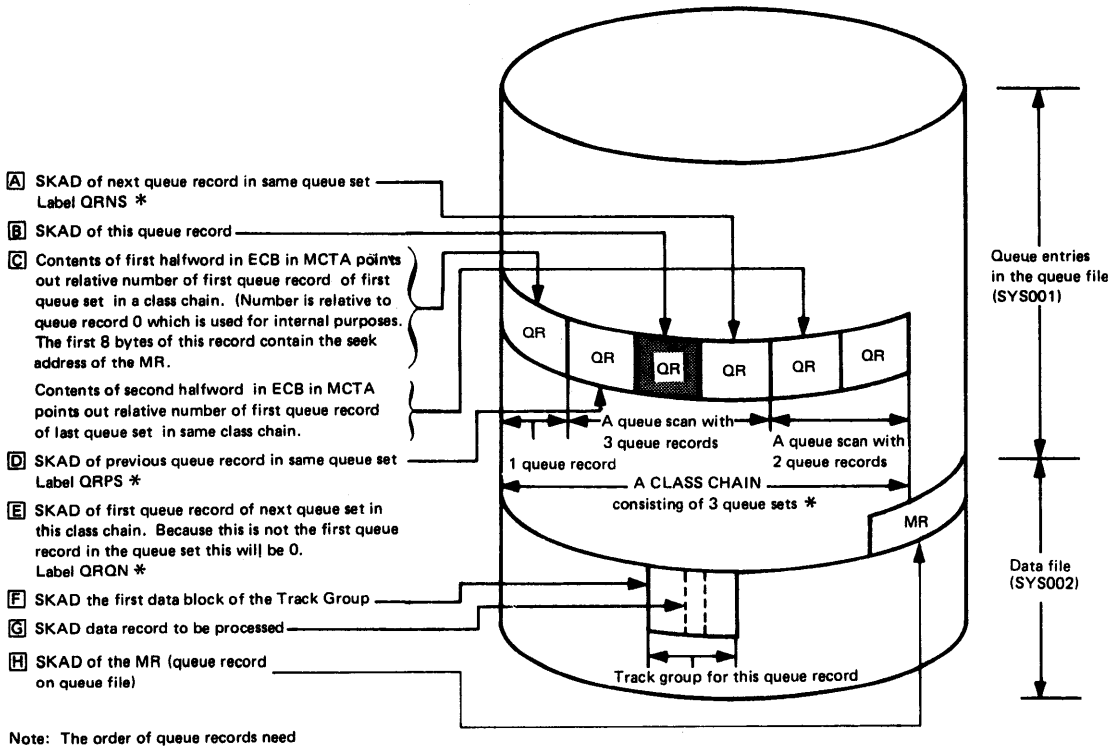


Figure 5-31. Relations Between Data Areas in Use by Task 3



Note: The order of queue records need not be continuous on the file as shown in this illustration.

*See Figure 5-34.

Figure 5-33. The Queue and Data Files of Task 4

The queue record being processed (shaded) is the second queue record of the second queue set in a class chain (F).

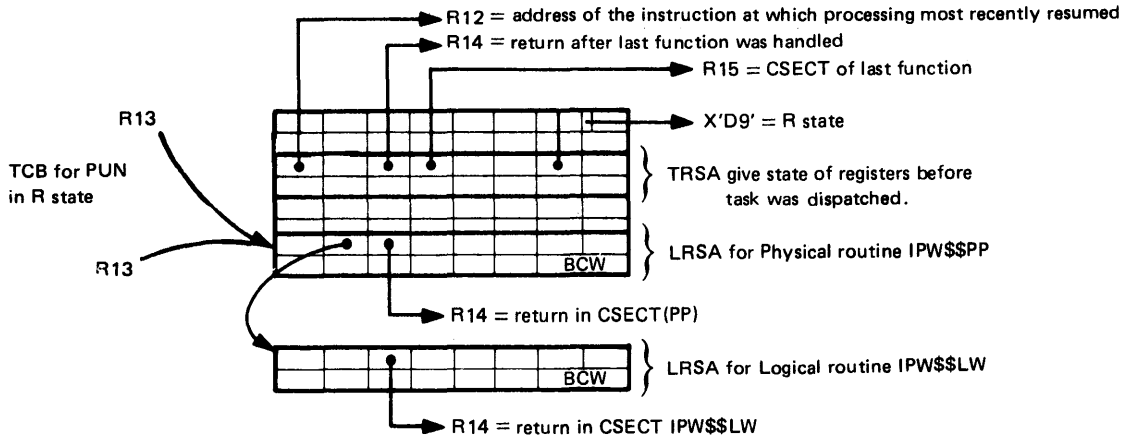


Figure 5-35. Relations Between Data Areas in Use by Task 5

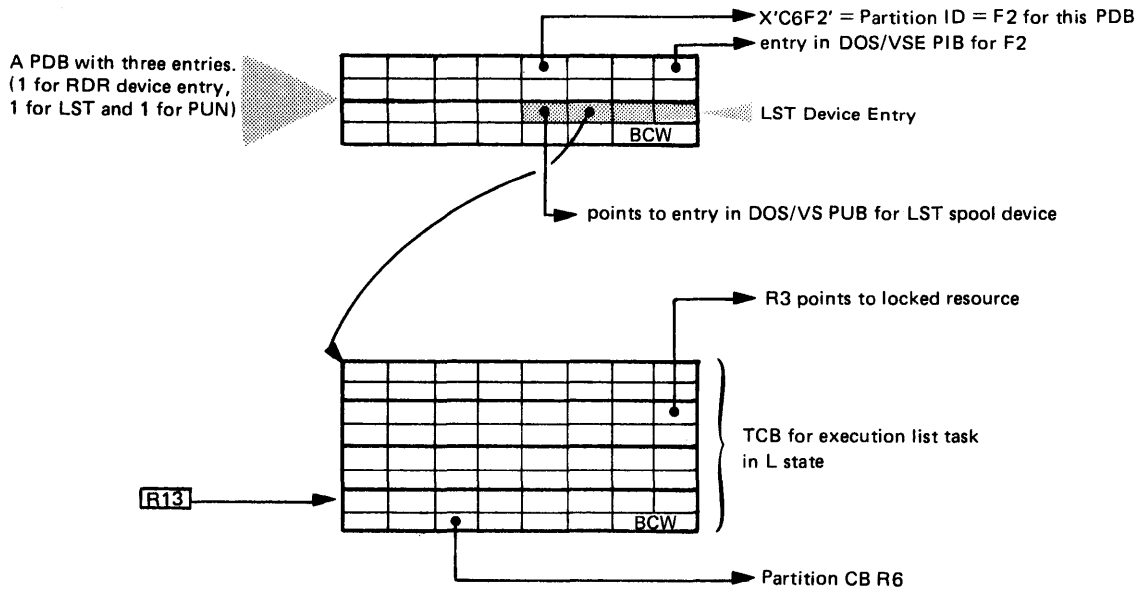


Figure 5-36. Relations Between Data Areas in Use by Task 6, Showing the PDB for this Execution List Task

6. DIAGNOSTIC AIDS

This section consists of error diagnostic flowcharts that indicate a method of dump analysis to isolate the cause of a program error in VSE/POWER.

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 VSE/POWER partition. The section begins with general debugging hints, a list of which follows.

- The standalone dump (DOSVSDMP)
- Identifying the VSE/POWER partition (the partition in which VSE/POWER is initialized)
- Identifying pages belonging to the fixable area
- Identifying the start of the pageable area
- Locating and identifying control blocks, tables and areas
- Identifying the start of a CSECT
- Establishing the "level" of a CSECT
- Determining the active routine and analyzing the register save areas.
- Analyzing event control blocks
- Using the buffer control words
- Analyzing TCBs
- Establishing queue records in queue sets and class chains
- RJE,BSC and PNET trace facility
- PNET BSC I/O logging on console
- VSE/POWER file dump program
- Establishing the last command issued
- An aid to eliminate components
- Problems related to VTAM.
- Queue and data file chaining error trapping

GENERAL DEBUGGING HINTS

Standalone Dump

It is recommended that the user generate a formatted standalone dump with translation (DOSVSDMP).

This dump should always be used when a standalone dump is required. Formatted page tables will save time in converting virtual to real addresses.

Identifying the VSE/POWER Partition

The start of the VSE/POWER 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.

Abbreviated * Mnemonic of Table or Area	Pointer to or Address of the Table/Area	Identifier in Translated Dump
Account file seek address	X'200' - X'207' of DMB	
Account control block	X'1C8' of partition	
Auxiliary Queue record	X'90' - X'93' of DMB	Real address
	X'94' - X'97' of DMB	Virtual address
Command proc. control blk.	X'138' of the CP TCB	
Command proc. TCB	X'258' of partition	TCBbObCP
DLRSA (or second LRSA)	R13 in TRSA for task executing code in logical routines X'4' - X'7' in logical routines X'4' - X'7' in LRSA (in TCB) for task executing code in physical routines.	
End address of VSE/POWER partition	Subtract 1 from contents of X'160' - X'163' of part.	
First fixed page	X'14' of SCB	
INIT/TERM TCB	X'25C' of partition	TCBbIbIT
Last permanent page	X'10' of SCB	
LCB	X'464' of partition	
LDA	X'D0' of a TCB (only if applicable)	Real address
	X'D4' - X'D7'	Virtual address
LMF (Line Manager Fields)	X'148' - X'157' of a RJE Line Manager task TCB	
LRSA	X'98' - X'C7' of a TCB for an RDR, LST, PUN, or XP task	This LRSA saves R14-R9 used by the physical routines
MCB for Q file	X'1FC' of partition	MCB QFILE 1
MCB data file 1	X'200' of partition	MCB DFILE 2
MCB data file 2	X'204' of partition	MCB DFILE 3
MCB data file 3	X'208' of partition	MCB DFILE 4
MCB data file 4	X'20C' of partition	MCB DFILE 5
MCB data file 5	X'210' of partition	MCB DFILE 6
Remote message contr. blk.	X'1D4' of partition	MSCB
Partition control block	X'A0' - X'A3' of the partition COMREG	PDB
Page control block	First 24 bytes of each page in fixable storage	

Figure 6-2 (Part 1 of 3). Locating and Identifying Control Blocks, Tables, and Areas in the Fixable Area

Determining the Active Routine and Analyzing Register Save Areas

It is important to know the routine in which a task is executing 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

Licensed Material - Property of IBM

160-163	TCSADV	Device address where the saved records art to be placed
164-167	TCSAPB	PUB entry address of the saving device
168-16B	TCSAR1	Device specific data passed from command processor task.
16C-16F	TCSART	Linkage register save field
170-173	TCSARN	Second linkage register save field
174-177	TCSADP	Reserved

- TCB expansion area for 2nd data block buffer. If a local printer task is started with the double buffering option, the task is equipped with an expanded TCB to save specific information required. The TCB is enlarged to the next multiple of 32 bytes.

158-15F	TC2DW	2nd data block seek address
160-163	TC2DA	Real address of second data buffer
164-167	TC2DV	Virtual address of second data buffer
168-177		Reserved Second linkage register save field

Notes:

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.

2. The high-order byte of this field will contain the command code of the current or last executed operation.

3. These labels refer to fields in a second LRSA described later in this section under "Second LRSA". The second LRSA has a format identical to that of the LRSA in the TCB.

4. 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).

PNET,SNA TRACE RECORD

Bytes Hex.	Data Length	Format	Data Description
00-07	8	char	Remote node name
08-1F	24	binary	Buffer header
20-83	100	binary	ACF/VTAM RPL
84-87	4	packed	Time of day (0HHMMSSF)
88-97	16	char	First 16 bytes of data
98-A7	16	char	Last 16 bytes of data
• Status of node			
A8-A9	2	binary	Action bytes 1, 2
AA	1	binary	Process byte
AB-AC	2	binary	Status bytes 1, 2
AD	1	-	Reserved
AE	1	binary	Node termination code
AF	1	binary	Termination subcode
B0	1	binary	Termination code qualifier
B1-B3	3	-	Reserved
B4-B5	2	binary	Number of receivers currently active
B6-B7	2	binary	Number of transmitters currently active
B8-BB	4	binary	Address of free input buffer queue
BC-BF	4	binary	Address of to-be-sent queue
C0-C3	4	binary	Tail pointer to-be-sent output queue
C4-C7	4	binary	Address of receive buffer
C8-CB	4	binary	Address of send buffer
CC-CD	2	binary	Buffer size
CE	1	binary	Maximum number of input buffers
CF	1	binary	Maximum number of output buffers
D0-D1	2	binary	Number of acquired input buffers
D2-D3	2	-	Unused
D4-D7	4	binary	Address of input buffer in use
D8-DB	4	binary	Address of output buffer sent
DC-E7	12	-	Reserved
E8-EB	4	binary	Address of suspended buffer queue
EC-EF	4	binary	Address free input buffer ahead queue
F0-F3	4	binary	Address send ahead queue
F4	1	binary	Send gate
F5	1	binary	Receive gate
F6-FA	5	binary	Session status byte 1 to 5
FB	1	-	Reserved
FC-FF	4	binary	Task-id of task writing trace record

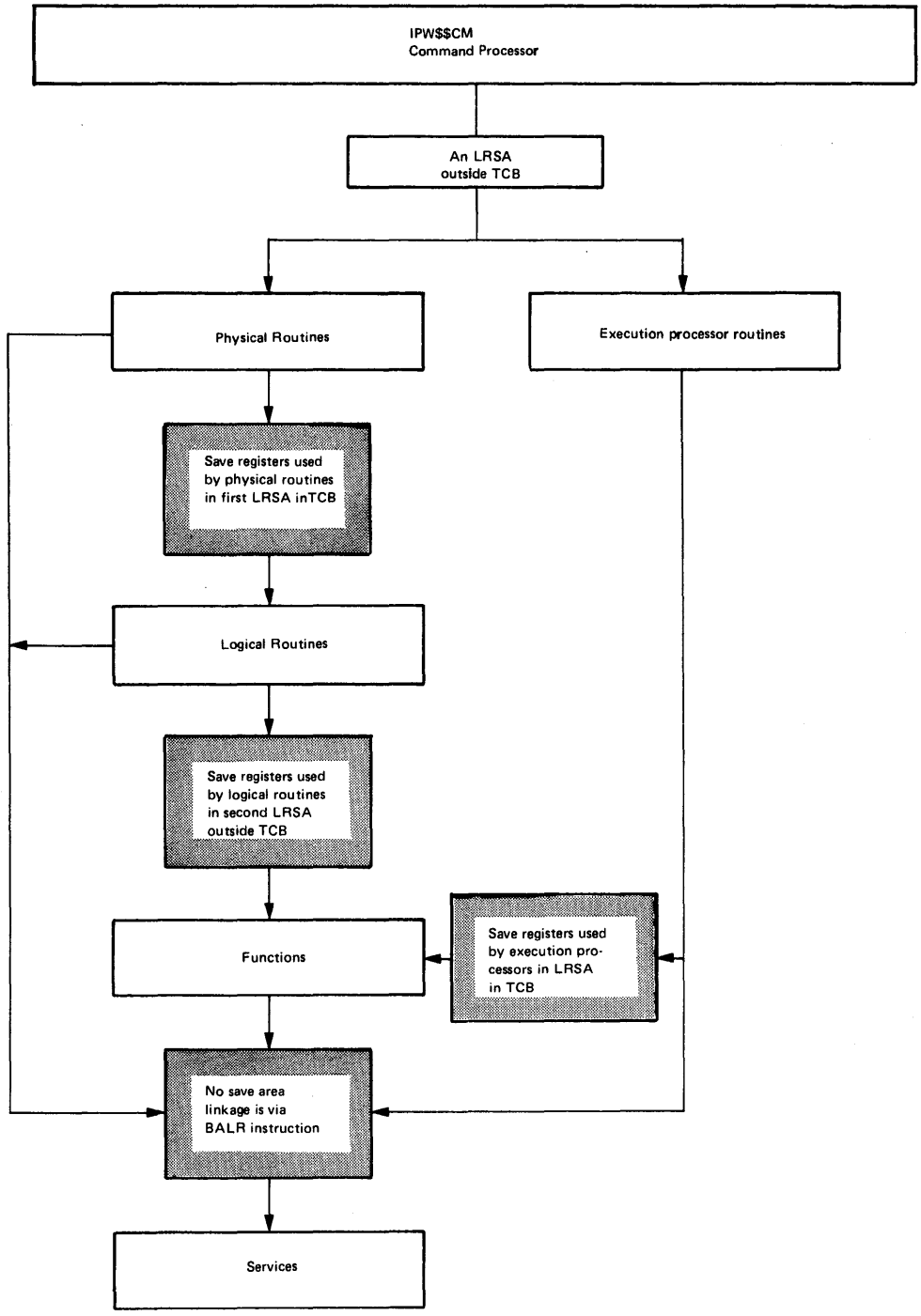


Figure 5-21. Summary of Linkage Register Save Areas

central operator's console at I/O completion. The operator can request I/O logging either at VSE/POWER start-up time via the // UPSI 01 statement or while VSE/POWER is running by turning on bit 1 (X'40) in the byte at displacement X'4AD' of the VSE/POWER partition.

```
LINE=cuu CCB=cccc SENSE=ssss OUT: BCB=bb FCS=ffff BSC=aaaa RCB=rrrr
         IN: BCB=bb FCS=ffff BSC=aaaa RCB=rrrr
```

Figure 6-6. I/O Logging Record Format

The meanings are as follows:

LINE=cuu is the 2-byte device address in binary format

CCB=cccc are the CSW Status bits

SENSE=ssss is the 2-byte sense information as returned from the device.

OUT or IN is the direction of the buffer

BCB=bb Is the block control byte either sent or received. See Figure 3-53 for the meaning of the byte.

FCS=ffff are the 2 bytes of the function control sequence field See Figure 3-52 for the meaning of the byte.

BSC=aaaa are the first two BSC control characters used in the buffer being sent or received. Valid characters are:

- X'1070' - ACK0
- X'3D00' - NAK
- X'1002' - start of text (data)
- X'012D' - SOH ENQ

RCB=rrrr Is the record control byte (RCB) and subrecord control byte (SRCB) of the first record in the buffer sent or received. See Figure 3-50 and Figure 3-51 for the meanings of the bytes.

Hardware Error Recording

Error recording is part of ERP processing. The error data is placed in the system error recorder file (SYSREC) for subsequent editing and printing. Error recording takes place under following conditions:

- BSC RJE line

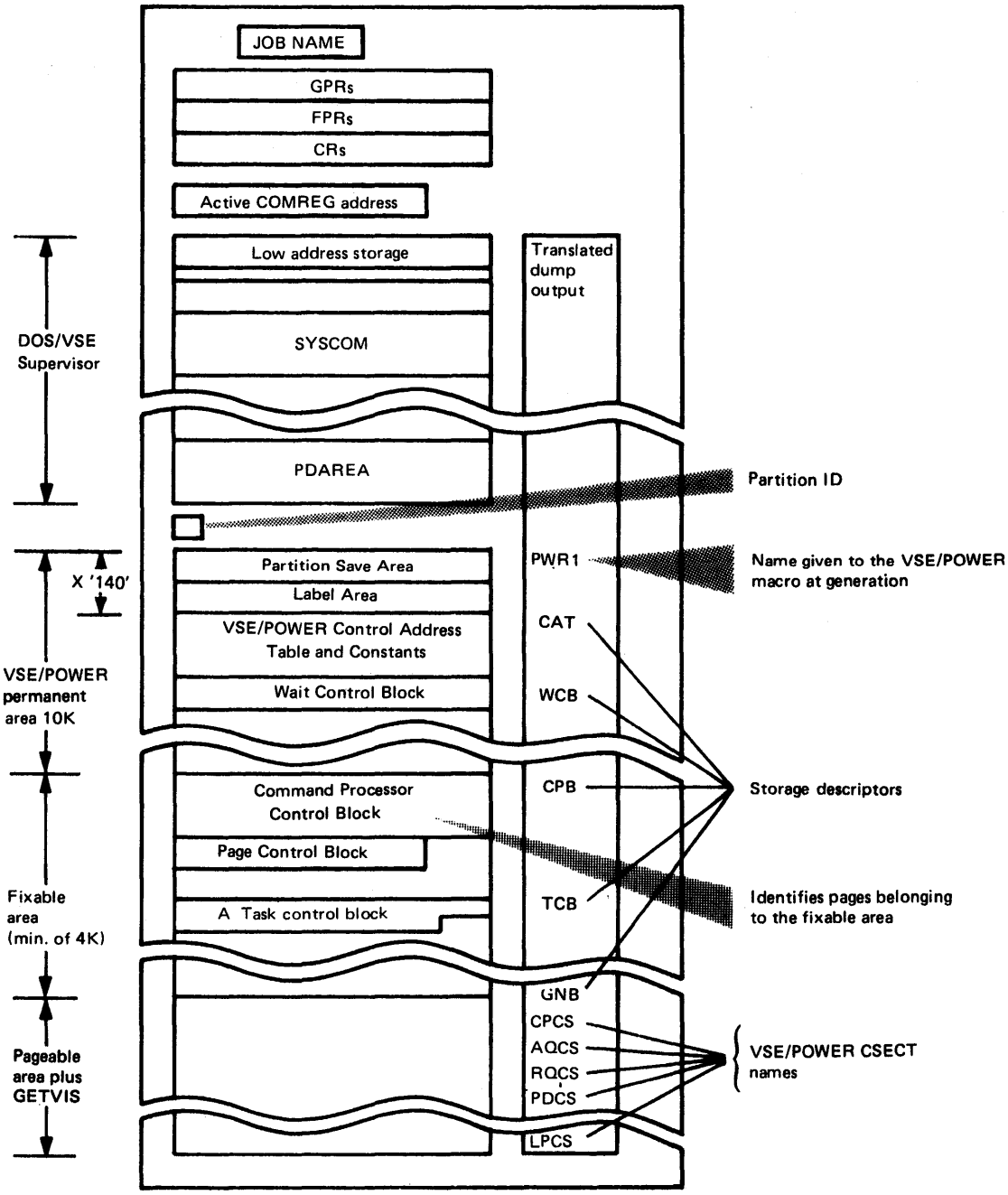


Figure 6-7. Pictorial Representation of a System Dump Containing the VSE/POWER Partition

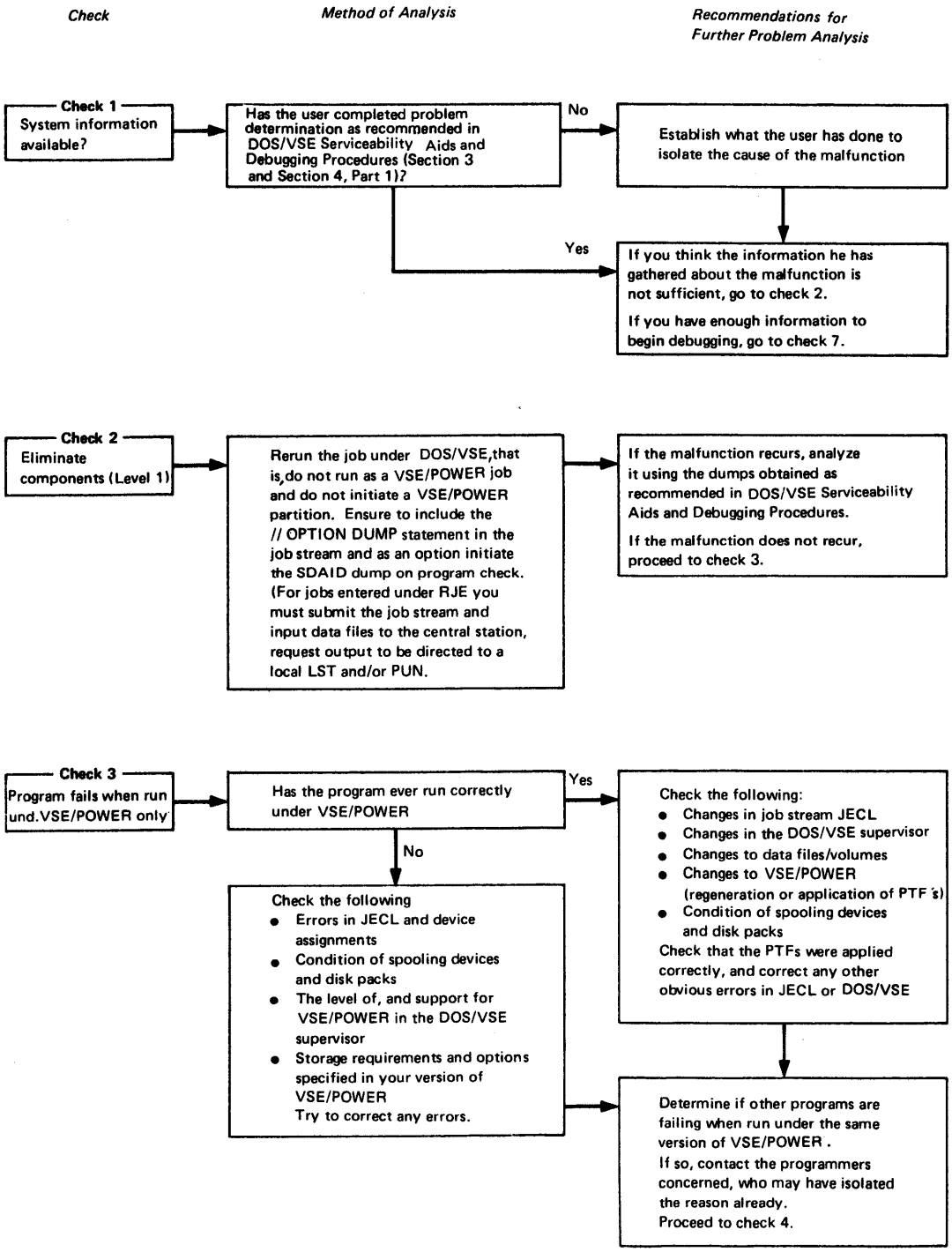


Figure 6-8 (Part 1 of 3). Initial Environmental Checks

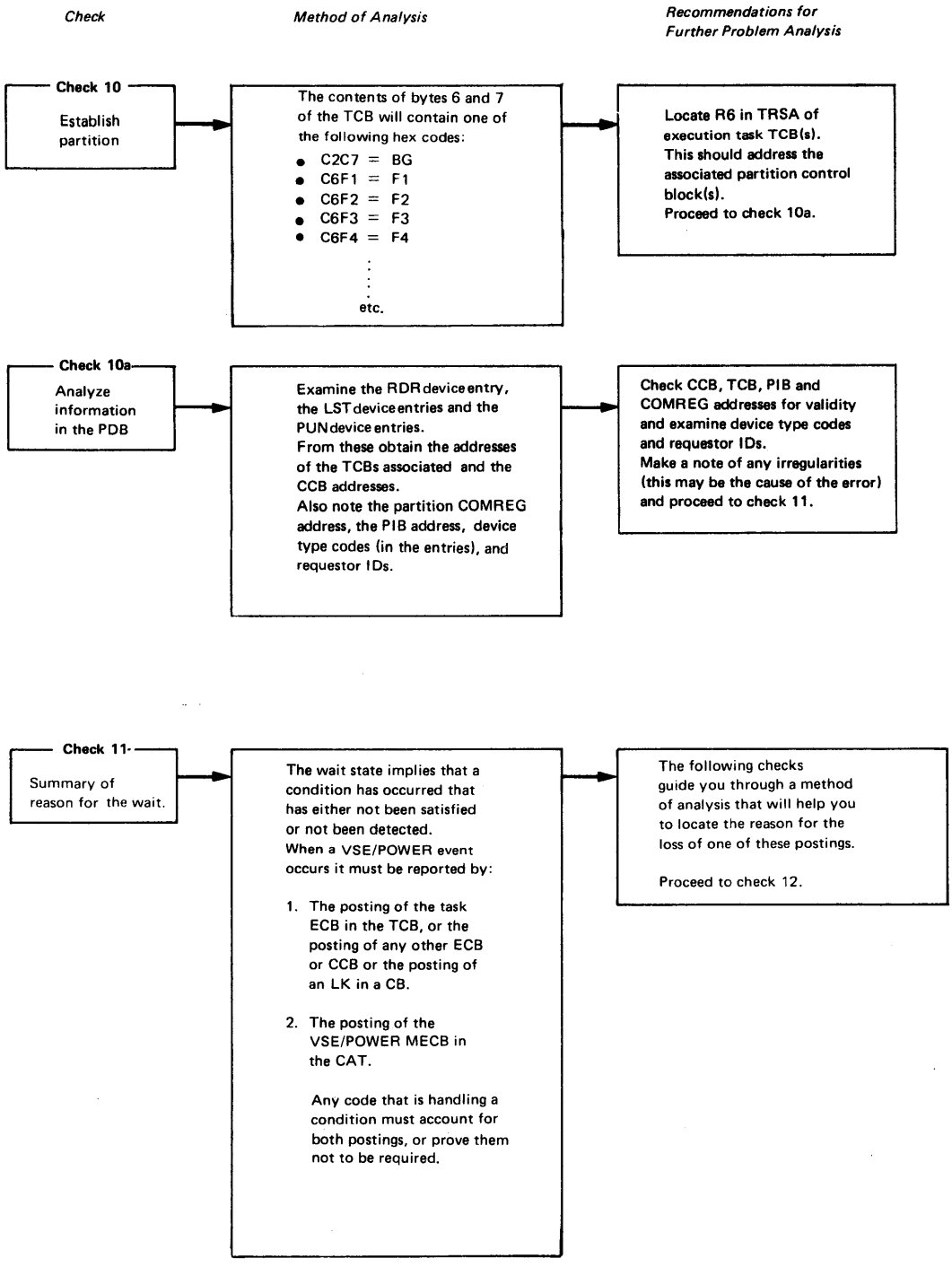


Figure 6-9 (Part 4 of 9). Wait State

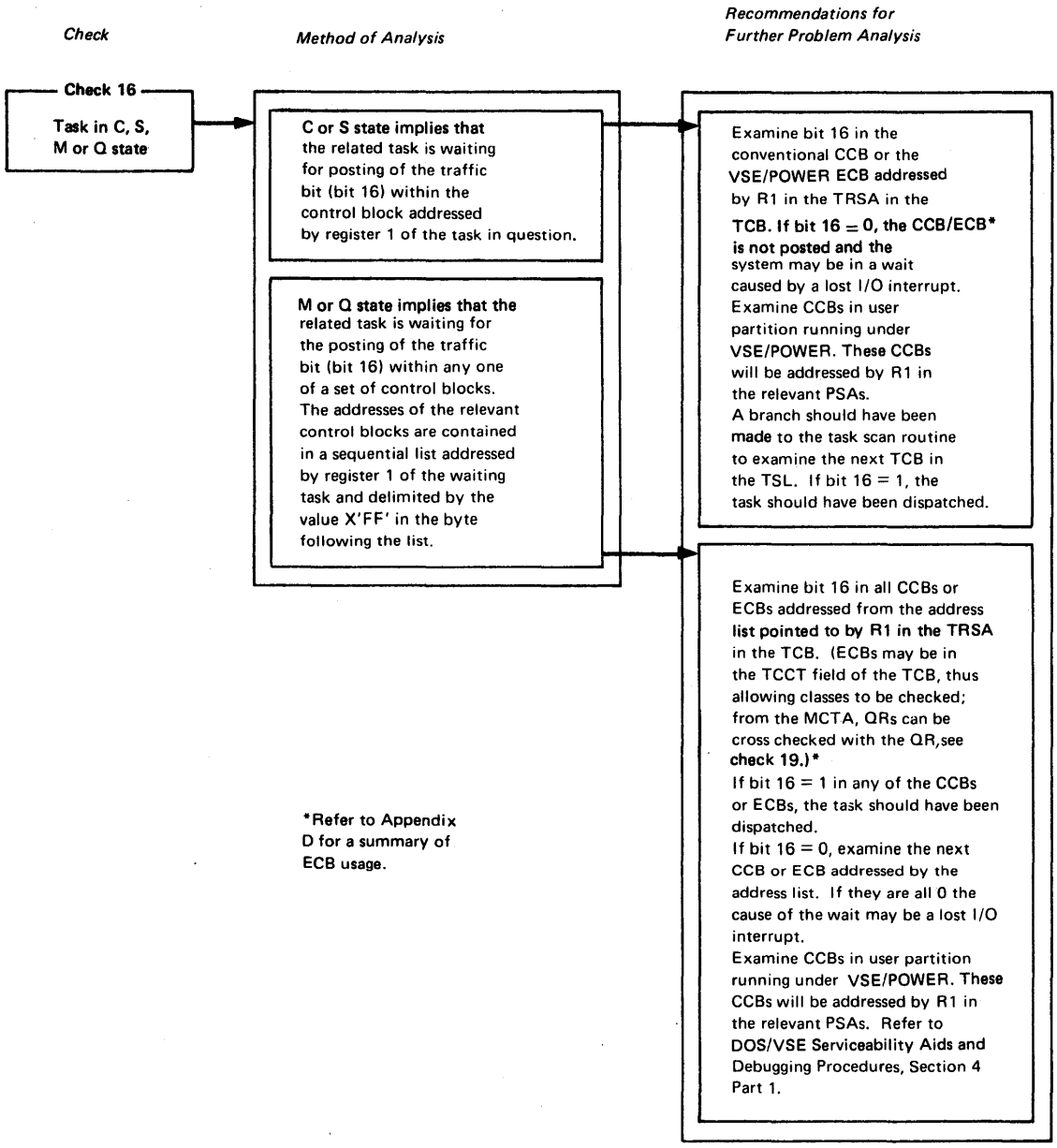


Figure 6-9 (Part 8 of 9). Wait State

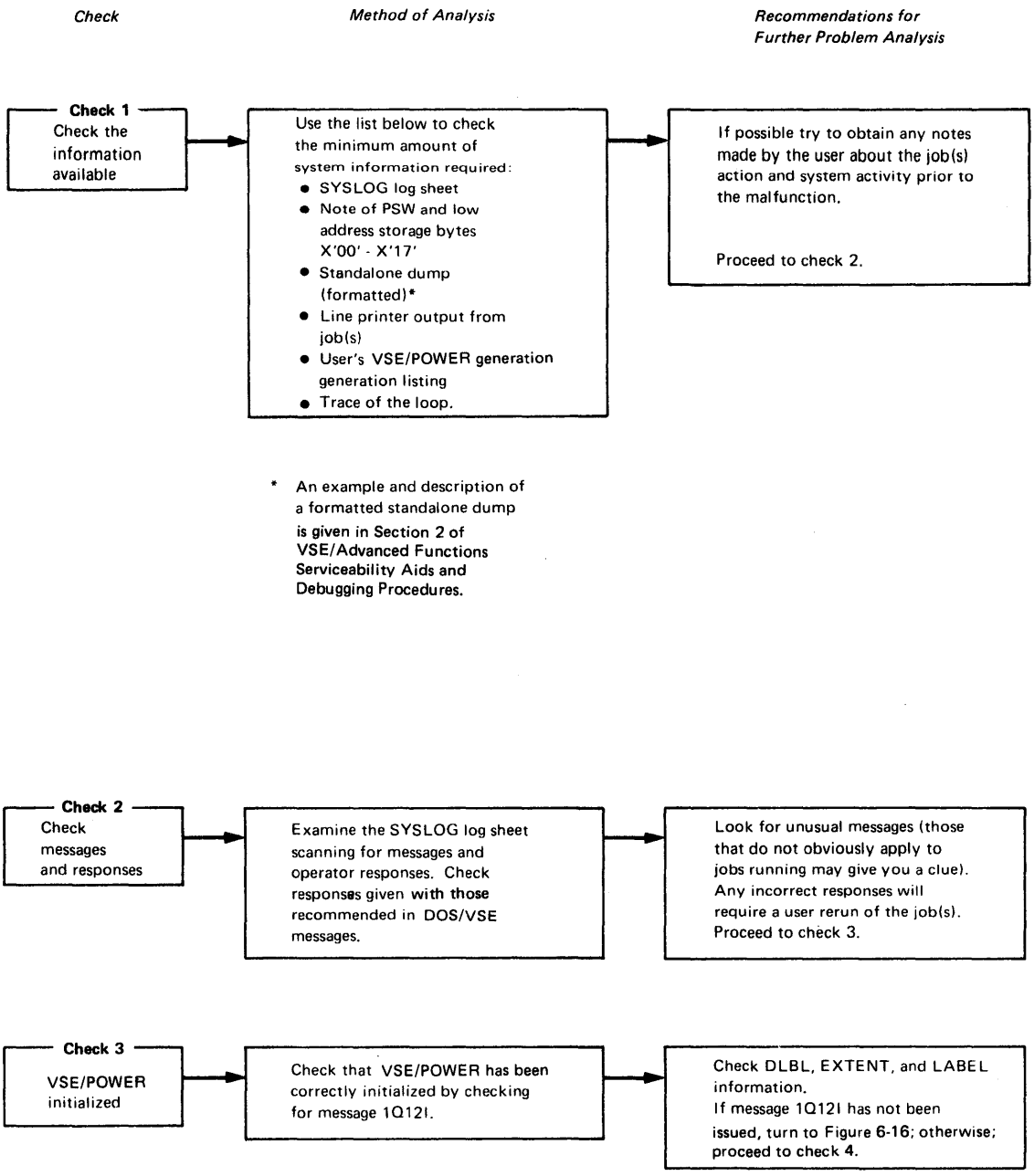


Figure 6-10 (Part 1 of 2). Loop

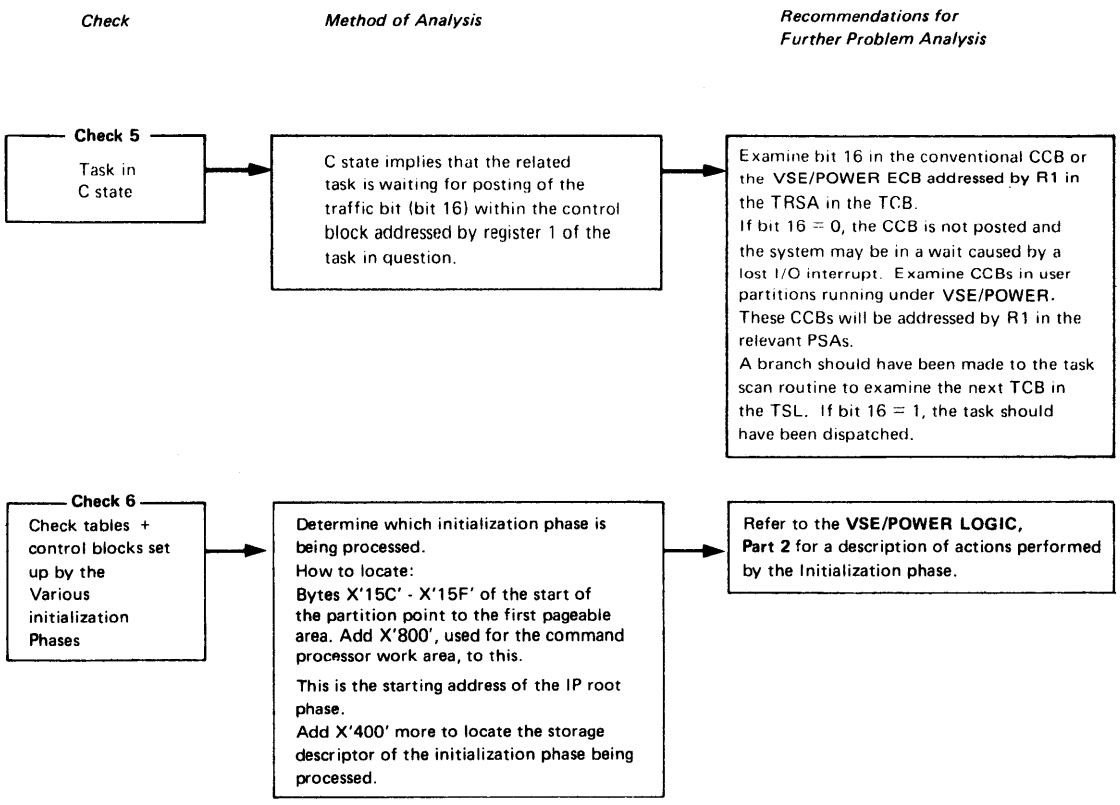


Figure 6-13 (Part 2 of 2). VSE/POWER Not Initialized

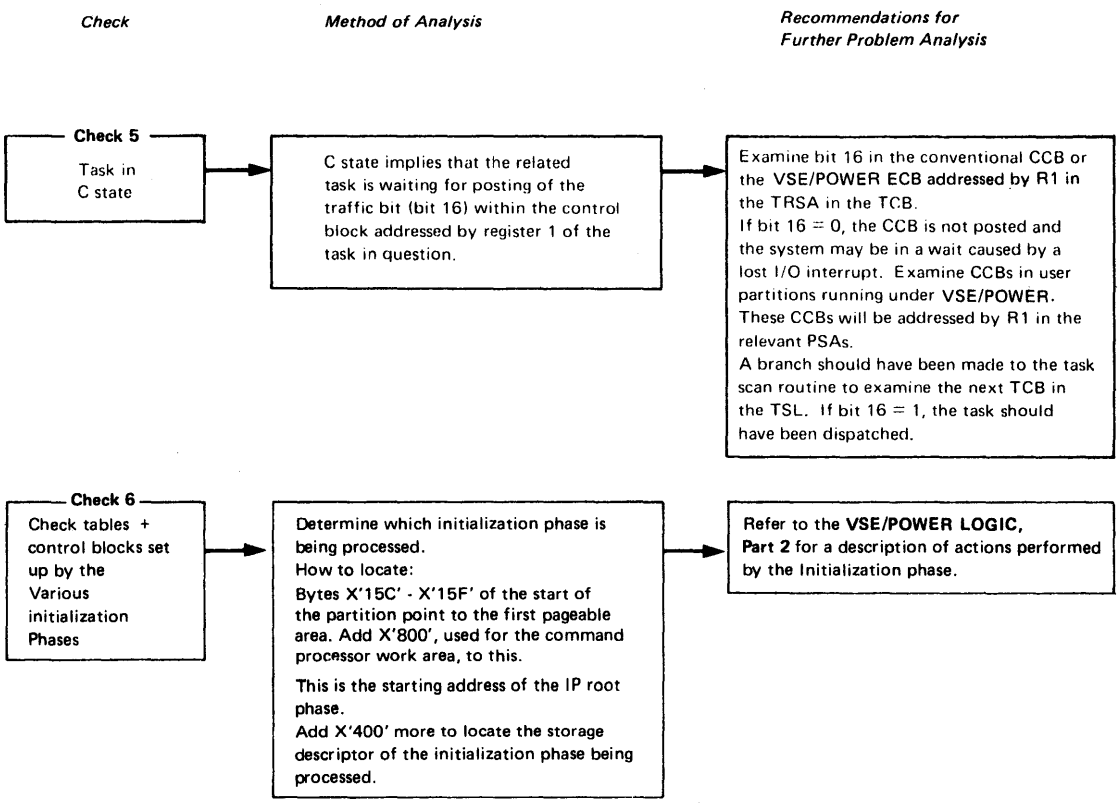


Figure 6-13 (Part 2 of 2). VSE/POWER Not Initialized

APPENDIX A. VSE/POWER STATUS BYTES IN THE VSE/ADVANCED FUNCTIONS SUPERVISOR

SYSCOM*

Location X'5C' - X'5F' contains the address of the VSE/POWER partition (label IPW$PDA in phase IPW$$NU), if VSE/POWER is initiated.

Location X'42' contains a flag byte:

X'08' = VSE/POWER supported
X'04' = VSE/POWER initialized

Partition COMREGS*

Location X'A0' - X'A3' contains the address of the partition control block (0 if no CB exists for this partition).

VSE/POWER FLAG BYTES

Location X'A4' contains VSE/POWER flags:

X'80' = VSE/POWER accounting support
X'40' = This partition under control of VSE/POWER
X'20' = This partition is the VSE/POWER partition
X'08' = Partition-waiting-for-work state

Location X'A5' reserved

* Refer to VSE/Advanced Functions Diagnosis: Service Aids, SC33-6099, for a full description and locations of SYSCOM and the partition COMREGs.

APPENDIX B. SUMMARY OF ECB USAGE

ECB usage is summarized in Figure B-1

ECB in:	Posted by : (Phase)	Unposted by:	Use when posted :
ACB	IPW$$GA/IPW$$SF	IPW$$PA/PF	account file is empty
CAT	Appendage	Task select.	Indicates work-to-do. for VSE/POWER
SCB	IPW$RLW	IPW$RSW	Work space is avail.
VSCB	IPW$RLV	IPW$RSV	Virtual storage avail.
DMB	IPW$$FQ	IPW$$RQ, IPW$$PD	Queue space is avail.
TCB (CP)	IPW$$I7	IPW$$CM	Indicates that IPW$$I7 has sent information to IPW$$CM
TCB (LD)	all PNET tasks IPW$$AQ, IPW$$MS, IPW$$CPS, IPW$$CP	IPW$$LD	If work is to do for PNET Driver
TCB (LMGR)	• Channel End Appendage • line start • line stop	IPW$$LM	work-to-do for line manager.
TCB (OB)	IPW$$IB, IPW$$MP, IPW$$SN	IPW$$OB	Indicates that trans. to SNA terminal which was previously suspended is to continue.
TCB (SN)	ACF/VTAM at completion of a RECEIVE ANY	IPW$$SN	Indicates that IPW$$SN must attach IPW$$IB
TCB (SN)	IPW$$SN, IPW$$LN, IPW$$IB, IPW$$OB, IPW$$MP, IPW$$LF, IPW$$VE, IPW$$LH, IPW$$MS	IPW$$SN	Indicates work-to-do for IPW$$SN
SRB	IPW$$AS (Subtask)		Indicates that service request is processed
ASAB	IPW$$AS	IPW$$AS (Subtask)	Indicates that service request is waiting to be processed.

Figure B-1. Summary of ECB Usage

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.

APPENDIX C. VSE/POWER STATUS REPORT

This report is printed on two occasions:

1. When VSE/POWER is initiated, when SYSLST is assigned to a line printer in the VSE/POWER partition.

2. When VSE/POWER is terminated with a PEND command that includes a physical address of a line printer.

The format for C-K-D devices is shown below. The format for FBA devices is similar but definitions shown below for tracks are for FBA blocks.

```
QUEUE FILE   IJQFILE

TOTAL NUMBER OF QUEUE RECORDS                                  xxxx RECORDS
NUMBER OF FREE QUEUE RECORDS                                   xxxx RECORDS
MAXIMUM NUMBER OF QUEUE RECORDS USED IN PRESENT SESSION        xxxx RECORDS

DATA FILE   IJDFILE

TOTAL NUMBER OF TRACKS                                          xxxx TRACKS
TRACK GROUP SIZE                                                 xx TRACKS
DATA BLOCK SIZE                                                  xx BYTES

ACCOUNT FILE   IJAFILE

TOTAL NUMBER OF TRACKS                                          xxxx TRACKS
PERCENTAGE OF FILE THAT IS FILLED                                     xx %

FIXABLE STORAGE ALLOCATED TO VSE/POWER PARTITION                 xx PAGES
NUMBER OF TIMES TASKS WERE WAITING FOR PFIXED STORAGE           xxx TIMES
MAXIMUM NUMBER OF PAGES FIXED                                    xx PAGES
MAXIMUM NUMBER OF TASKS ACTIVE AT ONE POINT IN TIME              xx TASKS
NUMBER OF TIMES TASKS WERE WAITING FOR VIRTUAL STORAGE          xxx TIMES

TIME INTERVALS FOR SHARED SPOOLING (SYSID=n) :
T1 =   x SECONDS, T2 =   x SECONDS, T3 =    x SECONDS, T4 =   x SECONDS

NUMBER OF NOTIFY MESSAGES LOST:                              xxxxx MESSAGE(S)
```

Figure C-1. Status Report

APPENDIX D. FORMAT OF INTERNAL MACROS

IPW$BUF - PNET Buffer Management

This macro is used as linkage to the buffer service routine (IPW$$BS).

The IPW$BUF macro has the following format:

```
                        CNTRL
                        FREE
                        GET
                        HEADQ
  &name   IPW$BUF   TYPE=MSG
                        QUEUE
                        PURNR
                        PURNT
                        RELEASE
                    ,MODE=IN
                          OUT
                  <,REG=(R1)>
                  <,WAIT=NO>
                        =YES
```

REG — is optional but if specified means that the address of the buffer that is to be referenced is already loaded into R1 and the address in the NCB should not be used. If not specified then the address from the NCB will be used. This keyword is only valid for TYPE=QUEUE,MODE=OUT.

WAIT — YES specifies that the task wants to be placed in the wait state until a buffer is available.
NO means that control will be returned directly to the calling task, with or without a buffer being available.
YES is default.

Register usage is as follows:

R0 — function code, describing the type of function to be performed.

01 - GET - MODE=IN

02 - RELEASE - MODE=IN

03 - FREE - MODE=IN

04 - QUEUE - MODE=IN

05 - GET - MODE=OUT

- Register 1 is zero, then ALL buffers are removed from the queue.
- Register 1 non zero then only buffers belonging to this task are removed from the queue.

If the buffer has the 'release-after-sent' flag set then the buffer is returned to the VSE/POWER storage pool. All other buffers are queued into the 'free output queue'.

8. QUEUE

   causes the TP-output buffer, addressed by register 1, to be queued as the last entry in the 'to-be-sent chain'. If the buffer just queued is the first in the queue, then the line driver is posted.

   If a line error has been detected or a 'SIGNOFF' record has been received, then the buffer is added to the 'free output queue'.

9. RELEASE

   causes all TP-buffers in the 'free output chain' (anchored to this NCB task entry), to be released and returned to the VSE/POWER storage pool.

IPW$GMS - General Message Service

The macro IPW$GMS must be used to invoke the general service function for all messages and commands that should be routed to the correct nodeid/user. The command/message must be in NMR format.

The format of the macro is as follows:

```
  &name   IPW$GMS  TYPE=DIST,NMR=(Rx)
                   TYPE=SUB
```

TYPE — DIST causes the link to the general message distribution service.

NMR — Rx is the address of the nodal message record that must be distributed.

TYPE — SUB causes a link to the general message service routine to do message substitution.

The register usage for TYPE=SUB is as follows:

R0 — message length - 1.

R1 — address of message.

R4 — address of message control table.

R5 — TCB address to be used for message modification.

- Register 1 is zero, then ALL buffers are removed from the queue.
- Register 1 non zero then only buffers belonging to this task are removed from the queue.

If the buffer has the 'release-after-sent' flag set then the buffer is returned to the VSE/POWER storage pool. All other buffers are queued into the 'free output queue'.

8. QUEUE

   causes the TP-output buffer, addressed by register 1, to be queued as the last entry in the 'to-be-sent chain'. If the buffer just queued is the first in the queue, then the line driver is posted.

   If a line error has been detected or a 'SIGNOFF' record has been received, then the buffer is added to the 'free output queue'.

9. RELEASE

   causes all TP-buffers in the 'free output chain' (anchored to this NCB task entry), to be released and returned to the VSE/POWER storage pool.

IPW$GMS - General Message Service

The macro IPW$GMS must be used to invoke the general service function for all messages and commands that should be routed to the correct nodeid/user. The command/message must be in NMR format.

The format of the macro is as follows:

```
  &name   IPW$GMS  TYPE=DIST,NMR=(Rx)
                   TYPE=SUB
```

TYPE — DIST causes the link to the general message distribution service.

NMR — Rx is the address of the nodal message record that must be distributed.

TYPE — SUB causes a link to the general message service routine to do message substitution.

The register usage for TYPE=SUB is as follows:

R0 — message length - 1.

R1 — address of message.

R4 — address of message control table.

R5 — TCB address to be used for message modification.

a user on another node, to the central operator, or to an ICCF user. &P The format of the macro is as follows:

```
  &name   IPW$NTY  MSG=(Rx)
                      =$mmmmm
                   NODE=(Ry)
                   USER=(Rz)
```

MSG — Rx contains the message id index
$mmmmm is the actual message number as obtained from the message definition module prefixed with $.

NODE — Ry contains the address of an eight byte field containing the target node name. If not specified the local node is assumed to be the required destination.

USER — Rz contains the address of an eight byte field containing the target userid, either a remote id in the form 'Rnnn' or an ICCF userid.

If not specified, the message is routed to the local console of the specified node.

IPW$RLV - Release Virtual Storage

This macro is used to release storage acquired with the IPW$RSV macro. The storage is returned to the appropriate pool and is available for other tasks. All tasks waiting for storage will again be posted in an attempt to satisfy the requirements.

The format of the macro is as follows:

```
  &name    IPW$RLV  <ADDR=(Rx)>
                    <,ALL>
                    <,OWNER=(Ry)>
                    <,ANCHOR=(Rz)>
```

ADDR — is the register which holds the address of the work space that should be released. The length of the buffer is found from the buffer prefix. It is not required if ALL is specified.

ALL — means that all storage belonging to this task should be released.

OWNER — specifies the address in Ry of the TCB that is to be used as owner for this storage. If not specified the issuing task will be considered as owner. The 'head' pointer will be taken from the owners TCB. Must NOT be specified if ANCHOR is specified.

ANCHOR — will be used as the 'head' pointer for the chain.

Upon return R0, R1, R2, and R3 are destroyed.

APPENDIX D. FORMAT OF INTERNAL MACROS

IPW$BUF - PNET Buffer Management

This macro is used as linkage to the buffer service routine (IPW$$BS).

The IPW$BUF macro has the following format:

```
                        CNTRL
                        FREE
                        GET
                        HEADQ
  &name   IPW$BUF  TYPE=MSG
                        QUEUE
                        PURNR
                        PURNT
                        RELEASE
                   ,MODE=IN
                         OUT
                  <,REG=(R1)>
                  <,WAIT=NO>
                        =YES
```

REG — is optional but if specified means that the address of the buffer that is to be referenced is already loaded into R1 and the address in the NCB should not be used. If not specified then the address from the NCB will be used. This keyword is only valid for TYPE=QUEUE,MODE=OUT.

WAIT — YES specifies that the task wants to be placed in the wait state until a buffer is available.
NO means that control will be returned directly to the calling task, with or without a buffer being available.
YES is default.

Register usage is as follows:

R0 — function code, describing the type of function to be performed.

01 - GET - MODE=IN

02 - RELEASE - MODE=IN

03 - FREE - MODE=IN

04 - QUEUE - MODE=IN

05 - GET - MODE=OUT

the 'received queue' is empty and on entry register 1 contained zero, then a return is made to the user with register 1 containing zero to indicate that no buffer was available. If the 'received queue' was empty and register 1 was NOT zero, then the task is put into a wait for a BSC event.

Note: This function is only invoked by the Receiver.

3. QUEUE

causes the input buffer being received, addressed by R1, to be queued as last entry in the 'received input chain' for the task, checks the number of buffers in the queue against the maximum value specified in the MAXBUF parameter for the node, and if the maximum is reached set a suspend for this task.

Note: This function is invoked only by the line driver when an input buffer is received successfully.

4. RELEASE

causes all buffers up to the maximum specified by MAXBUF, to be freed from the 'free input chain' and to be returned to VSE/POWER storage pool. If the number of buffers in the 'free input chain' is less than MAXBUF then all are freed except for one.

Note: This function is only invoked by the Line Driver.

MODE=OUT

1. CNTRL

causes a small TP buffer used for a NJE control record to be reserved and its address to be placed in R1. The buffer is marked to be freed after successful transmission. If register 1 contains zero on entry and no space is available then return is made to the user with register 1 as zero. If register 1 contains non zero on entry then the task is put into wait until space becomes available.

2. FREE

causes the output buffer currently being sent, addressed by register 1, to be queued as first entry in the 'free output chain' of the related NCB task entry.

If one of the following conditions arises then the buffer is released and its storage returned to the VSE/POWER storage pool.

- If the 'release-after-sent' flag is set in the buffer header.
- If the buffer owner is the line driver.
- SIGNOFF is in process or a line error has occurred.

If the 'short-on-buffer' condition is set in the NCB task entry (NCBEST2 = NCBESOB) then the task must also be posted.

R2 destroyed

R3 destroyed

R4 destroyed if OWNER or ANCHOR was specified.

Note: Even if WAIT=YES was specified, R1 must be checked to see if storage became available. It may happen that the task is posted again because of some abnormal condition, or a termination condition, and will then return without having acquired the storage.

IPW$STM - Set Timer Interval

The format of the IPW$STM macro is as follows:

```
IPW$STM  TIME=ttt
              (Ry)
         <,ECB=(Rx)>
         <,TQE=(R1)>
```

Licensed Material - Property of IBM

- Register 1 is zero, then ALL buffers are removed from the queue.
- Register 1 non zero then only buffers belonging to this task are removed from the queue.

If the buffer has the 'release-after-sent' flag set then the buffer is returned to the VSE/POWER storage pool. All other buffers are queued into the 'free output queue'.

8. QUEUE

   causes the TP-output buffer, addressed by register 1, to be queued as the last entry in the 'to-be-sent chain'. If the buffer just queued is the first in the queue, then the line driver is posted.

   If a line error has been detected or a 'SIGNOFF' record has been received, then the buffer is added to the 'free output queue'.

9. RELEASE

   causes all TP-buffers in the 'free output chain' (anchored to this NCB task entry), to be released and returned to the VSE/POWER storage pool.

IPW$GMS - General Message Service

The macro IPW$GMS must be used to invoke the general service function for all messages and commands that should be routed to the correct nodeid/user. The command/message must be in NMR format.

The format of the macro is as follows:

```
 &name   IPW$GMS  TYPE=DIST,NMR=(Rx)
                  TYPE=SUB
```

TYPE DIST causes the link to the general message distribution service.

LIST OF ABBREVIATIONS

AARA	Auxiliary account record area
ACB	Account control block
ACB	Access method control block
AQ	Add to queue
AQRA	Auxiliary queue record area
ASAB	Asynchronous service anchor block
BCA	Buffer control area
BCW	Buffer control word
CAT	Control address table
CB	Control block
CCB	Command control block
CI	Control interval for FBA
CIB	Communicator information block
CIDF	CI description field
CP	Command processor
CPB	Command processor control block
DLRSA	Double linkage register save area
DMB	Disk management block
DQ	Delete from queue
DSHR	Data Set Header Record
DRW	Disk request word
EAR	Execution account record
ECB	Event control block
ER	3540 Diskette reader
ETX	End of text
FBA	Fixed block architecture
FCB	Forms control buffer
FCS	Function Control Byte
FQ	Free queue
GD	Get data record
GNB	Generation table
IB	SNA inbound processor
IC	Pass command routine
IR	Internal reader
INIT/TERM	Initiator/Terminator
IP	Initialization processor
JECL	Job entry control language
JHR	Job Header Record
JTR	Job Trailer Record
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

GLOSSARY

Following is a definition of some of the terminology used in this manual.

Adjacent Node. Adjacent Nodes are any nodes which are directly connected with one another by a BSC connection or an SDLC session.

Alternate Route. It is possible to define for any destination an alternate path that may be used in the case that the main path, defined by the ROUTE1 parameter in the PNODE macro, is not available. This second route is called an alternate route. It may be specified by the ROUTE2 parameter in the PNODE generation when defining the destination node. It does not need to be a path using the same line discipline. It is NOT used as a 'load levelling' mechanism.

Command Switching. Command switching is the transmission of a command which has been received from the network to the next destination on its way to its final destination.

Compression. Compressing a data stream replaces two or more consecutive blanks by a one byte control character containing information as to how many blanks have been compressed. If three or more consecutive like non-blank characters are found then they are replaced by two bytes, the first control byte containing information as to how many occurences of the character have been compressed, and the second byte containing the actual character.

Compression is very widely used to reduce the size of the data stream before transmitting it via a tele-processing system.

Decompression. Decompression takes a compressed data string and expands it again to its original size by replacing the compression control bytes by the specified number of characters or blanks.

Direct Link. A Direct Link is defined as a connection between two adjacent nodes which is physically accomplished by a BSC line between the two nodes.

End Node. The End Node refers to that node which is designated as the final destination for the job or output. This can be the Local Node or any other node that is reachable within the network.

Execution Node. The execution node is the end node for job's. It is that node on which the job will be executed. It may be another VSE/POWER node or any other node supporting the networking protocol used by VSE/POWER PNET, and reachable from the local node.

Final Destination. Same as End Node.

that a part of the logical record will be put into the first block and the remainder of it will be placed at the start of the next block. Spanning may take place over any number of physical data records. To recover the logical record more than one physical record will have to be read.

Session. A Session can exist between any two nodes in the network. A session is established by use of ACF/VTAM or ACF/VTAME.

Store-and-Forward Node. Same as an intermediate node.

Topology Record. Topology records are written by JES2 NJE to dynamically describe the network. Records are sent whenever a node is started or stopped. These records are ignored by VSE/POWER.

Userid. The Userid is a 1-8 byte alpha-numeric identifier which may be used to identify the user who has submitted, or is to receive, the job or output.

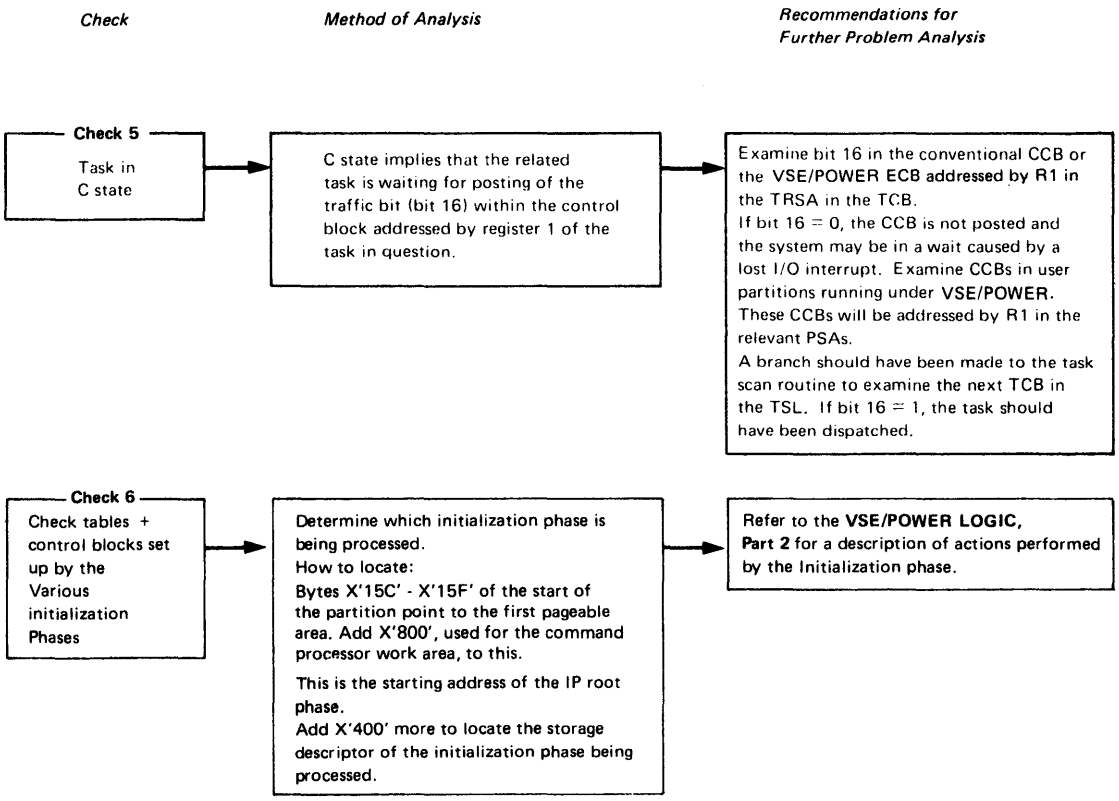


Figure 6-13 (Part 2 of 2). VSE/POWER Not Initialized

APPENDIX A. VSE/POWER STATUS BYTES IN THE VSE/ADVANCED FUNCTIONS SUPERVISOR

SYSCOM*

Location X'5C' - X'5F' contains the address of the VSE/POWER partition (label IPW$PDA in phase IPW$$NU), if VSE/POWER is initiated.

Location X'42' contains a flag byte:

X'08' = VSE/POWER supported
X'04' = VSE/POWER initialized

Partition COMREGS*

Location X'A0' - X'A3' contains the address of the partition control block (0 if no CB exists for this partition).

VSE/POWER FLAG BYTES

Location X'A4' contains VSE/POWER flags:

X'80' = VSE/POWER accounting support
X'40' = This partition under control of VSE/POWER
X'20' = This partition is the VSE/POWER partition
X'08' = Partition-waiting-for-work state

Location X'A5' reserved

* Refer to VSE/Advanced Functions Diagnosis: Service Aids, SC33-6099, for a full description and locations of SYSCOM and the partition COMREGs.

APPENDIX B. SUMMARY OF ECB USAGE

ECB usage is summarized in Figure B-1

ECB in:	Posted by : (Phase)	Unposted by:	Use when posted :
ACB	IPW$$GA/IPW$$SF	IPW$$PA/PF	account file is empty
CAT	Appendage	Task select.	Indicates work-to-do. for VSE/POWER
SCB	IPW$RLW	IPW$RSW	Work space is avail.
VSCB	IPW$RLV	IPW$RSV	Virtual storage avail.
DMB	IPW$$FQ	IPW$$RQ, IPW$$PD	Queue space is avail.
TCB (CP)	IPW$$I7	IPW$$CM	Indicates that IPW$$I7 has sent information to IPW$$CM
TCB (LD)	all PNET tasks IPW$$AQ, IPW$$MS, IPW$$CPS, IPW$$CP	IPW$$LD	If work is to do for PNET Driver
TCB (LMGR)	• Channel End Appendage • line start • line stop	IPW$$LM	work-to-do for line manager.
TCB (OB)	IPW$$IB, IPW$$MP, IPW$$SN	IPW$$OB	Indicates that trans. to SNA terminal which was previously suspended is to continue.
TCB (SN)	ACF/VTAM at completion of a RECEIVE ANY	IPW$$SN	Indicates that IPW$$SN must attach IPW$$IB
TCB (SN)	IPW$$SN, IPW$$LN, IPW$$IB, IPW$$OB, IPW$$MP, IPW$$LF, IPW$$VE, IPW$$LH, IPW$$MS	IPW$$SN	Indicates work-to-do for IPW$$SN
SRB	IPW$$AS (Subtask)		Indicates that service request is processed
ASAB	IPW$$AS	IPW$$AS (Subtask)	Indicates that service request is waiting to be processed.

Figure B-1. Summary of ECB Usage

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.

APPENDIX C. VSE/POWER STATUS REPORT

This report is printed on two occasions:

1. When VSE/POWER is initiated, when SYSLST is assigned to a line printer in the VSE/POWER partition.

2. When VSE/POWER is terminated with a PEND command that includes a physical address of a line printer.

The format for C-K-D devices is shown below. The format for FBA devices is similar but definitions shown below for tracks are for FBA blocks.

```
QUEUE FILE   IJQFILE

TOTAL NUMBER OF QUEUE RECORDS                                  xxxx RECORDS
NUMBER OF FREE QUEUE RECORDS                                   xxxx RECORDS
MAXIMUM NUMBER OF QUEUE RECORDS USED IN PRESENT SESSION        xxxx RECORDS

DATA FILE   IJDFILE

TOTAL NUMBER OF TRACKS                                         xxxx TRACKS
TRACK GROUP SIZE                                                 xx TRACKS
DATA BLOCK SIZE                                                  xx BYTES

ACCOUNT FILE   IJAFILE

TOTAL NUMBER OF TRACKS                                         xxxx TRACKS
PERCENTAGE OF FILE THAT IS FILLED                                    xx %

FIXABLE STORAGE ALLOCATED TO VSE/POWER PARTITION                 xx PAGES
NUMBER OF TIMES TASKS WERE WAITING FOR PFIXED STORAGE           xxx TIMES
MAXIMUM NUMBER OF PAGES FIXED                                    xx PAGES
MAXIMUM NUMBER OF TASKS ACTIVE AT ONE POINT IN TIME              xx TASKS
NUMBER OF TIMES TASKS WERE WAITING FOR VIRTUAL STORAGE          xxx TIMES

TIME INTERVALS FOR SHARED SPOOLING (SYSID=n) :
T1 =   x SECONDS, T2 =   x SECONDS, T3 =    x SECONDS, T4 =   x SECONDS

NUMBER OF NOTIFY MESSAGES LOST:                              xxxxx MESSAGE(S)
```

Figure C-1. Status Report

a user on another node, to the central operator, or to an ICCF user. &P The format of the macro is as follows:

```
&name  IPW$NTY  MSG=(Rx)
                   =$mmmmm
                NODE=(Ry)
                USER=(Rz)
```

MSG — Rx contains the message id index
$mmmmm is the actual message number as obtained from the message definition module prefixed with $.

NODE — Ry contains the address of an eight byte field containing the target node name. If not specified the local node is assumed to be the required destination.

USER — Rz contains the address of an eight byte field containing the target userid, either a remote id in the form 'Rnnn' or an ICCF userid.

If not specified, the message is routed to the local console of the specified node.

IPW$RLV - Release Virtual Storage

This macro is used to release storage acquired with the IPW$RSV macro. The storage is returned to the appropriate pool and is available for other tasks. All tasks waiting for storage will again be posted in an attempt to satisfy the requirements.

The format of the macro is as follows:

```
&name   IPW$RLV  <ADDR=(Rx)>
                 <,ALL>
                 <,OWNER=(Ry)>
                 <,ANCHOR=(Rz)>
```

ADDR — is the register which holds the address of the work space that should be released. The length of the buffer is found from the buffer prefix. It is not required if ALL is specified.

ALL — means that all storage belonging to this task should be released.

OWNER — specifies the address in Ry of the TCB that is to be used as owner for this storage. If not specified the issuing task will be considered as owner. The 'head' pointer will be taken from the owners TCB. Must NOT be specified if ANCHOR is specified.

ANCHOR — will be used as the 'head' pointer for the chain.

Upon return R0, R1, R2, and R3 are destroyed.

APPENDIX D. FORMAT OF INTERNAL MACROS

IPW$BUF - PNET Buffer Management

This macro is used as linkage to the buffer service routine (IPW$$BS).

The IPW$BUF macro has the following format:

```
                        CNTRL
                        FREE
                        GET
                        HEADQ
  &name  IPW$BUF  TYPE=MSG
                        QUEUE
                        PURNR
                        PURNT
                        RELEASE
                   ,MODE=IN
                         OUT
                  <,REG=(R1)>
                  <,WAIT=NO>
                        =YES
```

REG — is optional but if specified means that the address of the buffer that is to be referenced is already loaded into R1 and the address in the NCB should not be used. If not specified then the address from the NCB will be used. This keyword is only valid for TYPE=QUEUE,MODE=OUT.

WAIT — YES specifies that the task wants to be placed in the wait state until a buffer is available.
NO means that control will be returned directly to the calling task, with or without a buffer being available.
YES is default.

Register usage is as follows:

R0 — function code, describing the type of function to be performed.

01 - GET - MODE=IN

02 - RELEASE - MODE=IN

03 - FREE - MODE=IN

04 - QUEUE - MODE=IN

05 - GET - MODE=OUT

the 'received queue' is empty and on entry register 1 contained zero, then a return is made to the user with register 1 containing zero to indicate that no buffer was available. If the 'received queue' was empty and register 1 was NOT zero, then the task is put into a wait for a BSC event.

Note: This function is only invoked by the Receiver.

3. QUEUE

causes the input buffer being received, addressed by R1, to be queued as last entry in the 'received input chain' for the task, checks the number of buffers in the queue against the maximum value specified in the MAXBUF parameter for the node, and if the maximum is reached set a suspend for this task.

Note: This function is invoked only by the line driver when an input buffer is received successfully.

4. RELEASE

causes all buffers up to the maximum specified by MAXBUF, to be freed from the 'free input chain' and to be returned to VSE/POWER storage pool. If the number of buffers in the 'free input chain' is less than MAXBUF then all are freed except for one.

Note: This function is only invoked by the Line Driver.

MODE=OUT

1. CNTRL

causes a small TP buffer used for a NJE control record to be reserved and its address to be placed in R1. The buffer is marked to be freed after successful transmission. If register 1 contains zero on entry and no space is available then return is made to the user with register 1 as zero. If register 1 contains non zero on entry then the task is put into wait until space becomes available.

2. FREE

causes the output buffer currently being sent, addressed by register 1, to be queued as first entry in the 'free output chain' of the related NCB task entry.

If one of the following conditions arises then the buffer is released and its storage returned to the VSE/POWER storage pool.

- If the 'release-after-sent' flag is set in the buffer header.
- If the buffer owner is the line driver.
- SIGNOFF is in process or a line error has occurred.

If the 'short-on-buffer' condition is set in the NCB task entry (NCBEST2 = NCBESOB) then the task must also be posted.

- Register 1 is zero, then ALL buffers are removed from the queue.
- Register 1 non zero then only buffers belonging to this task are removed from the queue.

If the buffer has the 'release-after-sent' flag set then the buffer is returned to the VSE/POWER storage pool. All other buffers are queued into the 'free output queue'.

8. QUEUE

   causes the TP-output buffer, addressed by register 1, to be queued as the last entry in the 'to-be-sent chain'. If the buffer just queued is the first in the queue, then the line driver is posted.

   If a line error has been detected or a 'SIGNOFF' record has been received, then the buffer is added to the 'free output queue'.

9. RELEASE

   causes all TP-buffers in the 'free output chain' (anchored to this NCB task entry), to be released and returned to the VSE/POWER storage pool.

IPW$GMS - General Message Service

The macro IPW$GMS must be used to invoke the general service function for all messages and commands that should be routed to the correct nodeid/user. The command/message must be in NMR format.

The format of the macro is as follows:

```
&name   IPW$GMS  TYPE=DIST,NMR=(Rx)
                 TYPE=SUB
```

TYPE — DIST causes the link to the general message distribution service.

NMR — Rx is the address of the nodal message record that must be distributed.

TYPE — SUB causes a link to the general message service routine to do message substitution.

The register usage for TYPE=SUB is as follows:

R0 — message length - 1.

R1 — address of message.

R4 — address of message control table.

R5 — TCB address to be used for message modification.

R2 destroyed

R3 destroyed

R4 destroyed if OWNER or ANCHOR was specified.

Note: Even if WAIT=YES was specified, R1 must be checked to see if storage became available. It may happen that the task is posted again because of some abnormal condition, or a termination condition, and will then return without having acquired the storage.

IPW$STM - Set Timer Interval

The format of the IPW$STM macro is as follows:

```
IPW$STM  TIME=ttt
              (Ry)
         <,ECB=(Rx)>
         <,TQE=(R1)>
         <,WAIT=YES>
                NO
```

There is also a second format that can be used:

```
IPW$STM  CANCEL=YES,TQE=(R1)
```

ECB is the ECB which is posted when the time interval expires.

TIME ttt is the time interval in tenths of a second. The time may be specified as register notation (Ry).

TQE addresses a previously acquired TQE. If TQE= is not specified storage for the TQE is reserved and register 1 is used as pointer register.

WAIT YES means that the macro expands into a wait and the TQE storage is automatically released.
NO means that no wait is automatically generated
YES is default.

CANCEL YES can be used to delete a time interval which has been already setup. TQE= must also be specified and R1 must contain the address of the TQE that should be deleted

Note: Upon return registers R0, R1, R2 and R3 are destroyed.

IPW$UNV - Unchain Virtual Storage Element

This macro is used to unchain a specific element of a specified queue and to chain it to another queue. The address of the unchained element is returned in register 1. The element may be directly addressed or if no address is given then the first element of the queue is unchained.

LIST OF ABBREVIATIONS

AARA	Auxiliary account record area
ACB	Account control block
ACB	Access method control block
AQ	Add to queue
AQRA	Auxiliary queue record area
ASAB	Asynchronous service anchor block
BCA	Buffer control area
BCW	Buffer control word
CAT	Control address table
CB	Control block
CCB	Command control block
CI	Control interval for FBA
CIB	Communicator information block
CIDF	CI description field
CP	Command processor
CPB	Command processor control block
DLRSA	Double linkage register save area
DMB	Disk management block
DQ	Delete from queue
DSHR	Data Set Header Record
DRW	Disk request word
EAR	Execution account record
ECB	Event control block
ER	3540 Diskette reader
ETX	End of text
FBA	Fixed block architecture
FCB	Forms control buffer
FCS	Function Control Byte
FQ	Free queue
GD	Get data record
GNB	Generation table
IB	SNA inbound processor
IC	Pass command routine
IR	Internal reader
INIT/TERM	Initiator/Terminator
IP	Initialization processor
JECL	Job entry control language
JHR	Job Header Record
JTR	Job Trailer Record
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

SA	Save account
SAM	Sequential access method
SC	Scan reader JECL statement
SCB	Storage control block or String Control Byte
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
SRCB	String Record Control Byte
SUCB	SNA unit control block
TBB	Tape control block
TCB	Task control block
TMF	Task management field
TMS	Task management service
TR	Task terminator
TRSA	Task register save area
TSL	Task selection list
VSCB	Virtual storage control block
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
XR	Execution reader
XW	Execution writer

BIBLIOGRAPHY

To use this manual effectively, you should be familiar with the concepts and facilities of VSE/Advanced Functions described in the following manuals:

VSE/Advanced Functions:

- Installation, SC33-6096
- System Management Guide, SC33-6094
- Operating Procedures, SC33-6097
- System Control Statements, SC33-6095

RJE,SNA users should also be familiar with ACF/VTAM concepts and facilities as described in:

- ACF/VTAM Concepts and Planning, GC38-0282
- ACF/VTAM Macro Language Reference, SC38-0261

Other VSE/POWER publications are:

- VSE/POWER Installation and Operations Guide, SH12-5329
- VSE/POWER Remote Job Entry User's Guide, SH12-5328
- VSE/POWER Networking User's Guide, SC33-6140
- VSE/POWER Messages, SH12-5520
- VSE/POWER Reference Summary, SH12-5435
- VSE/POWER Shared Spooling User's Guide, SH12-5330

GLOSSARY

Following is a definition of some of the terminology used in this manual.

Adjacent Node. Adjacent Nodes are any nodes which are directly connected with one another by a BSC connection or an SDLC session.

Alternate Route. It is possible to define for any destination an alternate path that may be used in the case that the main path, defined by the ROUTE1 parameter in the PNODE macro, is not available. This second route is called an alternate route. It may be specified by the ROUTE2 parameter in the PNODE generation when defining the destination node. It does not need to be a path using the same line discipline. It is NOT used as a 'load levelling' mechanism.

Command Switching. Command switching is the transmission of a command which has been received from the network to the next destination on its way to its final destination.

Compression. Compressing a data stream replaces two or more consecutive blanks by a one byte control character containing information as to how many blanks have been compressed. If three or more consecutive like non-blank characters are found then they are replaced by two bytes, the first control byte containing information as to how many occurences of the character have been compressed, and the second byte containing the actual character.

Compression is very widely used to reduce the size of the data stream before transmitting it via a tele-processing system.

Decompression. Decompression takes a compressed data string and expands it again to its original size by replacing the compression control bytes by the specified number of characters or blanks.

Direct Link. A Direct Link is defined as a connection between two adjacent nodes which is physically accomplished by a BSC line between the two nodes.

End Node. The End Node refers to that node which is designated as the final destination for the job or output. This can be the Local Node or any other node that is reachable within the network.

Execution Node. The execution node is the end node for job's. It is that node on which the job will be executed. It may be another VSE/POWER node or any other node supporting the networking protocol used by VSE/POWER PNET, and reachable from the local node.

Final Destination. Same as End Node.

INDEX

A

B

D

M

N

S

V

W

3

READER'S
COMMENT
FORM

VSE/POWER
Program Logic Manual Part 1
Order No. LY12-5027-3

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

READER'S
COMMENT
FORM

VSE/POWER
Program Logic Manual Part 1
Order No. LY12-5027-3

This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name and mailing address:

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp is necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.