

IBM z/VSE  
VSE Central Functions



# VSE/POWER Networking

*Version 7 Release 1*



IBM z/VSE  
VSE Central Functions



# VSE/POWER Networking

*Version 7 Release 1*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page ix.

**Second Edition (March 2005)**

This edition applies to Version 7 Release 1 of IBM VSE/POWER, Program Number 5686-CF7, which is part of VSE Central Functions, Program Number 5609-ZVS, and to all subsequent releases and modifications until otherwise indicated in new editions.

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

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

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

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

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

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

© Copyright International Business Machines Corporation 1981, 2005. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Figures . . . . .</b>	<b>v</b>
--------------------------	----------

<b>Tables . . . . .</b>	<b>vii</b>
-------------------------	------------

<b>Notices . . . . .</b>	<b>ix</b>
--------------------------	-----------

Programming Interface Information . . . . .	ix
Trademarks and Service Marks . . . . .	x

<b>About This Book. . . . .</b>	<b>xi</b>
---------------------------------	-----------

Who Should Use This Book . . . . .	xi
How to Use This Book . . . . .	xi
Where to Find More Information . . . . .	xii
Abbreviations . . . . .	xiii

<b>Summary of Changes . . . . .</b>	<b>xv</b>
-------------------------------------	-----------

Changes to this Manual for VSE/POWER 6.7 and 7.1 . . . . .	xv
General Functional Extension VSE/POWER 6.7 . . . . .	xv
Additional Improvements Implemented by Field APARs . . . . .	xv
Changes to this Manual for VSE/POWER 6.6. . . . .	xvi

<b>Chapter 1. Getting Started . . . . .</b>	<b>1</b>
---	----------

Introduction to VSE/POWER Networking . . . . .	1
Overview of Data Flow Through a Network. . . . .	2
Connecting the Network . . . . .	5
CTCA Support. . . . .	6
Establishing PNET Support . . . . .	6
VSE/POWER Generation Considerations for PNET. . . . .	6
CTCA Requirements . . . . .	10
VTAM Requirements . . . . .	11
TCP/IP Requirements . . . . .	11
PNET Initialization . . . . .	11
Loading the PNET System . . . . .	11
Network Definition Table (NDT) . . . . .	12
Starting a PNET Session . . . . .	15
Providing Security Access to Nodes and Lines. . . . .	18

<b>Chapter 2. Using PNET. . . . .</b>	<b>21</b>
---------------------------------------	-----------

PNET Processing . . . . .	21
Preparing a Job for Transmission . . . . .	21
Disposition in the Transmit Queue. . . . .	21
Priority in the Transmit Queue . . . . .	23
Displaying Status Information . . . . .	23
Job Number Assignment . . . . .	24
Activating Transmitters and Receivers . . . . .	24
Draining Transmitters and Receivers . . . . .	25
Path Selection . . . . .	25
Shutting Down the Network. . . . .	26
Notify Support . . . . .	27
Accounting . . . . .	28
PNET Receiver Exit. . . . .	29
PNET Transmitter Exit. . . . .	33

Command Authorization . . . . .	38
Providing Data Security . . . . .	42
Automatic Restart of Nodes . . . . .	43
RJE with PNET . . . . .	44
Administrative Considerations . . . . .	44
Network Considerations . . . . .	44
Changing the Name of the Local Node During Warmstart . . . . .	45
VTAM Considerations . . . . .	49
TCP/IP Considerations . . . . .	50
Performance Considerations . . . . .	51
Network Operation with Shared Spooling . . . . .	54
Change of a Job's Characteristics . . . . .	54
Connections to a Backlevel VSE/POWER System . . . . .	58
Connections to RSCS . . . . .	59
Connections to JES2 NJE . . . . .	59
Connections to JES3 NJE . . . . .	60

<b>Chapter 3. Diagnostic Aids . . . . .</b>	<b>63</b>
---	-----------

PNET Trace in Telecommunication Trace Area . . . . .	63
PNET Console Trace . . . . .	63
PNET TCP and PNET SSL System Messages . . . . .	63
Problem Isolation in VSE/POWER Networking . . . . .	64
Common Problems . . . . .	64
TCP Node Cannot Sign On . . . . .	68
SSL Node Cannot Sign On . . . . .	72

<b>Appendix A. Examples of Defining VTAM BOOKs and Pacing . . . . .</b>	<b>81</b>
---	-----------

Defining B-BOOKs . . . . .	81
Node B B-BOOK . . . . .	81
Node D B-BOOK . . . . .	82
Specifying Pacing . . . . .	83

<b>Appendix B. Sample of a PNET Receiver Exit Routine . . . . .</b>	<b>85</b>
---	-----------

Control Statements for Assembly and Cataloging. . . . .	85
Programming Example Source Code . . . . .	85

<b>Appendix C. Sample of a PNET Transmitter Exit Routine . . . . .</b>	<b>95</b>
--	-----------

Control Statements for Assembly and Cataloging. . . . .	95
Programming Example Source Code . . . . .	95

<b>Appendix D. PNET BSC/CTCA/TCP/SSL Console Trace . . . . .</b>	<b>107</b>
--	------------

Console Trace Commands . . . . .	108
Additional TCP Console Trace Message . . . . .	108
Numeric Step Identifier . . . . .	109
Alphabetic Step Identifier . . . . .	111
Routing TCP Console Trace Messages to SYSLST . . . . .	114
Additional TCP Component Console Trace Messages . . . . .	115

<b>Appendix E. PNET Telecommunication Trace</b>	<b>117</b>
Format of the PNET BSC/CTCA Trace Record	118
Format of the PNET SNA Trace Record	119
Format of the PNET TCP and PNET SSL Trace Records	120

<b>Appendix F. Examples of Defining TCP/IP Connections</b>	<b>123</b>
Defining the TCP/IP for VSE/ESA Configuration File IPINITnn.L.	123

Defining the VSE/POWER Network Definition Table	124
Starting the TCP/IP Support of VSE/POWER	125
Stopping the TCP/IP Support of VSE/POWER	126
Multihoming Support (More than One Ip-Address)	126

<b>Glossary</b>	<b>129</b>
-----------------	------------

<b>Index</b>	<b>135</b>
--------------	------------

---

## Figures

1.	Data Flow Through a Network - Concepts	3	7.	Authorization Table for Remote (RJE) Users, Spool-Access CTL Users, or Cross-Partition CTLSPPOOL Users	41
2.	Sample Network	5	8.	Example of VTAM Definition for Pacing	83
3.	Example of an IBM VSE/POWER Generation including PNET/BSC Lines	8	9.	Example of Log-on Mode Table (MODETAB) Definition	84
4.	PNODE Generations for the Sample Network	10			
5.	Transmitter Combinations Not Allowed	24			
6.	Authorization Table for Central Operator of Remote Node or (Master-Password) Authorized Spool-Access CTL User of the Remote Node	39			





---

## Tables

1. Parameter List of the Transmitter Exit . . . . 34
2. Fields of the Parameter List Set by  
VSE/POWER and the User . . . . . 35



---

## Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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

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

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

IBM Deutschland GmbH  
Department 0790  
Pascalstr. 100  
70569 Stuttgart  
Germany

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

---

## Programming Interface Information

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

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

————— **Programming Interface Information** —————

————— **End of Programming Interface Information** —————

---

## Trademarks and Service Marks

The following terms used in this publication are trademarks or service marks of the IBM Corporation in the United States and/or other countries:

AS/400  
CICS  
CICS/VSE  
IBM  
Language Environment  
MVS  
NetView  
OS/390  
OS/400  
VM/ESA  
VSE/ESA  
VTAM  
z/OS  
z/VSE  
z/VM

---

## About This Book

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

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

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

This manual describes the networking support provided by VSE/POWER and is intended to help the reader to get familiar with the networking facilities VSE/POWER provides.

---

## Who Should Use This Book

This manual addresses users who exchange jobs, output, commands, and messages between different processors in a network. Such a network consists of spooling systems supporting the 'Network Job Interface (NJI)'.

---

## How to Use This Book

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

- Chapter 1, "Getting Started," on page 1 gives an overview of PNET, shows how to generate PNET, summarizes its requirements, discusses initialization of PNET, shows how to start sessions between nodes, and describes the use of passwords.
- Chapter 2, "Using PNET," on page 21 shows everyday communication with PNET and describes general administrative concepts.
- Chapter 3, "Diagnostic Aids," on page 63 discusses diagnostic tools and tells you how to find and solve commonly occurring problems.

The following information is included in the appendixes:

- Appendix A, "Examples of Defining VTAM BOOKs and Pacing," on page 81 gives examples of a VTAM generation and of specifying pacing.
- Appendix B, "Sample of a PNET Receiver Exit Routine," on page 85 provides a sample of a user receiver exit routine for checking information arriving at VSE/POWER via the network.
- Appendix C, "Sample of a PNET Transmitter Exit Routine," on page 95 provides a sample of a user transmitter exit routine for checking information that leaves VSE/POWER via the network.
- Appendix D, "PNET BSC/CTCA/TCP/SSL Console Trace," on page 107 describes the PNET console trace, which is used to record I/O between the nodes using a BSC, CTCA, or TCP line.
- Appendix E, "PNET Telecommunication Trace," on page 117 gives information about the PNET telecommunication trace.
- Appendix F, "Examples of Defining TCP/IP Connections," on page 123 gives sample TCP/IP connection definitions.

Additional help is provided at the back of the book:

- The glossary explains technical terms.
- The index helps you to locate information.

---

## Where to Find More Information

The following IBM manuals contain information pertinent to VSE/POWER:

- *VSE/POWER Administration and Operation*, SC33-8247
- *VSE/POWER Remote Job Entry*, SC33-6734
- *VSE/POWER Application Programming*, SC33-8248

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

For z/VSE, you may occasionally consult the following IBM manuals:

- *z/VSE Installation*, SC33-8222
- *z/VSE Operation*, SC33-8239
- *z/VSE Networking Support*, SC33-8235
- *z/VSE Guide to System Functions*, SC33-8233
- *z/VSE System Control Statements*, SC33-8225

For network planning, you may also need the following IBM manuals:

- *Planning for NetView, NCP, and VTAM*, SC31-7122
- *VM RSCS Networking V3R2 Planning and Installation*, SH24-5219
- *Network Job Entry Formats and Protocols*, SC23-0070.

For details on connecting NJE products, see

- *VSE/POWER 5.1 Networking Functions*, ITSC, GG24-3571.
- *VSE/POWER and OS/400 NJE Configuration Guide*, ITSC, GG24-4259.

These publications of the International Technical Support Organization (ITSO) are examples of *redbooks*. More information about redbooks can be found at the Web site

<http://www.redbooks.ibm.com>

For VM command reference, please refer to *z/VM V4R1.0 CP Command and Utility Reference*, SC24-6008.

For more information on TCP/IP, see

- IBM Redbook *Getting Started with TCP/IP for VSE/ESA 1.4*, SG24-5626.
- *TCP/IP for VSE/ESA IBM Program Setup and Supplementary Information*, SC33-6601.
- *z/VSE e-business Connectors User's Guide*, SC33-8231.

### **z/VSE Home Page**

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

You can find the z/VSE home page at:

<http://www.ibm.com/servers/eserver/zseries/zvse/>

### VSE/POWER Home Page

VSE/POWER also has a home page that offers current information on VSE/POWER.

You can find the VSE/POWER home page at:

<http://www.ibm.com/servers/eserver/zseries/os/vse/support/power/power.htm>

## Abbreviations

BSC	=	binary synchronous communication (type of node connection)
BSD	=	Berkeley Software Distribution
CDRM	=	cross-domain resource manager
CICS/VSE	=	Customer Information Control System/VSE
CTCA	=	channel-to-channel adapter (type of node connection)
cuu	=	channel-unit-unit (device-address format)
DSHR	=	Data Set Header Record
EOJ	=	end of job
FCB	=	forms control buffer
FIFO	=	first-in-first-out
ICA	=	integrated communications adapter
ICCF	=	Interactive Computing and Control Facility
IPL	=	initial program load (of operating systems)
ID	=	identifier/identification
ITSC	=	International Technical Support Center
ITSO	=	International Technical Support Organization
JCL	=	job control language
JECL	=	job entry control language
JES	=	Job Entry System (the OS/390 spooling/networking component)
JHR	=	Job Header Record
JTR	=	Job Trailer Record
KB	=	Kilobyte (= 1024 bytes)
ML	=	Multi-leaving
NAK	=	Negative Acknowledgement (BSC term)
NAT	=	node attached table
NCP	=	Network Control Program
NDT	=	network definition table
NJE	=	network job entry
NJI	=	network job interface
OS/390	=	IBM mainframe operating system
OSA	=	open systems adapter
PIU	=	path information unit
PNET	=	Power NETWORKing
RCB	=	record control byte
RJE	=	remote job entry
RSCS	=	Remote Spooling Communication Subsystem (the VM spooling/networking component)
SDLC	=	synchronous data link control (type of node connection)
SNA	=	system network architecture
SRCB	=	subrecord control byte
SSL	=	1. Secure Sockets Layer 2. Type of node connection using TCP/IP together with the SSL feature.
TCP	=	Type of node connection using TCP/IP
TCP/IP	=	Transmission Control Protocol/Internet Protocol
VCTC	=	virtual channel to channel (VM function)
VIO	=	virtual storage space (used for queue file copy)
VM	=	Virtual Machine (a type of IBM operating systems)
VSE	=	Virtual Storage Extended (a type of IBM operating system)
VSE/ESA	=	Virtual Storage Extended/Enterprise Systems Architecture

VTAM = Virtual Telecommunications Access Method  
z/OS = zSeries eServer operating system  
z/VSE = zSeries VSE  
z/VM = zSeries VM



---

## Summary of Changes

---

### Changes to this Manual for VSE/POWER 6.7 and 7.1

#### General Functional Extension VSE/POWER 6.7

##### **PNET Node Name Change during Warm Start**

The local PNET node name is not only contained in the spool entry queue record, but also in spool entry control records (job header and data set header records) within the spool data on the data file. Converting the spool entry node name previously could only be achieved by performing a cold start. This conversion process can now be requested during a VSE/POWER warm start, whenever a different local node name is introduced via the network definition table (NDT) as specified by the PNET= operand of the POWER macro generation, or by the SET PNET autostart override statement. After operator confirmation, the conversion process is started for all existing queue entries. "Work-in-progress" messages will accompany the processing as the VSE/POWER initialization is possibly delayed due to this new function, until finally the dynamic conversion of the spool files has completed. For details see "Changing the Name of the Local Node During Warmstart" on page 45

#### Additional Improvements Implemented by Field APARs

##### **Accelerate PNET TCP and SSL connections**

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

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

##### **Introduce TCP/IP Multihoming Support**

Multihoming support allows connection to another NJI system in a TCP/IP network to be addressed by more than one ip-address. This may be used, for example, for firewall support or supporting hardware boxes with unique ip-address.

Appendix F has been improved and extended to describe more details about the definitions concerning a TCP/IP connection. A paragraph about multihoming support has been added. For details see Appendix F, "Examples of Defining TCP/IP Connections," on page 123

---

## Changes to this Manual for VSE/POWER 6.6

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

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

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

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

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

**Note:** Existing NDTs must be re-assembled under VSE/POWER 6.6.

- The display of a Version 06.0 NDT has a new arrangement.
- The PSTART PNET command for an SSL node has the same format as used for TCP nodes. All other PSTOP, PDRAIN, PACT, PFLUSH, and PINQUIRE networking commands are independent of the node type and therefore transparent.
- The PINQUIRE command has been enhanced for
  - subselection via PNETSSL
  - new display of local node attributes
  - improved display of remote nodes
- The so-called PNET Console trace of BSC/CTC/TCP/SSL communications has been extended for
  - subselection of PNETSSL
  - comprehensive trace information for SSL nodes

**z/VSE Home Page**

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

You can find the z/VSE home page at:

<http://www.ibm.com/servers/eserver/zseries/zvse/>

**VSE/POWER Home Page**

VSE/POWER also has a home page that offers current information on VSE/POWER.

You can find the VSE/POWER home page at:

<http://www.ibm.com/s390/vse/support/power/power.htm>



---

# Chapter 1. Getting Started

---

## Introduction to VSE/POWER Networking

The networking function (PNET) is an optional part of VSE/POWER running with z/VSE. PNET is designed for use with multiple systems that are located at different places. Such a system can be:

- another z/VSE system running VSE/POWER
- a z/VM or VM/ESA system running RSCS
- a z/OS or OS/390 system running JES2 or JES3
- an AS/400 system

However, PNET may also be used in systems at the same location as an alternative to the VSE/POWER Shared Spooling function.

PNET provides for the transmission of selected jobs, operator commands, messages, and output data between communicating job entry nodes that are connected in a network. The nodes are connected by BSC lines, virtual CTCAs, VTAM-controlled SNA links, or TCP/IP connections.

A node of a job entry network may consist of a single z/VSE system or up to nine z/VSE systems forming a VSE/POWER shared spooling environment. You may specify the node in which a given job is run, the destination of the output, and the routing (among the nodes) for the jobs. System defaults apply if you do not specify a destination for the output.

A job may be entered into the network from any VSE/POWER controlled local input device, any remote terminal supported by VSE/POWER RJE, or through the spool-access support in a given node; it may be queued for transmission to any node in the network.

Job transmission is handled on a store-and-forward basis; that is, a job must be completely received by a node before any action is taken to run the job or transmit it to another node. Once a job has been completely received by a node, the transmitting node frees the resources that were allocated to the transmitted job. Jobs within a given priority are transmitted on a first-in-first-out basis.

When the job is transmitted to the node where it will run, it is placed in the queue to await its turn. Output files generated by the running of a job are spooled in the node where the job has run. If the output is to go to another node, it is transmitted either when an output segment is available or when the job has run.

Jobs and in-stream data destined for another node are queued for transmission at the node of entry. Jobs and output which pass through one or more nodes on the way to the final destination are spooled at every transit node.

Operator commands and messages may originate from any node within the network; they may be passed to any other node. Responses to VSE/POWER commands are routed back to the originator. The originator may be a console, a remote terminal, or a user in another partition. The VSE/POWER commands allow you to control jobs at any other VSE/POWER node in the network if these jobs were submitted by you. It is also possible to send a non-VSE/POWER command to a non-z/VSE system. In this case the command must be in a format which the node which should process the command understands.

### Overview of Data Flow Through a Network

A job may enter the network at one node, be routed to another node for execution, and directed to one or more nodes for output. Figure 1 on page 3 illustrates the processing flow and shows VSE/POWER components that participate in the transmission, routing, and receipt of jobs.

Jobs and associated in-stream data submitted to the job entry network from a local device or a remote terminal may run at the node of entry or any other node in the network. The node must have the proper configuration (for example, the necessary tape drives, direct access devices, and specific files). The node must be designated as the node where the job is to run.

If the path is lost or changed during the transmission of data, the subsystem that currently has the work:

- Holds the job, or
- Selects an alternate path (if possible) for transmission.

Commands are queued to the console transmitter for transmission. If the path to the final destination is lost, an attempt is made to return a diagnostic message to the originating console or user. Severe errors may result in commands being discarded without the possibility to issue a diagnostic message.

Messages flow through the network in a similar manner as commands. If, however, the path to the final destination is lost, the message is discarded. Commands and messages travel through the network with a higher priority than jobs or output data.

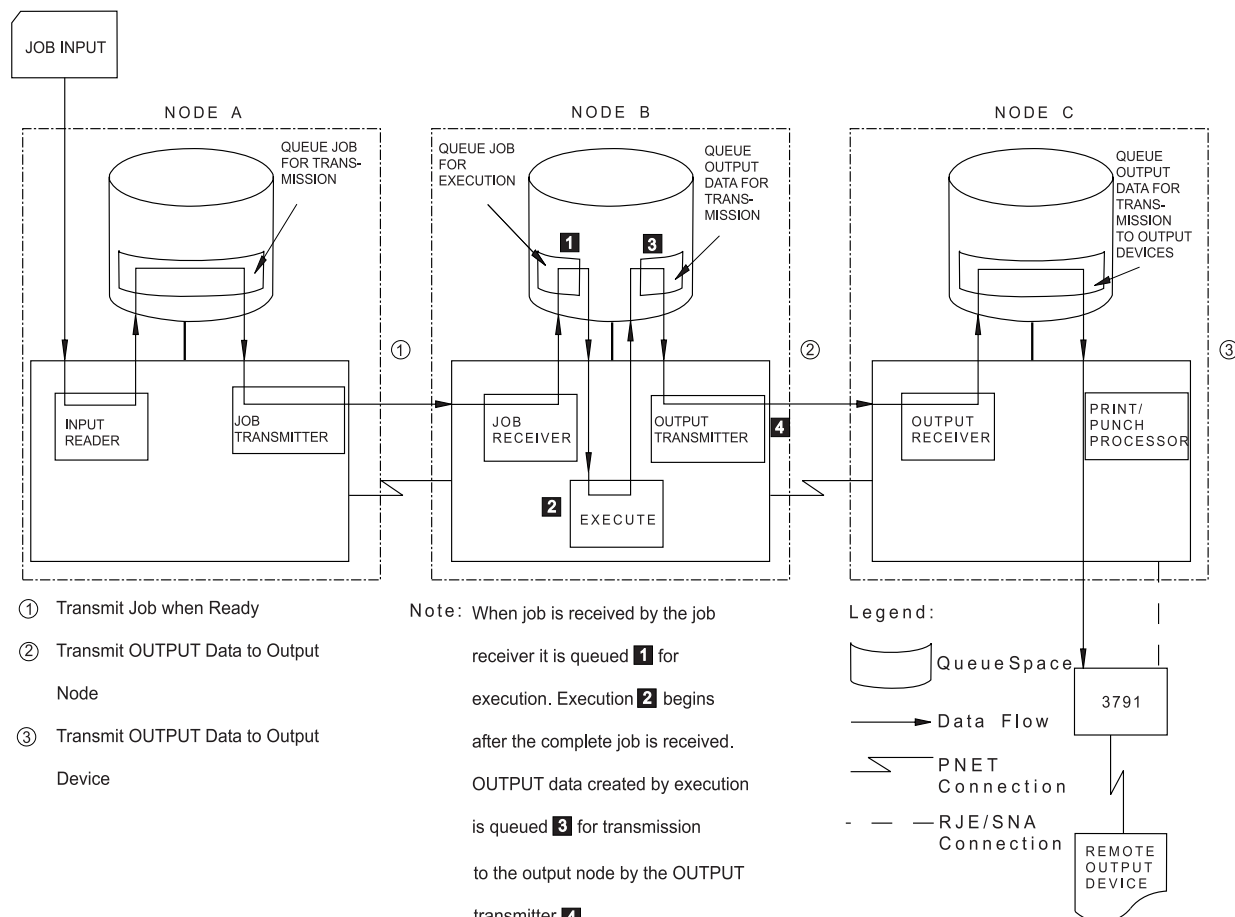


Figure 1. Data Flow Through a Network - Concepts

The execution node for a job is determined in the following ways:

- If nothing is specified, the node of entry is used.
- It may be defined by use of the \* \$\$ JOB statement.
- It may be modified at any time while the job is queued for running or transmission to another node, by an operator command. The execution node may be modified by the operator even if he is not the submitter of the job. Any operator at a remote node may also change the destination of the job provided he has the correct authority and ownership.

When a job reaches its final destination, it is queued for execution. Output files produced by the job are queued either for output or for transmission. This depends on whether the files are to be printed or punched at the executing node or at another node.

The destination node for output is determined as follows:

- The point of origin is used if not overwritten by other specification.
- By specifying the destination on a JECL statement.
- By use of VSE/POWER operator commands at the executing node or at any other node in the network with the correct authority level.

## Introduction

Output that is made available while the job is running by use of segmentation, is queued immediately for transmission, if it is destined for another node. Data that is to be held for remote node retrieval is transmitted to its final destination node and held there.

Figure 2 on page 5 illustrates a network consisting of nodes which are connected via a BSC line or which are using an SNA session or a TCP/IP connection. If PNET uses an SNA session, PNET communicates with VTAM, which contains the definition for the hardware used (an ICA can be used on a 9221, or an OSA on a 9672 system). If PNET uses a TCP/IP connection, PNET communicates with TCP/IP for VSE/ESA, which contains the definition for the hardware used (for example, an OSA).

Figure 2 on page 5 shows six nodes (NODEA, NODEB, NODEC, NODED, NODEE, and NODEF). Some nodes which are not connected directly to each other can exchange data via two different routes. For example, NODEC can send data to NODEB via NODEA or via NODED. Some nodes which are connected directly can exchange data via an alternate route in case the direct connection cannot be established. If, for example, the direct connection between NODEB and NODED cannot be established, NODEB can send data for NODED via NODEA and NODEC. NODEE uses two different TCP/IP connections:

- a normal TCP/IP connection to NODEB
- a secure TCP/IP connection to NODEF with the SSL (secure sockets layer) feature.



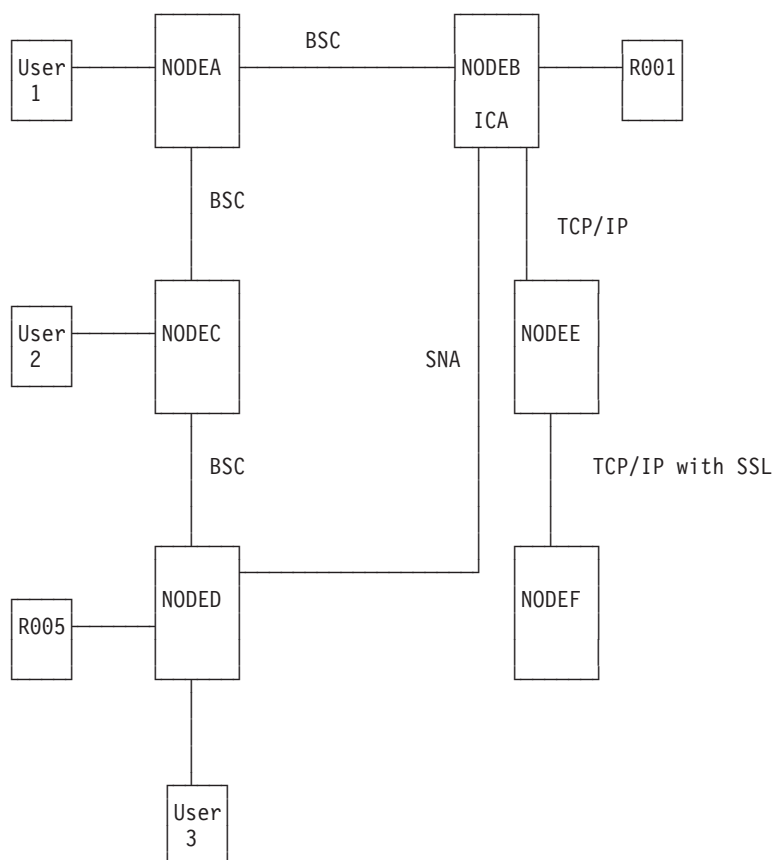


Figure 2. Sample Network

## Connecting the Network

The route information for all job entry subsystems within a network is contained in the NDT, which is generated via the PNODE macro. Additionally, VTAM maintains connection information for SNA PNET application-to-application sessions. Every VSE/POWER node of the network should have a copy of the NDT specifying all nodes in the network, unless you want to prevent particular nodes from communicating with certain other nodes in the network.

At connection time, PNET must be instructed (via a PSTART command) to prepare the line for transmission. The PSTART command for a node must be issued by operators at both nodes involved in the connection. If the other node is not a PNET node, an equivalent start command for the network must be given. Figure 2 shows an example of a small network with users and terminals attached at various nodes. This example is used in several parts of this book. The terminals may be any of the terminals supported by the VSE/POWER RJE support. The user may be anyone with access to VSE/POWER by the spool-access support.

### CTCA Support

VSE/POWER supports the virtual CTCA under VM. This CTCA support allows data to be exchanged between two virtual machines under control of a VM system if:

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

With this support, there is no need to use a hardware link (such as a telecommunication controller) for the exchange of data between virtual machines under the same VM system.

The support also offers an alternative to the VSE/POWER shared spooling function. All VSE/POWER systems can be linked together using PNET rather than a common spool file. This improves the overall reliability, since none of the other systems is impacted when one system ends abnormally.

**Note:** To use the real CTCA, let VTAM control the CTCA and run PNET/SNA.

---

### Establishing PNET Support

Before you can use PNET, you must consider generation and setup requirements that are either of a general networking nature or are related to a specific PNET communication type.

If your network uses SNA links, you should consult "VTAM Generation and Interface Requirements" in *VSE/POWER Administration and Operation*. If your network uses TCP/IP connections, you should consult "TCP/IP for VSE/ESA Setup and Interface Requirements" in *VSE/POWER Administration and Operation*.

## VSE/POWER Generation Considerations for PNET

### POWER Macro Requirement

The specification required in the POWER macro to include PNET support in general is:

```
PNET=phasename
```

where phasename is the name of the phase under which the NDT has been cataloged. This specification is sufficient to cause the loading of all phases necessary to include the BSC, CTCA, SNA, TCP, and SSL support for network operation. The specification of PNET= is also shown in Figure 3 on page 8.

The specified NDT name may be overridden by specifying:

```
SET PNET=newname
```

in the autostart procedure, where 'newname' is the phasename of any other NDT.

VSE/POWER systems participating in a network may be connected by BSC lines, virtual CTCAs, or SNA, or TCP/IP connections.

### BSC Lines (Controlled by VSE/POWER)

BSC lines that are attached to an IBM 37xx Communications Controller are operated in emulation mode. For PNET, the 37xx must be generated to emulate a

2703. Every attached PNET line must be defined during VSE/POWER generation by means of the PLINE macro, and also must be added during VSE IPL as the correct unit, for example:

```
ADD cuu,2703,EML
```

### Virtual CTCAs (Controlled by VSE/POWER)

Natively, VSE/POWER supports only the virtual CTCA as described in “CTCA Support” on page 6. The CTCA support is identical to the BSC support (regarding its external behavior), except that it does not require definition by means of the PLINE macro. Like a BSC line, a CTCA must be added during VSE IPL as the correct unit:

```
ADD cuu,CTCA,EML
```

### SNA Links

SNA PNET application-to-application sessions using SDLC lines, a real or virtual CTCA, Token Ring, or OSA require that

- the PNODE LOCAL=YES entry specify the APPLID=name operand if the default for APPLID=own-nodename is not desired
- the PNODE LOCAL=NO entries specify the APPLID=name operand

### TCP/IP Connections

Two types of PNET support exist for TCP/IP connections:

- PNET TCP, which processes all TCP/IP connections that do not use the SSL feature
- PNET SSL, which processes all TCP/IP connections that make use of the SSL feature.

Each support is activated whenever at least one remote node is defined that uses the appropriate support. Such a remote node is defined via the PNODE macro using the operand LOCAL=NO and

- the operand IPHOSTAD or IPHOSTNM to use a TCP/IP connection without the SSL feature
- the operand ISHOSTAD or ISHOSTNM to use a TCP/IP connection with the SSL feature.

For IPHOSTAD and ISHOSTAD, the dotted-decimal IP address has to be specified, and for IPHOSTNM and ISHOSTNM the symbolic logical hostname.

In addition the operand PORT or SPORT can be specified if the default values are not to be used. PORT is used for PNET TCP support (default is 175), SPORT is used for PNET SSL support (default is 2252). If PORT or SPORT is used for the local node (LOCAL=YES), the local node uses the specified value for incoming connection requests, which means the remote node must use the specified number in its definition for your local node. If PORT or SPORT is used for a remote node (LOCAL=NO), the local node uses the specified value for its outgoing connection request whenever a PSTART command for the remote node has been issued. This means the remote node itself must have used the specified number in its definition for incoming requests.

### PLINE Macro Requirements (BSC Only)

This macro is needed only if the connection is a BSC line. For the required PLINE operands see *VSE/POWER Administration and Operation*. The PLINE macros must be included in the same assembly as the VSE/POWER generation, because it is contained in the same cataloged phase.

An example of the PLINE required at NODEA is in Figure 3 on page 8.

## Establishing PNET Support

**The TIMEOUT Operand:** The TIMEOUT operand for PNET specifies the number of minutes the line is allowed to fail in its attempt to contact the remote node. When this time expires, a leased line is stopped and a switched line is disabled and a restart is initiated.

TIMEOUT should be used for a leased line only if it is necessary to avoid processing timeouts every three seconds.

TIMEOUT should be used for a switched line to eliminate erroneous calls after timeout; it makes the line available for a new call (especially critical with auto-answer). If this timeout limit is exceeded, and automatic restart for the line is allowed, PNET enters the restart processing routines for the node concerned.

### Example of a VSE/POWER Generation

Figure 3 is an example of a VSE/POWER generation that could be used at NODEA (as shown in Figure 2 on page 5); the example assumes that the line to NODEB is a leased BSC line and the line to NODEC is a switched BSC line. Operands that are allowed to default are omitted; for default values, see *VSE/POWER Administration and Operation*.

```
POWNODEA POWER ACCOUNT=YES, Accounting support
          JSEP=(2,2), Job separators
          SPOOL=YES, Cross-partition support
          TRACESZ=64, 65.536 bytes trace area
          PNET=NODEANDT Name of the NDT for NODEA

          PLINE ADDR=X'010', Line to NODEB
          PSWRD=LINEB Password which NODEB must use

          PLINE ADDR=X'020', Line to NODEC
          PSWRD=LINEC, Password which NODEC must use
          TIMEOUT=5, Timeout of five minutes
          SWITCH=YES Switched line

END
```

Figure 3. Example of an IBM VSE/POWER Generation including PNET/BSC Lines

### PNODE Macro Requirements

This macro is used to specify the NDT, that is, the layout of the network. It is not necessary to know what system is at the other end of the connection/session to enable communication. It is necessary, however, to know that system if you want to run a job in that system.

For all non-adjacent nodes, you need to know only the first node that jobs, output, messages, or commands must pass on the way to the final destination. It is the responsibility of the next node to know how to get the jobs, output, messages, or commands further on the path to the final destination.

When any connection to another node uses an SNA link, the APPLID operand must have been coded in the PNODE macro for the local node unless it is desired to use the nodename as the application ID. This application ID makes this node known to VTAM and must be defined in the startup procedures for the local VTAM system. The application IDs specified in other PNODE macros must be made known as cross-domain resources to the VTAM on the local system.

In the NDT, you can specify alternate routing for a node. This means that, if there is a problem with a particular connection, the job or output may be sent via an

alternate route. In Figure 2 on page 5, NODEB has an alternate route to NODED via NODEA. This is used, if started, whenever the primary route - the direct session between NODEB and NODED - is not available.

Alternate routing is more fully discussed under “Path Selection” on page 25.

The NDT is generated by means of PNODE macros. For a full description of the operands of the PNODE macro see *VSE/POWER Administration and Operation*. The NDT must have an entry for every node within the network to which it is required to transmit data. Entries for nodes to which a communication path is to be established require your special attention: Certain operands which you may specify for these nodes influence the performance and integrity of the network. For more information on performance tuning, see “Performance Considerations” on page 51.

The PNODE macros should be checked to ensure that:

- All nodes are defined.
- All nodes that are referenced in any ROUTE1 or ROUTE2 operands are available for a connection or a session.
- Any passwords specified for the nodes correspond with the passwords which the other nodes have defined.
- The size of the transmission buffer specified is consistent with that specified on the other node.
- The number of buffers for transmitters and receivers is consistent.

Figure 4 on page 10 shows, as an example, the PNODE generation macros required for the network shown in Figure 2 on page 5.

## Establishing PNET Support

For Node A:

```
NODEANDT PNODE NODE=NODEA, LOCAL=YES, PWD=NODEA
PNODE NODE=NODEB, AUTH=JOB, PWD=NODEB, BUFSIZE=800
PNODE NODE=NODEC, BUFSIZE=800, AUTH=NET, MAXBUF=(8,8), PWD=NODEC
PNODE NODE=NODED, ROUTE1=NODEC
END
```

For Node B:

```
NODEBNDT PNODE NODE=NODEB, LOCAL=YES, PWD=NODEB, APPLID=VTAMB
PNODE NODE=NODEA, AUTH=NET, BUFSIZE=800, PWD=NODEA
PNODE NODE=NODEC, ROUTE1=NODEA, AUTH=NET
PNODE NODE=NODED, APPLID=VTAMD, ROUTE2=NODEA
PNODE NODE=NODEE, IPHOSTAD=019.009.019.144
END
```

For Node C:

```
NODECNDT PNODE NODE=NODEC, LOCAL=YES, PWD=NODEC
PNODE NODE=NODEA, AUTH=NET, BUFSIZE=800, MAXBUF=(8,8), PWD=NODEA
PNODE NODE=NODEB, ROUTE1=NODEA, AUTH=NET
PNODE NODE=NODED, BUFSIZE=500
END
```

For Node D:

```
NODEDNDT PNODE NODE=NODED, LOCAL=YES, APPLID=VTAMD
PNODE NODE=NODEA, AUTH=NET, ROUTE1=NODEC
PNODE NODE=NODEB, APPLID=VTAMB, ROUTE2=NODEC, PWD=NODEB
PNODE NODE=NODEC, BUFSIZE=500
END
```

For Node E:

```
NODEENDT PNODE NODE=NODEE, LOCAL=YES
PNODE NODE=NODEB, IPHOSTAD=019.009.019.145
PNODE NODE=NODEF, ISHOSTAD=019.009.119.045
END
```

For Node F:

```
NODEFNDT PNODE NODE=NODEF, LOCAL=YES
PNODE NODE=NODEE, ISHOSTAD=019.009.019.044
END
```

*Figure 4. PNODE Generations for the Sample Network*

## CTCA Requirements

VSE/POWER supports the virtual CTCA under VM. Before you can use a CTCA, you must define it to the VM system by means of the CP DEFINE command. This command must be given at both VM guest machines:

```
DEFINE CTCA cuu
```

Next, the virtual CTCAs of the two guest machines must be logically connected by means of the CP COUPLE command:

```
COUPLE cuu1 user-id cuu2
```

For more information on the CP DEFINE and COUPLE commands, see *z/VM V4R1.0 CP Command and Utility Reference*, SC24-6008.

To define the CTCA to VSE, use the IPL ADD command:

```
ADD cuu,CTCA,EML
```

### VTAM Requirements

If you use SNA links for communication to and from VSE/POWER, certain requirements for the installation of VTAM and NCP (if an IBM Communications Controller is used) must be satisfied:

- VTAM must be installed on the systems planned to participate in the sessions. If the system used for network operation can support an ICA or a CTCA, VTAM without NCP may be used for PNET communications.
- NCP must be resident in the IBM 37xx.
- You must define the configuration of the network to VTAM by coding B-BOOKs; the definition must show the configuration as it appears to VTAM.
- You can use the pacing facility of VTAM to prevent PNET from filling all the VTAM buffers when PNET transmits jobs or output.

For information on how to specify B-BOOKs and pacing, see Appendix A, “Examples of Defining VTAM BOOKs and Pacing,” on page 81.

### TCP/IP Requirements

If you use TCP/IP for communication from and to VSE/POWER, the installation of TCP/IP for VSE/ESA requires:

- a DEFINE LINK command for each connection device “cuu”
- a DEFINE ROUTE command for each routing to an attached host (in some cases, the information can be obtained dynamically)
- If you use the SSL feature, a private key must be generated. For more information, see chapter “Configuring Your z/VSE Host for SSL” in the *z/VSE e-business Connectors User’s Guide*.

---

## PNET Initialization

The networking function may be initialized as part of the VSE/POWER initialization process. After the message VSE/POWER INITIATION COMPLETED ... has been received, the operator must issue the PSTART command for the node with which he wishes to communicate. Autostart may also be used for the initialization process; in this case, the PSTART command for the node must be placed in the input stream after the FORMAT= statement.

### Loading the PNET System

If PNET=phasename has been specified in the POWER generation macro, VSE/POWER loads the phases required to operate the network during the VSE/POWER initialization process. If there is insufficient GETVIS space available to load the Network Definition Table (NDT), or the NDT cannot be found in the library, message 1Q08I is issued, but initialization of the base VSE/POWER system is not terminated.

At the end of the initialization process, the NDT is loaded into the partition GETVIS space. The name of the NDT, as specified in the VSE/POWER generation, may be overridden by using the statement:

## PNET Initialization

SET PNET=newname

In either case, the NDT is checked as explained under “Correctness Checking of the NDT.”

## Network Definition Table (NDT)

### Loading the NDT

The NDT is automatically loaded during PNET initialization. The NDT may also be loaded by the operator by means of a PLOAD command. To allow the dynamic loading of an NDT, a new area in the GETVIS space of the VSE/POWER partition is used for every PLOAD command issued by the operator. This means that, if the operator frequently issues the PLOAD command, the available GETVIS space may be exhausted (message 1Q7AI issued). In this case, and if an NDT is to be loaded, the operator must:

1. issue the PEND command to terminate VSE/POWER,
2. reinitialize VSE/POWER, and
3. reissue the PLOAD command.

Normally you do not have to reload an NDT since a network is not a dynamically changing environment.

It is possible to define entries for nodes which do not yet exist in the network. If defined, this avoids the need to regenerate the NDT when the node does exist. See “Starting of z/VSE Subtasks for TCP/IP Connections” on page 13.

**Note:** The name specified for the local node of any NDT loaded after PNET initialization must be identical to the local node name specified in the originally loaded NDT. If it is required to change the name of the local node, the procedure to be followed is explained under “Changing the Name of the Local Node During Warmstart” on page 45.

### Correctness Checking of the NDT

After the NDT is loaded (by the initialization process or via the PLOAD command), it is checked for correctness. The operator is informed if errors occur. The following points are checked:

- The descriptor 'Version 06.0' of the NDT is checked to ensure that its generated structure is compatible with the installed level of VSE/POWER.
- The NDT is then checked to ensure that all nodes specified in any ROUTE operands are actually defined in the table as links or direct sessions.
- All application IDs defined in the NDT are checked to ensure that they are uniquely defined.
- The name of the local node is checked to ensure that it is the same as the local node name of any NDT which is currently being used.

For TCP nodes (using TCP/IP without the SSL feature):

- If the local node (LOCAL=YES) of the new NDT specifies a different PORT number, the new NDT is rejected with message 1RA5I RC=0005 if the TCP/IP interface is already open.
- The new NDT is also rejected if it contains multiple node definitions for the same IP address with the same port.

For SSL nodes (nodes using TCP/IP with the SSL feature):



- If the local node (LOCAL=YES) of the new NDT specifies a different number for the SPORT operand, the new NDT is rejected with message 1RA5I RC=0006 if the TCP/IP SSL interface is already open.
- If the local node (LOCAL=YES) of the NDT specifies a different type of security protocol for the SECTYPE operand, the new NDT is rejected with message 1RA5I RC=0007 if the TCP/IP SSL interface is already open.
- If the local node (LOCAL=YES) of the NDT specifies a different key database sublibrary for the KEYSRING operand, the new NDT is rejected with message 1RA5I RC=0008 if the TCP/IP SSL interface is already open.
- If the local node (LOCAL=YES) of the NDT specifies a different key sublibrary member for the DNAME operand, the new NDT is rejected with message 1RA5I RC=0009 if the TCP/IP SSL interface is already open.

If message 1RA5I is issued during loading of the new NDT, the NDT is not activated, and:

- if loaded by the initialization process, the networking function is not started.
- if loaded via the PLOAD command, the old NDT is retained and used.

If any error was found during the loading of the NDT, the operator should inform the system programmer about the error. When the error is corrected and the NDT assembled again and cataloged, the operator may issue a PLOAD command to load the new phase.

### Starting of z/VSE Subtasks for TCP/IP Connections

If an NDT is successfully loaded, the following z/VSE subtasks may be attached in the VSE/POWER partition:

- The TD subtask is attached if the NDT contains at least one TCP node, a node communicating via a TCP/IP connection without the SSL feature. The node has been defined using the operand IPHOSTAD or IPHOSTNM.
- The SD subtask is attached if the NDT contains at least one SSL node, a node communicating via a TCP/IP connection with the SSL feature. The node has been defined using the operand ISHOSTAD or ISHOSTNM.

These subtasks remain active even if another NDT is later loaded without a non-local TCP or SSL node, respectively. This enables the local VSE/POWER to listen for remote connection attempts and respond to another host's attempt to establish a connection with an indication that the local system is active but no local connection start command has yet been issued.

To deactivate one of these subtasks, the command 'PSTOP TCPIP' or 'PSTOP TCPSSL', respectively, can be used.

If an NDT is successfully loaded, messages are issued to trace the initialization of the subtask. An example of the PLOAD command that could be used on NODEB is shown below, which starts the TD subtask used for TCP nodes (nodes not using the SSL feature).

```
PLOAD PNET,NODEBNDT
```

```
F1 0001 1RB4I PLOAD NETWORK DEFINITION TABLE NODEBNDT LOADED
F1 0117 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL INITAPI ISSUED
F1 0001 1RTMI TCP/IP SUBTASK ATTACHED
F1 0117 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL SOCKET ISSUED
F1 0117 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL GETHOSTID ISSUED
F1 0117 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL BIND ISSUED
F1 0117 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL LISTEN ISSUED
```

## PNET Initialization

Note the different z/VSE task identifiers (immediately following the partition identifier F1). The messages with task identifier 0001 are issued by the VSE/POWER maintask, whereas the messages with task identifier 0117 are issued by the TD subtask, which runs as a z/VSE subtask in the VSE/POWER partition.

If the program TCP/IP for VSE/ESA has not yet completed its initialization (usually running in partition F7), the following messages are issued.

```
PLOAD PNET,NODEBNDT
```

```
F1 0001 1RB4I PLOAD NETWORK DEFINITION TABLE NODEBNDT LOADED
F1 0117 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL INITAPI ISSUED
F1 0001 1RTMI TCP/IP SUBTASK ATTACHED
F1 0117 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL SOCKET ISSUED
F1 0117 1RT2I TCP/IP: EZASMI MACRO-REQUEST SOCKET FAILED, RC=-0000001,
ERRNO=+0001117, TS=08
F1 0117 1RTJA TCP/IP: INITIALIZATION OF INTERFACE UNSUCCESSFUL, RETRY GOING
ON
```

If TCP/IP for VSE/ESA cannot complete its initialization, issue the PSTOP TCPIP command to delete message 1RTJA.

An example of the PLOAD command that could be used on NODEF is shown below, which starts the SD subtask used for SSL nodes (nodes using the SSL feature).

```
PLOAD PNET,NODEFNDT
```

```
F1 0001 1RB4I PLOAD NETWORK DEFINITION TABLE NODEFNDT LOADED
F1 0117 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL INITAPI ISSUED
F1 0001 1RVMI TCP SSL SUBTASK ATTACHED
F1 0117 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL SSL-INIT ISSUED
F1 0117 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL SOCKET ISSUED
F1 0117 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL GETHOSTID ISSUED
F1 0117 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL BIND ISSUED
F1 0117 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL LISTEN ISSUED
```

If the TCP/IP for VSE/ESA program has not yet completed its initialization, the 'SOCKET' socket call fails and, as previously described, the messages 1RT2I and 1RTJA are issued. If TCP/IP for VSE/ESA cannot complete its initialization, issue the PSTOP TCPSSL command to avoid message 1RTJA.

An example of the PLOAD command that could be used on NODEE is shown below, which starts both subtasks, TD and SD.

```
PLOAD PNET,NODEENDT
```

```
F1 0001 1RB4I PLOAD NETWORK DEFINITION TABLE NODEENDT LOADED
F1 0105 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL INITAPI ISSUED
F1 0001 1RTMI TCP/IP SUBTASK ATTACHED
F1 0118 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL INITAPI ISSUED
F1 0001 1RVMI TCP SSL SUBTASK ATTACHED
F1 0105 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL SOCKET ISSUED
F1 0118 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL SSL-INIT ISSUED
F1 0118 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL SOCKET ISSUED
F1 0118 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL GETHOSTID ISSUED
F1 0105 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL GETHOSTID ISSUED
F1 0118 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL BIND ISSUED
F1 0118 1RT7I TCP SSL: INTERFACE STARTING, SOCKET CALL LISTEN ISSUED
F1 0105 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL BIND ISSUED
F1 0105 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL LISTEN ISSUED
```

Note the different z/VSE task identifiers of the TD and SD subtasks. Each of these tasks issues its own initialization messages.

## Starting a PNET Session

A session between two nodes is started by means of the PSTART command. There are small differences between the start of a BSC, CTCA, or TCP/IP connection, or an SNA session; these are discussed below.

After the PSTART has been issued, the attempt to establish the session remains pending until the node with which you are trying to communicate also issues:

- A PSTART command, or
- An equivalent command if the system with which you try to communicate is not a system with VSE/POWER.

The connection between two nodes is not considered complete until the successful exchange of SIGNON records has been achieved. The sign-on process is used to agree on the size of buffers to be used between participating nodes and to optionally check authority by means of password exchange.

Once a connection or session is established and sign-on is complete, any job/output that can be dispatched and is destined for or routed via the other node automatically starts to be transmitted.

For further control of transmitters and receivers, see “Activating Transmitters and Receivers” on page 24 and “Draining Transmitters and Receivers” on page 25. Reasons for using multiple transmitters or receivers are discussed under “Number of Transmitters and Receivers” on page 53.

The initialization of the network, including the issuing of the PSTART command for all nodes that need be activated, may also be accomplished by inserting the PSTART command in your autostart statements. For all the possible options of the PSTART command, see *VSE/POWER Administration and Operation*.

### BSC Connection

PNET can communicate with other nodes via a half-duplex BSC line. (The line discipline used is called multi-leaving). Both leased line and switched line connections are supported.

An example of the PSTART command that could be used on NODEA to set up communication with NODEB is shown below (optional operands are not shown).

The example assumes that no in-core line trace is required and that the default automatic restart is used. For details on the PSTART command, see *VSE/POWER Administration and Operation*.

```
PSTART PNET,NODEB,,010
```

An optional node and/or a line password may be specified. For more information on password processing, see “Providing Security Access to Nodes and Lines” on page 18. For further information of automatic restart see “Automatic Restart of Nodes” on page 43.

For a switched line you have to issue the PSTART command prior to dialing the remote node.

If the desired connection cannot be established because the corresponding PSTART command has not yet been entered at the other node, message 1RC6I CONNECTION PENDING FOR NODE ... is issued after nine seconds; it reappears about every 27 minutes, depending on the ‘timeout’ events signaled by the communication

## PNET Initialization

controller. This message confirms that the physical network line functions are correct up to the modem located on the own side.

### CTCA Connection

Starting a CTCA connection is the same as starting a BSC line, except that for a CTCA (which does not require a PLINE generation macro) no default outgoing line password exists. Although a line password may be specified in the PSTART command if the receiving (non-PNET) system needs it, VSE/POWER does not check any incoming line password for correctness.

An example of the PSTART command that could be used on NODEA to set up communication with NODEB is shown below:

```
PSTART PNET,NODEB,,400
```

If the desired connection cannot be established because the virtual CTCAs of the two guest machines are not yet 'COUPLED' or the other node has not yet entered its PSTART command, message 1RC6I CONNECTION PENDING FOR NODE ... is issued after 1.5 minutes; it reappears about every 12 minutes. For CTCA connections, the occurrence of message 1RC6I depends on internal time intervals, since no BSC-like 'timeout' events occur.

**Note:** Even if all provisions have been made correctly (including the PSTART command on both sides), the node connection might still not be completed immediately, because the initial 'contact' attempts collided adversely. After some seconds, the initial contact is tried again, thus enabling the systems to recover without operator interference.

### SNA Session

When any connection to another node uses an SNA link, the APPLID operand must have been coded in the PNODE macro for the local node. See "PNODE Macro Requirements" on page 8. Furthermore, the corresponding VTAM sublibrary must be included in the phase search chain of the VSE/POWER partition.

For example, this PSTART command from NODEB would start the session with NODED:

```
PSTART PNET,NODED
```

You may optionally specify a node password. For more information on password processing, see "Providing Security Access to Nodes and Lines" on page 18.

The first PSTART command given for an SNA node also opens the VTAM application interface to VSE/POWER. If VTAM is not active at the time the first PSTART command is given, VSE/POWER, at intervals, retries to open the interface. All PSTART commands which were given for SNA nodes are queued until it is possible to open the VTAM interface. The start of BSC, CTCA, TCP, or SSL nodes is not affected by the waiting SNA nodes.

If the connection to another SNA node cannot be established because VTAM cross domain resources are not activated yet or because the corresponding PSTART command has not yet been entered at the other node, VTAM failure messages and message 1RC6I CONNECTION PENDING FOR NODE ... are issued. Every 10 seconds a reattempt is made for session establishment, resulting in repetitions of VTAM messages; the message 1RC6I reappears only about every 20 minutes.

To extend the reattempt interval of 10 seconds to x minutes, use the 'SET OPNDST=xxx' statement, included in the VSE/POWER autostart procedure.

If xxx is greater than 20 minutes, message 1RC6I will reappear only with every connection reattempt. Also, if the remote node is controlled by VSE/POWER, the frequency of message 1RC7I NODE ... AWAITING CONNECTION at the remote node may be influenced by SET OPNDST.

For more information on the SET OPNDST statement, see *VSE/POWER Administration and Operation*.

### TCP/IP and TCP/IP SSL Connections

When a TCP/IP connection is used to communicate with another node, the IPHOSTAD or IPHOSTNM operand for a TCP node, or the ISHOSTAD or ISHOSTNM operand for an SSL node, must have been coded in the PNODE macro. See "TCP/IP Connections" on page 7. The PSTART command has the same format for a TCP and SSL node. An example of the PSTART command that could be used on NODEE to set up communication with NODEB is shown below:

```
PSTART PNET,NODEB
```

You may optionally specify a node password. For more information on password processing, see "Providing Security Access to Nodes and Lines" on page 18.

Immediately after the PSTART command, the following message is issued:

```
1RC6I CONNECTION PENDING FOR NODE ...
```

Message 1RC6I reappears about every 12 minutes until the two nodes have successfully established a connection, which is reported by the following message:

```
1RB3I NODE ... SIGNED-ON ON LINK ....
```

If the connection cannot be established within a reasonable time (at most 4 minutes), see "TCP Node Cannot Sign On" on page 68 and "SSL Node Cannot Sign On" on page 72 for details about possible failures.

If the remote system (NODEB in the above example) is a z/VSE system and an NDT has been loaded which contains the definition for a TCP node, message 1RTHI is issued on the remote system indicating that a connection attempt was made and that another system is waiting for the missing PSTART command.

```
1RTHI TCP/IP: NODE ... AWAITING CONNECTION
```

**Providing Security for TCP/IP Connections:** If the operand ISHOSTAD or ISHOSTNM has been specified in the PNODE macro, the Secure Sockets Layer (SSL) feature is used to exchange data between two nodes. In this case, all job, list, and punch output, as well as messages and commands, are encrypted. Even network control records, which are exchanged during the initialization of the communication and contain for example the node password, are encrypted.

A private password and a certificate for the SSL feature can be generated by using the skeleton of SKSSLKEY.PROC, which can be found in ICCF library 59. Within this skeleton, a sublibrary and a membername must be specified to catalog a private password and certificate. The membername specifies the name of a private password and certificate (which are distinguished by the membertype) and must be specified in the DNAME operand of the PNODE macro. A different DNAME can be specified for each non-local node, but usually the same password and certificate are used for all nodes. In this case you specify DNAME for the local node and omit the DNAME for the non-local nodes. The sublibrary must be specified in the KEYRING operand of the PNODE macro for the local node.

## PNET Initialization

Once the connection between two nodes is established, the PINQUIRE command can be used to display information about the encryption used for this connection. The command PINQUIRE NODE=remote-nodename displays several messages, one of these has the following format:

```
1R56I LOCAL NODE IS ACTING AS CLIENT|SERVER,ENCRYPT=enc-type,CIPHER=X'cc'
```

where:

### **enc-type**

describes the type of encryption specified in the PNODE macro, which is either WEAK, NORMAL or STRONG.

**cc** is the cipher which is used for encryption after the initialization of the SSL feature has been successfully completed.

## Providing Security Access to Nodes and Lines

Passwords provide security for the network system. Using passwords prevents unauthorized users from creating sessions with a node.

VSE/POWER allows specifying the following types of passwords for use with all communications disciplines of PNET:

- Node password
- Line password

Node and line passwords are optional. If the receiving node does not have passwords generated, passwords are not checked by that node.

The passwords may be:

- Statically defined in the PNODE and PLINE (BSC only) generation macros, and
- Dynamically changed for this start by specifying the password fields in the PSTART command.

### **Node Password**

The node password is exchanged during the sign-on process between the two communicating nodes. If you choose to use password processing, you must specify a password in the local entry of the NDT or in the PSTART command.

If the receiving system has a password defined for your node, the password received is compared with that specified for your node. The following happens:

- If the two passwords match, the sign-on is accepted.
- If the two passwords do not match, the sign-on is rejected. A message informs the operator that there was a password error.

If a node password is *not* given in the PSTART command, VSE/POWER uses the password defined in the NDT for the local node. For example:

If NODEA and NODEB are communicating via BSC line X'010' on NODEA, the PSTART command which should be issued on NODEB is:

```
PSTART PNET,NODEA,,010
```

VSE/POWER uses the password for the local entry of the NDT at NODEB.

If a password error occurs, check the password in every node to ensure that the generated entry actually matches the password specified by the other node. For example, considering the PNODE generation defined in Figure 4 on page 10:

If the PSTART command shown below is issued at NODEB, it would be rejected by NODEA. (The correct PSTART command would be the one shown above.)

```
PSTART PNET,NODEA,NODEA,010
```

The specified node password (NODEA) does not match the password generated in NODEA for NODEB (which, in this example, is NODEB).

## Line Password

- BSC Lines

Use the PSWRD operand of the PLINE macro to specify the line password. The line password is exchanged during the sign-on process between the two communicating nodes. The password which is sent to the other node may be changed during the initialization process by specifying a different password with the PSTART PNET, .... command.

The line password is checked using the password defined for the line in the receiving node. If no password is defined for the line in the receiving node, checking is not done.

Assume, for example, that LINEB is defined at NODEA as line password for NODEB for communicating via line X'010' on NODEA. The PSTART command which NODEB must enter is:

```
PSTART PNET,NODEA,,010,LINEB
```

If no password is specified, the connection is not established because NODEA expects the line password LINEB.

- SNA links

Line passwords are not supported.

- CTCA lines

A line password may be optionally specified for CTCA lines in the PSTART command. This password is included in the SIGNON record which is exchanged between systems. If the receiving system is a z/VSE with VSE/POWER, this password is not checked. Some non-PNET systems check the password.

- TCP/IP connections

Line passwords are not supported, neither for TCP nodes nor for SSL nodes.

## PNET Initialization



---

## Chapter 2. Using PNET

---

### PNET Processing

#### Preparing a Job for Transmission

To transmit a job for running at another node, you must insert one additional operand in the \* \$\$ JOB statement.

The statement which could be used to submit job TEST from NODEA to NODED is as follows:

```
* $$ JOB JNM=TEST,XDEST=NODED
```

After this job has run, the output is directed back to the submitter on NODEA.

If the submitter wants to be notified when the job has run, the following additional operand must be specified:

```
* $$ JOB JNM=TEST,XDEST=NODED,NTFY=YES
```

If the output from the job should be directed to another node and/or user, the LDEST operand may be used in the \* \$\$ JOB statement:

```
* $$ JOB JNM=TEST,...,LDEST=(NODEC,USER2)
```

This definition will send all list output from the job to NODEC and the output is flagged as being for USER2.

#### Disposition in the Transmit Queue

Any job or output queue entry intended for transmission to another node has the following two disposition attributes:

1. its *local* disposition (intended for processing in the RDR/LST/PUN queue) as specified by the DISP= operand (with default DISP=D) of the \* \$\$ JOB/LST/PUN statement or as specified by field SPLDDP (see PWRSP) for Spool Access PUT-OUTPUT. The local disposition for output is also reflected in the DSHR in flags NDHGF2HB and NDHGF2HA when passed to other nodes. For details, see "Network Job Entry Formats and Protocols", SC23-0070.
2. its *transmission* disposition (intended for processing in the XMT queue) as specified by the TDISP= operand (with default TDISP=D) of the \* \$\$ JOB/LST/PUN statement or as specified by field SPLOTDP (see PWRSP) for Spool Access PUT-OUTPUT.

When a job/output entry is created and enters the transmit queue directly, its intended transmission disposition becomes effective and is made visible in the D(isp) column of an XMT queue display. The entry is then selected for transmission by a suitable active transmitter task according to the TDISP=D|K|H|L rules described in *VSE/POWER Administration and Operation*. In case of the default TDISP=D, for example, the queue entry is available for transmission and is deleted from the transmit queue after transmission is complete. Because only the local processing disposition is passed to other nodes, the following rules apply:

- the intended transmission disposition takes effect only in the XMT queue of the originating node. However, on intermediate VSE/POWER nodes the entry

## PNET Processing

passes through the XMT queues with disposition D. Hence, if TDISP=K, the originating node will retain a copy of the transmitted entry in its XMT queue.

- the intended local disposition of the queue entry is preserved and takes effect only on the final target node, or on any intermediate node, where the entry is moved by operator influence from the XMT queue to the local RDR/LST/PUN queue.

When the transmission disposition of an entry in the XMT queue is changed by the PALTER ...,DISP= command and the entry is later moved to a local queue, the (changed) transmission disposition becomes invisible. On the local queue the transmission disposition is not displayed and cannot be altered. It becomes visible and effective again as soon as the entry is moved back to the XMT queue. The same rules apply for the local disposition of an entry in a local queue, with local disposition preserved when entry is moved to the XMT queue and re-established when the entry is moved back to local again.

When a job/output entry is created and first enters a local queue, the entry can be moved by the PALTER ...,NODE= command to the transmit queue, where the intended transmission disposition becomes effective. If, however, the PALTER ...,NODE=,DISP= command is used, where the disposition is also altered, it becomes effective as the transmission disposition (and overwrites any intended setting). The corresponding behavior applies when an entry is moved from the XMT queue to a local queue and a disposition is specified at the same time in the PALTER command. See also the DISP operand of the PALTER command in *VSE/POWER Administration and Operation*.

As expected, the PALTER ...,NODE=,DISP= command moving an entry from a local to the transmit queue preserves the local disposition. Only if the SET PALTER=BOTHDISP autostart statement was activated during VSE/POWER startup, both the transmission *and* local dispositions are altered to the specified DISP= value. This behavior has been offered for compatibility with VSE/POWER 4.1.0 and earlier versions. For further details see the 'SET' statement in *VSE/POWER Administration and Operation*.

Remember the following main disposition rules:

- Disposition in a queue display of a local queue means local disposition, namely execution disposition for jobs in the RDR queue or processing disposition for output entries in the LST/PUN queue. The transmission disposition is invisible and cannot be altered on a local queue.
- Disposition in a queue display of the XMT queue means transmission disposition. The local disposition is invisible and cannot be altered on the transmit queue.
- All queue entries enter the transmit queue with their intended TDISP|SPLOTDP transmission disposition, unless overwritten by the PALTER command.
- When jobs are read in or output is created for the XMT queue by a specified target node name, they will enter this queue with transmission disposition H (see message 1RA1I) as soon as the node name is unknown to the originating system.
- When queue entries are moved between transmit and local queues, the disposition of the queue just 'left' is preserved.
- Whenever received entries enter the local queues of the final target node, the preserved local disposition becomes valid.

## Priority in the Transmit Queue

When a job or output is created and enters the transmit queue, its processing priority for local queues, as defined by default rules or by the PRI= JECL operand, is preserved and is used *at the same time* as the transmission priority, which influences the selection by job or output transmitters. The same applies if a queue entry is moved by the PALTER command from a local queue into the transmit queue. For entries already residing in the transmit queue, any further alteration of priority influences only the transmission priority. Because only the local processing priority is passed to other nodes, this priority takes effect again:

- as the transmission priority when the job or output is received on intermediate nodes
- as the local processing priority when the job or output is received on its target node
- as the local processing priority when the job or output is moved by the PALTER command from the transmit queue of another node to a local queue.

If, however, both destination and priority of a local queue entry are changed by one PALTER command, the specified priority takes effect as the transmission priority. When an entry is moved from the transmit queue to a local queue and the PALTER command changes priority at the same time, it takes effect as the local processing priority.

The following rules apply:

- Priority in a queue display of local queues means local processing priority.
- Priority in a queue display of the XMT queue means transmission priority.
- All queue entries enter the transmit queue with their current processing priority (including priority changes on local queues). The local priority is preserved in this case.
- Whenever XMT queue entries are PALTER'ed to local queues, the preserved local priority becomes effective again (unless overwritten by the same PALTER command).

## Displaying Status Information

Various PNET status information can be obtained through the PDISPLAY and PINQUIRE commands.

The PDISPLAY PNET command provides the current active PNET connection configuration:

- active nodeids
- routing information
- security authorization
- buffer size
- SNA APPLID or TCP/IP address
- TCP/IP port number

The PINQUIRE command provides:

- with the NODE=local-nodeid option, statistics on currently started nodes and the TD and SD subtask status and IP address information
- with the NODE=nodeid option, the current PNET connection status for the given remote node
- with the ALL option, the current PNET connection status for all started nodeids and/or all PNET/BSC lines (and further non-PNET information)
- with PNET, the current connection status of all started nodeids

- with PNETxxx (where xxx = BSC, CTC, SNA, TCP, or SSL), the current connection status of all nodeids started for the specified communication type.

### Job Number Assignment

VSE/POWER assigns the VSE/POWER job number to every job when it first enters a VSE/POWER system. This job number is stored together with other information about the job. The original job number travels together with the job through the network and it is displayed in the 1RB5I (entry received) message together with the new jobnumber assigned to the job by the receiving system. The PNET user exit routine may be used to influence VSE/POWER so that the receiving system queues the job by its original number.

### Activating Transmitters and Receivers

After a connection or session has been established, one job and one output transmitter and all job and output receivers are set into non-drained status. This means that transmission may start immediately if there is an eligible entry in the transmission queue.

For every active transmitter there must be a corresponding receiver available on the receiving node. This means that if JOB OUT transmitter 2 is activated, JOB OUT receiver 2 must be in a non-drained status on the receiving node. To activate transmitters and receivers, use the PACT command.

If you need to run multiple transmitters (possible reasons are given under “Number of Transmitters and Receivers” on page 53), the operator must use the PACT command to activate the required transmitter of the correct type (either job or out). A description of the PACT command is contained in *VSE/POWER Administration and Operation*.

An example of the PACT command required to start an additional output transmitter from NODEA to NODEB is:

```
PACT PNET,NODEB,TR2,OUT
```

VSE/POWER supports up to eight transmitters concurrently active on any session. These may be a mixture of job and output transmitters, but only seven transmitters of any one sort may be concurrently active. Certain JOB transmitters exclude certain OUT transmitters. Figure 5 shows which combinations of transmitters are *not* allowed.

	NOT ALLOWED					
JOB	2	3	4	5	6	7
OUT	7	6	5	4	3	2

*Figure 5. Transmitter Combinations Not Allowed*

From Figure 5 you see that “JOB transmitter 4” cannot be started while “OUT transmitter 5” is already in “non-drained” status. If you need to activate “JOB transmitter 4”, you must drain “OUT transmitter 5” by use of the PDRAIN command.

## Draining Transmitters and Receivers

*To drain a transmitter* means to put it into a state where it is no longer available for use by PNET for transmission of data to another node in the network.

*To drain a receiver* means to put the receiver into a state where it always rejects any request from a transmitter at another node to start a transmission.

If it is required to suspend receipt of data from another node for some period of time while still being able to transmit jobs or output, the receivers must be drained by means of the PDRAIN command. This may be necessary because, for example, the node is running short of spool space and needs to transmit data to create space to continue normal operation.

If you need to drain transmitters, use the PDRAIN command. The following is an example which drains JOB transmitter 2 from NODEA to NODEB:

```
PDRAIN PNET,NODEB,TR2,JOB
```

The following commands will cause all JOB and OUT (output) receivers for the connection to NODEB to be drained at the end of the current job:

```
PDRAIN PNET,NODEB,RV*,OUT,E0J  
PDRAIN PNET,NODEB,RV*,JOB,E0J
```

The commands will have no effect either on transmitters to NODEB nor on receivers which have been started for any other node.

For a full description of these commands, see *VSE/POWER Administration and Operation*.

If a transmitter tries to transmit to another node on which the corresponding receiver is in “drained” or “shut down” status, that transmitter is automatically set into drained status.

During shutdown of VSE/POWER (after PEND has been given), all transmitters for all nodes are set to indicate that they are drained at end-of-job. That is, when the job or output has finished transmitting, the transmitter will not search for more work to do, but will stop instead.

## Path Selection

VSE/POWER allows transmission via a prime or an alternate route. The decision which route to take is made by PNET automatically. The decision depends on the criteria given below. Alternate routing is not used for load levelling.

### Prime Route

The prime route between two nodes will always be chosen if a connection or session is available. That means that the two nodes are “signed on” to one another. The prime route will continue to be used for all traffic between the two nodes until the connection is broken. If the connection or session is broken, the system automatically tries to use any alternate route specified by the ROUTE2 operand in the PNODE macro.

It is possible that messages from another node including output in response to commands from any node, may be lost if a line error occurs while they are being transmitted. The message transmitter on the other node will try to send any new messages via any alternate route; any old messages queued on the prime route may be lost.

### Alternate Route

The alternate route, if one is defined for a node, will always be used if the prime route is not connected. The alternate route may use a different line discipline than the prime route. All transmissions, including commands and messages use the alternate route.

As soon as the prime route becomes available, the following happens:

- Transmission of new jobs and output start to use the prime route.
- Any messages and commands that are generated after the prime route became available use the prime route.
- Messages and commands that are already queued to the alternate route continue to use the alternate route.
- Jobs or output that are transmitted when the prime route becomes available continue to be transmitted via the alternate route until the end of the transmission. Then, the prime route is used again for any further transmission to the particular node.

## Shutting Down the Network

Shutting down VSE/POWER through the PEND command also causes shutting down of the network. You may, however, want to stop networking while still allowing the local VSE/POWER spooling function to continue operation.

### Shutting Down a Node

You can accomplish this by issuing one of the following commands for each of the active nodes:

```
PSTOP PNET,node_id,EOJ
PSTOP PNET,node_id
PSTOP PNET,node_id,FORCE    (BSC, CTC, TCP, and SSL only)
```

Depending on whether or not EOJ is specified, VSE/POWER does the following:

- With EOJ: The connection with the other node is terminated when transmission of all current jobs and output sent or received is completed.
- Without EOJ or FORCE: The connection is immediately terminated. Job and/or output that was just transmitted at both nodes is re-queued for transmission.
- With FORCE: The connection is immediately terminated without regard for the NJE communication protocol. Job and/or output that was just transmitted at both nodes is re-queued for transmission.

If you only want to stop transmitter or receiver activity for a particular node, use the PDRAIN command as specified under “Draining Transmitters and Receivers” on page 25.

### Shutting Down the TCP/IP Interfaces

To shut down networking support for TCP/IP connections only, special PSTOP commands exist:

```
PSTOP TCP/IP[,EOJ],FORCE]
PSTOP TCP/SSL[,EOJ],FORCE]
```

VSE/POWER does the following:

- With EOJ (default): when all TCP or SSL nodes have been stopped successfully (you must issue the 'PSTOP PNET,nodeid' command for signed-on nodes), the TCP/IP interface is terminated.

- With FORCE: the TCP/IP interface is abruptly terminated, accompanied by message 1Q2CI PSW=.... .... CANCEL BY PSTOP, followed by an IDUMP of the VSE/POWER partition. All TCP or SSL nodes are terminated. Jobs and/or output that were being transmitted are queued again for transmission.

**Note:** Use this command format only when the TD or SD subtask does not react to PSTOP TCPIP,EOJ or PSTOP TCPSSL,EOJ, although all TCP or SSL nodes have been stopped.

## Notify Support

The notify support within VSE/POWER allows users to be informed when:

- A job has run on a node within the network.
- Output has arrived at its final destination (only if the final destination is a VSE/POWER node).
- A job or output has been transmitted to the next node on its path to its final destination.

The request for notification may come from:

- The VSE/POWER NTFY operand in the \* \$\$ JOB statement, or
- The equivalent statement of any other supported system.

### Notify Support for VSE/ICCF Users

The notify support of VSE/POWER can be used to inform VSE/ICCF users of the progress of jobs and output as described above.

The requirements for using the support in conjunction with VSE/ICCF are described in *VSE/POWER Administration and Operation*. To use the notify support with VSE/ICCF:

- Specify the NTFYMSG operand of the POWER macro
- Submit a JECL statement (see Example 3 below).

VSE/POWER maintains the VSE/ICCF notify message queue: When the number of messages exceeds the number specified in NTFYMSG (when the queue is full), the *oldest* message is deleted, and the count for “lost message” is updated. The new message is *discarded* if GETVIS space is insufficient. The lost message count is included in the status report produced at termination time of VSE/POWER or by the PDISPLAY STATUS command.

### Examples - Notify Support

The following examples show how to specify JECL statements to obtain notification. The examples are based on the network shown in Figure 2 on page 5.

**Example 1 - Notify for Remote User:** The job is submitted by user R001 on NODEB for running on NODEA. The JECL statement causes the remote user R001 to be notified when:

- The job has been transmitted to NODEA,
- The job has run at that node,
- The job’s output has been received by NODEB.

\* \$\$ JOB JNM=T2,XDEST=NODEA,NTFY=YES,...

**Example 2 - Notify for Remote User:** The job is submitted by user 1 on NODEA and runs on NODEA. The JECL statements cause the remote user R001 to be notified when:

- The job has run, and when

## PNET Processing

- The output has been received by NODEB.
- \* \$\$ JOB JNM=T1,NTFY=(NODEB,R001)
- \* \$\$ LST DEST=NODEB

**Example 3 - Notify for VSE/ICCF User:** The job is submitted by user AAAA on NODEB for running on NODEC. The JECL statement causes the originating VSE/ICCF to be notified when:

- The job has been transmitted to NODEC,
  - The job has run at that node,
  - The job's output has been received by NODEB.
- \* \$\$ JOB JNM=T3,XDEST=NODEC,NTFY=YES,...

## Accounting

The NETACCT operand of the \* \$\$ JOB statement lets you specify an account ID for NJE jobs. VSE/POWER passes this account ID in the job header record to other nodes. The receiving JES2 node translates this ID according to its NETACCT table into a local 4-digit account number by which the NJE job can be billed. VSE/POWER does not make use of this value, but reflects it in the account records.

To reduce the amount of file space needed for the account records, you can save account file space by collecting account record information for only those networking records which you actually need. For details on the ACCOUNT operand of the POWER generation macro please see *VSE/POWER Administration and Operation*.

VSE/POWER writes three types of account records during PNET processing:

- PNET Network account record, ID 'N'.  
This is written for every session that was established when the session is terminated either normally or abnormally. It contains such information as sign-on and sign-off time, total transmission count during the session, line error statistics, and other information which is useful in accounting for the use of the telecommunication line.  
Message and command traffic on a session are accounted for in this record.
- PNET Receiver account record, ID 'V'.  
This is written for every job or output which is received on a session with another node. The records contain information specific to a job or to the output produced by a job. It lets you account transmission costs to individual jobs.
- PNET Transmitter account record, ID 'M'.  
This is written for every job or output which is transmitted on a session with another node. The records contain information which is specific to a job or to the output produced by a job. It lets you account transmission costs to individual jobs.

The account records contain the address of the cuu used for BSC and CTC nodes. For other nodes, the account records contain the type of the node in readable format, for example 'SNA', 'TCP', or 'SSL'.

The layouts of these account records are described in *VSE/POWER Application Programming*.



## PNET Receiver Exit

### Programming Interface Information

When generating VSE/POWER, you can specify an exit routine to be used whenever an input record (see 1. and 2. below) of a job or output is received from the network.

#### Loading the Exit

To load and enable a receiver exit routine at initialization time specify the NETEXIT operand in the POWER generation macro. For details on the operand, refer to *VSE/POWER Administration and Operation*.

The PNET receiver exit routine gets control for job or output. The receiver exit example shows after label REX140 (see page 89) how to decide whether the exit is called for job or output.

This exit routine may be the same routine that is used for input from local devices or another routine. If you add the exit routine after VSE/POWER initialization, you can load it dynamically with the PLOAD command.

You can activate and deactivate the PNET exit routine with the PVARV command. You can display information about the exit routine with the PDISPLAY EXIT command. For more information on the PLOAD, PVARV and PDISPLAY commands, please see *VSE/POWER Administration and Operation*.

#### Exit Routine Processing

To help you make this input routine reentrant, VSE/POWER passes a receiver exit *work area* to the routine. It is taken from the GETVIS space of the VSE/POWER partition and can be used by the routine to hold special variables and information needed along with that queue entry or to hold records which should be inserted. The size of the work area is defined by the user in the POWER macro (maximum 65,535 bytes, default zero) along with the phase name specified under the NETEXIT operand. The work area is initialized with X'00' and exists as long as a job or output is being received.

The first four bytes of the work area are the length field which is refreshed every time before passing control to the exit routine. The address of the work area is passed to the exit routine in register 3.

The PNET receiver exit routine gets control from VSE/POWER if:

1. The record received is:
  - A job header record
  - A job trailer record
  - A data set header record.
2. The record is JOB data and starts with:
  - // - z/VSE, z/OS, or OS/390 JCL
  - /\* - End of input or JES2/JES3 control statements
  - /& - End of job
  - \* - Comments or VSE/POWER JECL statements.

#### Programming Considerations

When coding your routine, observe the following:

## PNET Receiver Exit

- In your exit routine, use the DSECTs as listed in the example on page 85 to address fields within the network control records. The layouts of these records may change.
- The exit routine must not alter any fields in any control blocks to which it has access.
- The exit routine must not alter the content of registers 10, 11 and 12. These registers are reserved for VSE/POWER use only. All other registers can be used by the exit routine.
- The exit routine should not invoke VSE/Advanced Functions services (SVCs) which may wait, either voluntarily or involuntarily, for an event or resource. Instead, the exit must use the VSE/POWER IPW\$WFC macro to wait for a VSE/POWER event. A VSE wait from the exit routine would stop all the VSE/POWER task processing, including any devices, such as readers, printers and remote workstations under its control.
- Transmission control records (such as ML records) are not passed on to the exit routine.
- One byte X'FE' records, which are records that are for VSE/POWER internal use only, are always translated to X'03' records, that is to 'no operation records' (NOP).
- The exit routine must *not* use 31-bit addressing mode or access-register mode. When the exit routine loses control (for example, due to a page fault), the status of the currently used addressing mode and the access registers are not saved. Thus, when the exit routine gets control back again, the previously used addressing mode and the access registers cannot be restored. Because the exit routine must not use the access-register mode, the exit routine cannot use data spaces.
- VSE/POWER does not accept an exit routine which has already been loaded into the SVA.

### Register Content on Entry

The register content on entry to the receiver exit routine are:

Register	Content
0	Address of the record to be processed
1	Length of the record
2	RCB and SRCB in the low order bytes
3	Pointer to the work area for the exit routine
14	Return address to VSE/POWER
15	Entry point address of the exit routine

The exit routine has access to all control blocks available in the environment, for example:

Register	Content
10	Common Address Table
11	Receiver Task Control Block

### Return Specifications

Before returning control to VSE/POWER, the exit routine must specify a double-byte code in register 15 to signal what action VSE/POWER must take with this record. This code must be returned to VSE/POWER in the rightmost two bytes of return register 15. Valid register contents with return codes from a NETEXIT routine are:

**X'0000xx00' - Normal**

Process this record. Fields within the record may be changed, but the address or the length of the record must not be changed.

**X'0000xx04' - Delete**

Ignore this record. This is not valid for header or trailer records.

**X'0000xx08' - Insert**

Process the new record created by the exit routine. This is not valid for header or trailer records. On return to VSE/POWER, register 0 must contain the address of the record to be added and register 1 must contain its length. After the new record has been inserted, the old record is returned to the input routine.

**X'0000xx10' - Flush**

The complete VSE/POWER input is to be flushed. Current data reception is terminated. The transmitting system queues the data again, but in HOLD status. This could be used to prevent jobs from being received if they do not conform to the computing center standards.

**X'0000xx14' - Modify**

This is only valid for job header or job trailer records. The current job header or job trailer record has been changed by the exit routine. This could mean that a user section has been inserted in the record. On return to VSE/POWER, register 0 must contain the address of the total record and register 1 must contain the length of the total record.

**X'0000xx20' - Keep Original Number**

This code becomes effective only for job header records - it is ignored for other record types. VSE/POWER will then queue the received job/output with the original jobnumber from the originating node into one of the VSE/POWER queues. Thus any queue entry may be identified uniquely by the same jobname and jobnumber all throughout the network; however, duplicate jobnumbers on the receiving system are more likely.

**X'0000xx24' - Modify and Keep Original Number**

This code is a combination of X'14' and X'20'. The actions taken are those described under return codes X'14' and X'20'.

The variable X'xx' portion of the return codes may be:

**X'00' - Blank Truncation**

VSE/POWER truncates trailing blanks in received job and output entries.

**X'80' - No Blank Truncation**

This code becomes effective only for job header records - it is ignored for other record types. VSE/POWER will then spool the received job/output to the queue without truncating trailing blanks. The records are spooled as they were created by the sender.

**Note:** Do not use

```
LOAD HALFWORD R15,=X'8000'
```

to set "no blank truncation" because it will propagate the high order bit into the 2 leftmost register bytes.

If an invalid return code is passed or a delete, insert, or modify request is passed for illegal record types, the receiver task is terminated with message 1RB61, while the transmitting system puts the corresponding entry back in the queue, but in

## PNET Receiver Exit

HOLD status (disposition D becomes H, K becomes L). The receiver task may be activated again by a new transmission request.

### Accounting for the Receiver Exit Routine

The receiver account record reflects the number of records inserted or deleted by a receiver exit routine.

### Handling Failures of Exit Routine

When a failure has been detected in a receiver exit, recovery will be performed, if possible, and:

1. message 1Q2CI is issued indicating the exit failure
2. message 1Q2KI is issued indicating that VSE/POWER continues to process.

For details, refer to “Handling of Exit Failures” in *VSE/POWER Application Programming*.

### Tracing Exit Failures

When debugging logic failures of your exit routine, it may be helpful to obtain a snapshot dump of the VSE/POWER partition at a predefined processing point of your routine. Include the call of macro IPW\$IDM into the exit to have an Idump generated without terminating the system. For details, refer to “Dump of the VSE/POWER Partition at Program Request” in *VSE/POWER Administration and Operation*.

### Intercommunication Between Exit Routines

One-byte field QRUEX has been added to the Queue Record Area (QRA). This field is provided exclusively for use by VSE/POWER exit routines but is not otherwise checked or processed by VSE/POWER. It can be used to store information during the lifetime of a queue record.

When exits gain control, TCB field TCQV (use mapping macro IPW\$DTC) points to the queue record (use macro IPW\$DQR) of the queue entry being processed. The TCB can be addressed by register 11. The NETEXIT, for example, can set one or more flags in QRUEX while receiving a job. When the job starts to execute in a VSE partition, all of its created output queue entries will inherit the contents of QRUEX. An XMTEXIT or OUTEXIT can then check QRUEX in the created output entries and take appropriate action. This is an easy way to pass information from an input exit routine like JOBEXIT/NETEXIT to output exit routines like OUTEXIT/XMTEXIT. The contents of QRUEX are lost when the queue entry is transmitted to another node.

**Considerations for received jobs:** TCQV contains a valid address only *after* the job header record (JHR) has been processed by VSE/POWER. If a job is received, the contents of QRUEX will be propagated to every output queue entry that is created by executing this job, provided the following rules are observed:

- The update of QRUEX must take place when POWER has passed the record following the JHR to the exit routine.
- If a \* \$\$ JOB statement has been received, QRUEX must be updated again, because a new queue record is created for every \* \$\$ JOB statement. The update must take place when POWER has passed the record following the \* \$\$ JOB statement (or, if JECL continuation is used, following the last continuation statement of the \* \$\$ JOB statement) to the exit routine.

**Considerations for OUTPUT:** TCQV contains a valid address only *after* the job header record (JHR) has been processed by VSE/POWER. The update of QRUEX

must take place when VSE/POWER has passed the record following JHR to the exit routine. The record following the JHR is usually the data set header (DSHR). Each DSHR except the first might lead to creation of an additional queue record with different characteristics, so that QRUEX must be updated when the Receiver Exit gains control for the record after the DSHR.

#### Rules Summary:

1. Check the 4-byte pointer to the queue record area TCQV for a non-zero value before using it.
2. Update QRUEX after each
  - Job header record
  - Data set header record
  - \* \$\$ JOB statement or its last continuation

#### Coded Example

A sample receiver exit routine is shown in Appendix B, "Sample of a PNET Receiver Exit Routine," on page 85. It is also provided to you as an A-Book in PRD1.MACLIB (NETEXAMP.A).

End of Programming Interface Information

## PNET Transmitter Exit

### Programming Interface Information

In addition to the receiver exit (NETEXIT), VSE/POWER provides a transmitter exit to inspect job or output records before they are transmitted to another node.

You can write your own transmitter exit or modify an existing one to perform one of the following tasks:

- delete a single data record
- insert a single data record
- change the content of data/network control records
- modify network control records
- flush the transmission of a queue entry (job/output).

#### Loading the Exit

To load and enable a transmitter exit routine at initialization time specify the XMTEXIT operand in the POWER generation macro.

In addition, you can also load and, thereby, enable the exit routine dynamically by means of the PLOAD command while VSE/POWER is running.

You can activate or deactivate a transmitter exit routine with the PVARY PVARY command. You can display information about the exit routine with the PDISPLAY EXIT command. For more information on the PLOAD, PVARY and PDISPLAY commands as well as the XMTEXIT operand, see *VSE/POWER Administration and Operation*.

The exit routine is given control for each record to be transmitted to any node.

#### Exit Routine Processing

**Transmitter Exit Work Area:** The transmitter exit must be reentrant, because several job/output transmitter tasks may operate at the same time. To help you

## PNET Transmitter Exit

make this exit routine reentrant, VSE/POWER passes a transmitter exit *work area* (XMTEXTIT work area) to the exit routine. Such a work area is assigned to each transmitting task and can, therefore, be used to hold special variables and information needed along with the queue entry currently transmitted.

You define the size of the work area in the POWER macro, along with the transmitter exit phase name, or with the PLOAD command if a new transmitter exit routine is to be loaded when VSE/POWER is running. The maximum work area size is 65,535 bytes. In order to avoid unnecessary storage space, you should only specify the amount of storage you really need.

The work area exists as long as a job or output entry is being transmitted. With every new entry the work area is initialized to X'00' entirely. The first four bytes of the work area contain the length of the work area; this information is refreshed every time before the user-written exit routine gets control.

The address of the XMTEXTIT work area is passed on to the exit via a *parameter list*.

### Parameter List

In contrast with the receiver exit routine, no registers are used to pass over data between the exit and VSE/POWER. Instead, register 1 points to a parameter list (see Figure 6) which contains important information about the record being transmitted, for both, VSE/POWER and the exit. The layout of the parameter list is described by macro IPW\$DTX.

Table 1. Parameter List of the Transmitter Exit

Field Name	Description	Length
TEXRV	Record address	4
TEXRL	Record length	4
TEXCC	Operation code (carriage control character)	1
TEXRT	Record type TERJHR EQU X'80' Job header record TERJTR EQU X'40' Job trailer record TERDSHR EQU X'20' Data set header record TERNCD EQU X'00' Data record or printer control record	1
TEXDT	Data/printer control record type TEDJRNC EQU X'00' not defined TEDCPDS EQU X'10' CPDS record TEDASA EQU X'02' ASA record TEDLPCM EQU X'01' Line print/card move record	1
TEXOT	Type of data stream TEOLST EQU X'80' List output TEOPUN EQU X'40' Punch output TEOJOB EQU X'20' Job data	1
TEXWA	Address of the exit's work area	4
TEXRC	Return code from exit TEROK EQU X'00' Process record TERDEL EQU X'04' Delete record TERINS EQU X'08' Insert record TERFLS EQU X'10' Flush queue entry TERMOD EQU X'14' Process modified network control record TERFLSH EQU X'18' Flush queue entry with HOLD	1

Field TEXDT of the parameter list specifies the data record or printer control record indicated by TEXRT = X'00' in more detail.

The fields of the parameter list set by VSE/POWER and the user are shown in the following table.

Table 2. Fields of the Parameter List Set by VSE/POWER and the User

Field	VSE/ POWER	Return Specifications Indicated by the Exit				
		Normal Processing	Deletion	Flush/ Flush Hold	Insertion	Modification
TEXRV	P				X	X
TEXRL	P				X	X
TEXCC	P				X	
TEXRT	P				X	
TEXDT	P				X	
TEXOT	P					
TEXWA	P					
TEXRC		X	X	X	X	X

**Note:** Fields denoted by 'P' are set by VSE/POWER before the transmitter exit gets control. Fields denoted by 'X' must be set by the user.

### Programming Considerations

When coding your routine, observe the following:

- In your exit routine, use the DSECTs as listed in the example on page 95 to address fields within the network control records. The layouts of these records may change.
- The exit routine must not alter any fields in any control blocks to which it has access.
- The exit routine must not alter the content of register 10, 11 and 12. These registers are reserved for VSE/POWER use only. All other registers can be used by the exit routine.
- The exit routine should not invoke VSE/Advanced Functions services (SVCs) which may wait, either voluntarily or involuntarily, for an event or resource. Instead, the exit must use the VSE/POWER IPW\$WFC macro to wait for a VSE/POWER event. A VSE wait from the exit routine would stop all the VSE/POWER task processing, including any devices, such as readers, printers and remote workstations under its control.
- Transmission control records (such as ML records) are not passed on to the exit routine.
- One byte X'FE' records, which are records that are for VSE/POWER internal use only, are always translated to X'03' records, that is to 'no operation records' (NOP).
- The exit routine must *not* use any of the following addressing modes:
  - 31-bit addressing mode
  - Access-register mode.

When the exit routine loses control (for example, due to a page fault), the status of the currently used addressing mode and the access registers are not saved. Thus, when the exit routine gets control back again, the previously used addressing mode and the access registers cannot be restored.

## PNET Transmitter Exit

Because the exit routine must not use the access-register mode, the exit routine cannot use data spaces.

- VSE/POWER does not accept an exit routine which has already been loaded into the SVA.

### Register Content on Entry

When the routine receives control, the contents of certain registers are as follows:

Register	Content
1	Address of the parameter list
14	Return address to VSE/POWER
15	Entry point address of the exit

The exit routine has access to all control blocks available in the environment, for example:

Register	Content
10	Common Address Table
11	Transmitter Task Control Block

### Return Specifications

The exit routine must return control to VSE/POWER with one of the following return codes set in the field TEXRC of the parameter list.

#### X'00' - Process

Process record passed to the transmitter exit. Fields within the record may be changed, but not address or length of the record. Message 1RA9I with RC=0008 is issued if record address or record length was changed.

#### X'04' - Delete

Delete this record. Job header, job trailer and data set header records can **not** be deleted. Message 1RA9I with RC=0009 is issued to indicate this failure.

#### X'08' - Insert

Insert and process a new record. VSE/POWER will return the original record to the exit routine once more.

The address, the length and the operation code (for output data) must be provided in the according fields of the parameter list. A record with record length or record address of X'00' is rejected. Message 1RA9I with RC=000B is issued. This message is also issued when the inserted record is a job type record and its length is greater than the maximum data record length. The length of the data records is up to 80/128 bytes for job data and up to 32,767 bytes for output data.

If the exit tries to insert a record before a job header record, message 1RA9I with RC=000D is issued. The insertion would destroy the NJE transmission protocol.

Job header, job trailer and data set header record can **not** be inserted. Message 1RA9I with RC=000A is issued if the user-written exit routine inserts a record which is not allowed.

#### X'10' - Flush

Flush XMT queue entry. Transmission of the affected queue entry is interrupted and the queue entry is deleted if its original disposition was D. If its original disposition was K, it is set to L.



**X'14' - Modify**

Process network control records whose length or address has been changed. This is only valid for job header, data set header or job trailer records. The current header or trailer record has been changed by the exit routine. This could mean that a user section has been inserted by the exit in the record.

Message 1RA9I with RC=000C is issued if the current record is not a network control record (job header record, job trailer record or data set header record).

**X'18' - Flush**

Flush XMT queue entry with HOLD option. Transmission of the affected queue entry is interrupted and the queue entry is retained in the XMT queue in HOLD status; a queue entry with disposition D is queued again with disposition H, a queue entry with disposition K is queued again with disposition L.

For any other return code received from the transmitter exit (not X'00', X'04', X'08', X'10', X'14', X'18') message 1RA9I with RC=0007 is issued. In all above cases where message 1RA9I is issued, the transmission of the queue entry is interrupted and the queue entry is queued again with disposition 'HOLD' (disposition D→H, K→L). The transmitter continues with the next available queue entry that is eligible for transmission.

**Accounting for the Transmitter Exit Routine**

The transmitter account record reflects the number of records inserted or deleted by a transmitter exit routine.

**Handling Failures of Exit Routine**

When a failure has been detected in a transmitter exit, recovery will be performed, if possible, and:

1. message 1Q2CI is issued indicating the exit failure
2. message 1Q2KI is issued indicating that VSE/POWER continues to process.

For details, refer to "Handling of Exit Failures" in *VSE/POWER Application Programming*.

**Tracing Exit Failures**

When debugging logic failures of your exit routine, it may be helpful to obtain a snapshot dump of the VSE/POWER partition at a predefined processing point of your routine. Include the call of macro IPW\$IDM into the exit to have an Idump generated without terminating the system. For details, refer to "Dump of the VSE/POWER Partition at Program Request" in *VSE/POWER Administration and Operation*.

**Intercommunication Between Exit Routines**

See "Intercommunication Between Exit Routines" on page 32.

**Coded Example**

A sample transmitter exit routine is shown in Appendix C, "Sample of a PNET Transmitter Exit Routine," on page 95. It is also provided to you as an A-Book in PRD1.MACLIB (XMTEXAMP.A).

End of Programming Interface Information

### Command Authorization

The following figures give an overview of command authorization for operators of remote nodes.

Figure 6 on page 39 shows which commands an operator of another node may use on the local node, having been given the available level of authority. This table is for the central operator of a remote node and for a (master-password) authorized Spool-Access CTL user of a remote node.

Figure 7 on page 41 shows which commands a remote (RJE) operator, an unauthorized Spool-Access CTL user, or any cross-partition CTLSPOOL user connected to another node can use on the local node.

#### Command Authorization for Remote Nodes

The access of operators and users of other nodes to VSE/POWER running in the local node is controlled by three levels of authority:

**NET** This is the highest level of authority which another node can be given. It entitles the operator of the other node to perform many functions on your node without being the owner of the jobs or output.

**JOB** This is the default authority which another node receives if nothing else is specified in the NDT. It allows the central operator of the other node to manipulate only those jobs for which he is the originator or, in the case of output, only that output which is destined for his node or which was generated by a job submitted by his node.

**NOJOB**

This is the lowest level of authority which another node may be given. The central operator of the other node has no right to manipulate anything in the queues of your node. He may only issue certain display commands.

The authority level that another node is to be given is specified in the entry for that node in your own NDT. If the other node is a VSE/POWER node, the PXMIT command may be used to have different commands routed to your own node.

Commands coming from a remote (RJE) user, an unauthorized Spool-Access CTL user, or any cross-partition CTLSPOOL user get the same authority as the originating node, up to a JOB authority level. For a VSE/POWER node, only commands that the user can issue on his local node are allowed.

**Authorization Table for Central Operator of Remote Node or (Master-Password) Authorized Spool-Access CTL User of the Remote Node**

Authorization	NET	JOB	NOJOB
PACCOUNT	NO	NO	NO
PACT	YES	NO	NO
PALTER job	YES	YES (1)	NO
PALTER partition	NO	NO	NO
PBRDCST	YES	YES	YES
PCANCEL status	YES	YES	YES
PCANCEL job	YES (1)	YES (1)	NO
PDELETE job	YES	YES (1)	NO
PDELETE MSG	YES	NO	NO
PDELETE FCB	NO	NO	NO
PDISPLAY job	YES	YES	NO
PDISPLAY MSG	YES	YES	YES
PDISPLAY A	YES	YES	NO
PDISPLAY M	NO	NO	NO
PDISPLAY Q	NO	NO	NO
PDISPLAY STATUS	NO	NO	NO
PDISPLAY T	YES	YES	YES
PDISPLAY DYNC	YES	YES	YES
PDISPLAY PNET	YES	YES	YES
PDISPLAY TAPE	NO	NO	NO
PDISPLAY VIO/QFL	NO	NO	NO
PDISPLAY TRINFO	YES	YES	YES
PDISPLAY EXIT	YES	YES	NO
PDISPLAY TASKS	YES	YES	NO
PDISPLAY SPDEV(T)	YES	YES	NO
PDISPLAY AUSTMT	YES	YES	NO
PDRAIN	YES	NO	NO
PEND	NO	NO	NO
PFLUSH partition	NO	NO	NO
PFLUSH PNET task	YES	NO	NO
PFLUSH task	NO	NO	NO
PGO	NO	NO	NO
PGO DEV	YES	NO	NO
PHOLD job	YES	YES (1)	NO
PINQUIRE	YES	YES	YES

*Figure 6. Authorization Table for Central Operator of Remote Node or (Master-Password) Authorized Spool-Access CTL User of the Remote Node (Part 1 of 2)*

## Command Authorization

Authorization	NET	JOB	NOJOB
PLOAD PNET	YES	NO	NO
PLOAD exit	NO	NO	NO
PLOAD DYNC	NO	NO	NO
POFFLOAD	NO	NO	NO
PRELEASE job	YES	YES (1)	NO
PRESET	NO	NO	NO
PRESTART	NO	NO	NO
PRESTART DEV	YES	NO	NO
PSEGMENT partition	YES	YES(1)	NO
PSETUP	NO	NO	NO
PSETUP DEV	YES	NO	NO
PSTART partition	NO	NO	NO
PSTART RJE	YES	NO	NO
PSTART PNET	YES	NO	NO
PSTART task	NO	NO	NO
PSTART DEV	YES	NO	NO
PSTART trace	NO	NO	NO
PSTOP cuu	NO	NO	NO
PSTOP partition	NO	NO	NO
PSTOP RJE	YES	NO	NO
PSTOP PNET	YES	NO	NO
PSTOP SAS	YES	NO	NO
PSTOP DEV	YES	NO	NO
PSTOP trace	NO	NO	NO
PSTOP TCPIP	YES	NO	NO
PSTOP TCPSSL	YES	NO	NO
PVARY	NO	NO	NO
PXMIT	NO	NO	NO
PXMIT DEV	YES	NO	NO
Note: (1) Only for jobs originated by the remote operator.			

Figure 6. Authorization Table for Central Operator of Remote Node or (Master-Password) Authorized Spool-Access CTL User of the Remote Node (Part 2 of 2)

**Authorization Table for Remote (RJE) Users, Spool-Access CTL Users, or Cross-Partition CTLSPOOL Users**

Node Authorization	NET/JOB		NOJOB	
	Remote User	XPART User	Remote User	XPART User
PACCOUNT	NO	NO	NO	NO
PACT	NO	NO	NO	NO
PALTER job	YES (1)	YES (1)	NO	NO
PALTER partition	NO	NO	NO	NO
PBRDCST	YES	YES	YES	YES
PCANCEL status	–	YES	–	YES
PCANCEL job	YES (1)	YES (1)	NO	NO
PDELETE job	YES (1)	YES (1)	NO	NO
PDELETE MSG	NO	NO	NO	NO
PDELETE FCB	NO	NO	NO	NO
PDISPLAY job	YES	YES	YES	NO
PDISPLAY MSG	YES	YES	YES	YES
PDISPLAY A	YES	YES	NO	NO
PDISPLAY M	NO	NO	NO	NO
PDISPLAY Q	NO	NO	NO	NO
PDISPLAY STATUS	NO	NO	NO	NO
PDISPLAY T	YES	YES	YES	YES
PDISPLAY DYNC	YES	YES	YES	YES
PDISPLAY PNET	YES	YES	YES	YES
PDISPLAY VIO/QFL	NO	NO	NO	NO
PDISPLAY TRINFO	YES	YES	YES	YES
PDISPLAY EXIT	YES	YES	NO	NO
PDISPLAY TASKS	YES	YES	NO	NO
PDISPLAY SPDEV(T)	YES	YES	NO	NO
PDISPLAY AUSTMT	YES	YES	NO	NO
PDRAIN	NO	NO	NO	NO
PEND	NO	NO	NO	NO
PFLUSH partition	NO	NO	NO	NO
PFLUSH PNET task	NO	NO	NO	NO
PFLUSH task	NO	NO	NO	NO
PGO	NO	NO	NO	NO
PGO DEV	NO	NO	NO	NO
PHOLD job	YES (1)	YES (1)	NO	NO
PINQUIRE	YES	YES	YES	YES

Figure 7. Authorization Table for Remote (RJE) Users, Spool-Access CTL Users, or Cross-Partition CTLSPOOL Users (Part 1 of 2)

Node Authorization	NET/JOB		NOJOB	
	Remote User	XPART User	Remote User	XPART User
PLOAD PNET	NO	NO	NO	NO
PLOAD exit	NO	NO	NO	NO
PLOAD DYNC	NO	NO	NO	NO
POFFLOAD	NO	NO	NO	NO
PRELEASE job	YES (1)	YES (1)	NO	NO
PRESET	NO	NO	NO	NO
PRESTART	NO	NO	NO	NO
PRESTART DEV	NO	NO	NO	NO
PSEGMENT partition	YES(1)	YES(1)	NO	NO
PSETUP	NO	NO	NO	NO
PSETUP DEV	NO	NO	NO	NO
PSTART partition	NO	NO	NO	NO
PSTART RJE	NO	NO	NO	NO
PSTART PNET	NO	NO	NO	NO
PSTART task	NO	NO	NO	NO
PSTART DEV	NO	NO	NO	NO
PSTART trace	NO	NO	NO	NO
PSTOP cuu	NO	NO	NO	NO
PSTOP partition	NO	NO	NO	NO
PSTOP RJE	NO	NO	NO	NO
PSTOP PNET	NO	NO	NO	NO
PSTOP SAS	NO	NO	NO	NO
PSTOP DEV	NO	NO	NO	NO
PSTOP trace	NO	NO	NO	NO
PSTOP TCP/IP	NO	NO	NO	NO
PSTOP TCPSSL	NO	NO	NO	NO
PVARY	NO	NO	NO	NO
PXMIT	NO	NO	NO	NO
(1) Only for jobs or output originated by, or destined for, the cross-partition user.				

Figure 7. Authorization Table for Remote (RJE) Users, Spool-Access CTL Users, or Cross-Partition CTLSPPOOL Users (Part 2 of 2)

## Providing Data Security

### VSE/POWER Protection of Data

The retrieving and manipulating of queue entries can be controlled by use of the node ID and user ID of jobs or output. Before a user is allowed to retrieve output from the VSE/POWER queues, the user ID given with the request must match the user ID of the job or output.

The information stated above is particularly important on remote nodes as the password, which may also be specified for a job or output entered by a cross-partition user, cannot be used when manipulating the job or output at a remote node.

### VSE/POWER Interface to the z/VSE Access Control Function

The SEC operand of the \* \$\$ JOB statement allows a job to *enter* into a VSE/POWER 5.2 node (or later versions) to access z/VSE protected resources without using the // ID statement. When specifying the SEC operand, take the following into account:

- The values of the SEC operand in the \* \$\$ JOB statement to specify the VSE security userid and password
- The z/VSE Access Control generation operands which define security to the z/VSE system
- The VSE/POWER SECNODE operand in the POWER generation macro.

**Note:** One of the ways a job can lose its security authorization at execution time is if both of the following hold true:

- The job executes using inherited VSE security authorization from a "parent" job, instead of specifying an explicit security userid and password, *and*
- The executing CPU of a target node does not have the same "security zone" (SECNODE value) as the "parent" job.

For information on the SEC and SECNODE operands, refer to *VSE/POWER Administration and Operation*. For information on the VSE Access Control, refer to *z/VSE Administration*.

## Automatic Restart of Nodes

Once sign-on is complete and provided the NR operand was not specified on the PSTART command, automatic restart is performed for a node in the following cases:

1. If the connection was broken due to timeout limit exceeded (BSC)
2. If a switched BSC line times out during the dialing process
3. If the session was terminated for the node (SNA)
4. If the other node issues a PSTOP command for the connection to your node. This may mean that communication is to be terminated for the rest of the VSE/POWER session, or it may mean a temporary shutdown of communications due to some problem on the other node (a shortage of disk storage, for example). If it is known that communication will not be restarted during the current VSE/POWER session, a PSTOP command should be issued for the node; this frees the resources which are reserved by the pending connection.
5. If a NAK was received (CTCA)
6. If I/O completes with a unit exception (which should not occur) or a unit check (CTCA).
7. If the return code and ERRNO returned by TCP/IP for VSE/ESA allow a restart (TCP and SSL nodes).

If any of the above cases occurs, VSE/POWER attempts to restart the connection and displays:

- Message "1RB8I AUTOMATIC RESTART OF NODE ..."

## Automatic Restart of Nodes

- Message “1RC6I CONNECTION PENDING FOR NODE ...” if the other node is not active or cannot re-activate its connection. In this case, only the default transmitters and receivers are set to the non-drained status.

If the remote node restarts, the message “1RB8I NODE node-id HAS RESTARTED” is displayed. All transmitters and receivers that were in a non-drained status when the restart occurred are set to non-drained again. No operator intervention is necessary to activate required transmitters or to drain unwanted receivers.

Any jobs or output, which were being transmitted when the restart occurred, are automatically returned to the queue in a dispatchable status. The jobs/output are eligible to be transmitted again as soon as the connection is established. All jobs/output which were being received at the time of the restart are purged by the receiving system.

## RJE with PNET

Input to the network is allowed from an RJE terminal. The RJE user may submit a job to run on any other node within the network by specifying XDEST on the \* \$\$ JOB statement. The output from the run is returned to the RJE station, unless a JECL operand is provided to override this default assignment. It is also possible to transmit the output of the job to any other node and remote station within the network.

Output may be directed to any remote station within the network by use of the LDEST, PDEST, or DEST operands in the JECL statements if the execution node is a VSE/POWER node. If the node is a non-VSE node, the correct JCL must be used to ensure the non-default routing. If, however, LDEST or PDEST is used in the \* \$\$ JOB statement, the default routing that is used by the executing node is updated with the specified values, and no operating system-specific JCL is required.

Figure 7 on page 41 shows the commands an RJE operator may issue on a remote VSE/POWER node depending on the authority of his node. The commands which an operator may issue on non-VSE nodes have to be determined by reference to the books relevant for the networking system being used.

---

## Administrative Considerations

### Network Considerations

Before initializing PNET as part of a network, the following conditions must be fulfilled at the installations which will participate in the network.

- All installations in a network must agree upon node names.
- Output classes must be coordinated so that print and punch classes are consistent among nodes in the network.
- Installations must agree upon the use of line and node passwords, when and how they may be changed, and how they are protected.
- Installation managers must decide on the command authority the participating nodes are to have over their computing center; a policy should be established for handling the alteration of that authority.
- Receiver and transmitter exits must tolerate the different records which may flow through a network.
- All installations in an SNA NJE network must agree upon unique application names to be used in defining every PNET application to VTAM.



## Changing the Name of the Local Node During Warmstart

The local node name is defined by the network definition table (NDT), which has been generated by the PNODE macro using the operand LOCAL=YES for the local node name. To change the local node name, an NDT with the **new** local node name can be specified during a VSE/POWER warm start:

1. either via the PNET operand of the POWER macro which is used to generate the POWER phase
2. or via the autostart statement SET PNET, which is interpreted during the initialization of VSE/POWER.

When an NDT with a new local node name is to be loaded, the operator is informed about the node name change by the following messages:

```
1RE5I NETWORK DEFINITION TABLE ndtname FOUND WITH NEW LOCAL NODE NAME
      nodeid
1RE6D CONFIRM CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 BY
      'YES', ELSE 'NO'
```

When *YES* is entered, VSE/POWER issues the following messages:

```
1RB4I PLOAD NETWORK DEFINITION TABLE ndtname LOADED
1RECI STATUS REPORT $LSTNNNN BEING CREATED DUE TO NODE NAME CHANGE
1Q8GI STATUS REPORT DISPLAYED IN LIST ENTRY $LSTnnnn
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 IN PROGRESS
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 COMPLETED
```

Message 1RB4I is issued to confirm the loading of the NDT. Message 1RE7I with the ending words 'IN PROGRESS' is issued, when VSE/POWER starts changing the node name within the first queue entry. Message 1RE7I with the ending words 'COMPLETED' is issued, when VSE/POWER has processed the last queue entry. Message 1RE7I with the ending words 'IN PROGRESS' is repeated every 30 seconds until the node name change has completed.

Before the change of the local node is started, the PDISPLAY ALL,LST is issued to create a status report of all queue entries. The report is a list queue entry with the name \$LSTnnnn, where nnnn are the last four digits of the assigned jobnumber. This report may be helpful, if the local node change fails, which should never occur (see "Process Interruption of Node Name Change" on page 48.). If the node name change has been completed successfully (reported by message 1RE7I), the status report may be deleted in the list queue.

When the local node name is changed, the following occurs:

1. the name of the local PNET node within the internal master record is changed
2. the local node name in all queue entries is changed, which means the node name is changed in the
  - a. queue record
  - b. job header record
  - c. data set header record

The following local node name occurrences may be changed in these records:

- a. the origin node name
- b. the execution node name
- c. the target node name
- d. the target node name for the list output
- e. the target node name for the punch output
3. queue entries are dequeued from the transmission queue and enqueued to the local reader, list or punch queue with its local disposition, if the old target node name is equal to the new local node name.

## Administrative Considerations

### Notes on Usage

- The system ID, which can be specified by the SYSID operand of the POWER macro, remains unchanged for all queue entries.
- The security zone, which can be specified by the SECNODE operand of the POWER macro, remains unchanged for all queue entries.
- Whenever a node name change has to be rejected during a warm start, the network definition table (NDT) is not loaded and hence the network function is not made available. The following messages are issued:

```
1RA5I PLOAD INVALID NETWORK DEFINITION TABLE ndtname, RC=rrrr
1Q08I UNABLE TO INITIALIZE NETWORKING FUNCTION, RC=0004
```

RC=rrrr within message 1RA5I describes one of the various reasons, why the NDT could not be loaded. For example, RC=0011 is displayed, if NO is entered in reply to message 1RE6D. In order to initialize the PNET function, VSE/POWER must be restarted using an NDT with the old node name.

- The node name change is rejected, if a network definition table (NDT) with a new local node name has been specified
  - in a PLOAD command after VSE/POWER has been completed its initialization.
  - in an autostart command PLOAD PNET,ndtname, which is actually processed after the initialization warm start has been complete

Then following message is issued:

```
1RA5I PLOAD INVALID NETWORK DEFINITION TABLE ndtname, RC=0004
```

### Internal Restrictions

- If a queue entry contains more than one data set header record, only the first data set header record is updated.
- If a job header or data set header record contains a section with a layout unknown to VSE/POWER, the section is not updated with the new local node name. For example following sections are not updated:
  1. a security section
  2. an OPTB section
  3. a user defined section

### Handling of I/O Errors when Processing Data Records

Due to the update of the:

1. master record
2. queue record
3. job header record
4. data set header record

the following errors may occur:

1. a read I/O error,
2. a write I/O error or
3. a logic error:
  - a. an invalid length value within the various control sections of a job header or data set header record was found
  - b. no job header or data set header record was found at all.

If any such error occurs, messages are issued to report the problem and the initialization of VSE/POWER continues, for example:

- if an I/O error occurs during writing of the master record or a queue record, the queue file is rebuilt by formatting the queue file on disk and writing out the storage copy of the queue file. The following messages will appear:

```
1Q63I PERM I/O ERROR WRITING/READING QUEUE FILE MASTER RECORD
1QF5I QUEUE FILE IS BEING RE-BUILT
1QF6I QUEUE FILE SUCCESSFULLY RE-BUILT
```

or

```
1Q61I WRITE I/O ERROR ON cuu , CKD=.....
1QF5I QUEUE FILE IS BEING RE-BUILT
1QF6I QUEUE FILE SUCCESSFULLY RE-BUILT
```

- If a job header or data set header record can not be read or written, the queue entry is set in a hold disposition ('H' or 'L'). The following is a sample of messages, where the reason code rrrr can be a value between 0011 and 0014:

```
1Q61I READ I/O ERROR ON DFILE 02 151, CKD=01BE000001, BLOCK= 0
1RE9I LST QUEUE ENTRY HUS01 00002 KEPT WITH HOLD DISPOSITION, RC=rrrr
```

- If a logic error occurs when updating a job header or data set header record, the queue entry is set in a hold disposition ('H' or 'L'). The following is a sample of a message, where the reason code rrrr can be a value between 0001 and 0010, or between 0015 and 0019:

```
1RE9I RDR QUEUE ENTRY HUS02 0003 KEPT WITH HOLD DISPOSITION, RC=rrrr
```

### Recovery from Abend during Node Name Change

If VSE/POWER has terminated abnormally during the change of the local node name, VSE/POWER resumes changing the local node name at a subsequent warm start and informs the operator by the messages:

```
1RB4I PLOAD NETWORK DEFINITION TABLE ndtname LOADED
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 RE-INITIATED
```

After the standard queue file recovery has been completed, the change of the local node name is resumed:

```
1QB7I FULL QUEUE FILE RECOVERY IN PROGRESS
1QB8I QUEUE FILE RECOVERY COMPLETED
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 IN PROGRESS
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 COMPLETED
```

**Note:** While VSE/POWER completes the change of the local node name to nodeid2, it rejects another change to nodeid3 at the same time. Consider the following scenario: VSE/POWER with node name nodeid1 has been terminated abnormally during the change of the local node name to nodeid2. Thereafter VSE/POWER is started with a network definition table ndt3 introducing a local node name nodeid3 different to nodeid2. VSE/POWER re-initiates changing the local node name to nodeid2 and completes the warm start, but does not activate network definition table ndt3 and does not initialize the PNET function. Following messages are issued:

```
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 RE-INITIATED
1RA5I PLOAD INVALID NETWORK DEFINITION TABLE ndtname, RC=0010
1Q08I UNABLE TO INITIALIZE NETWORKING FUNCTION, RC=0004
1QB7I FULL QUEUE FILE RECOVERY IN PROGRESS
1QB8I QUEUE FILE RECOVERY COMPLETED
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 IN PROGRESS
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 COMPLETED
```

Note also that the same procedure occurs, if nodeid3 is equal to the original node name nodeid1. If the networking function is to be initialized, VSE/POWER must be restarted using an NDT with node name nodeid2.

## Administrative Considerations

### Process Interruption of Node Name Change

If VSE/POWER has been terminated **abnormally twice** during the process of changing the local node name, VSE/POWER does no longer resume the node name change at a subsequent warm start, but informs the operator as follows:

```
1RE7I  CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 INTERRUPTED
1RB4I  PLOAD NETWORK DEFINITION TABLE ndtname LOADED
1QB7I  FULL QUEUE FILE RECOVERY IN PROGRESS
1QB8I  QUEUE FILE RECOVERY COMPLETED
1REAI  CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 INCOMPLETE,
        NOT ALL QUEUE ENTRIES PROCESSED
```

The easiest way to then complete the node name change, is to update all queue entries with the new local node name by following steps:

1. save all queue entries to tape (using the POFFLOAD BACKUP command for example)
2. delete all queue entries on spool file (using the PDELETE command, but to delete all queue entries, do not have any queue entries in 'active state' DISP='\*')
3. re-load all queue entries from tape to the spool file (using the POFFLOAD LOAD command with the operand **oldnode** and optionally NOJNO).

The number of queue entries to be offloaded can be reduced by using the information of message 1REBI and the status report with the name \$LSTnnnn, which has been created by VSE/POWER before starting the node name change (see messages 1RECI and 1Q8GI). Message 1REBI displays the last queue entry, which has been processed successfully. Only those queue entries in the status report \$LSTnnnn have to be offloaded which are displayed after the queue entry reported in message 1REBI. Message 1REBI has been issued after message 1Q2CI at the time VSE/POWER terminated previously, for example:

```
1Q2CI  PSW=070D000000000002 ILC=1, CC=20 OPERATION EXCEPTION
        A/T=OFF/WN TASK=I IT, (00501040)
1REBI  LAST QUEUE ENTRY PROCESSED SUCCESSFULLY BY NODE NAME CHANGE:
        XMT HUSL03 02865 3 L H L
```

The information about the last queue entry processed successfully has the same format as used by the PDISPLAY command:

```
queue jobname jobnumber priority disposition class
and optionally:
sysid, original queue and S=jobsuffix
```

If VSE/POWER is up and running after having reported an incomplete node name change by message 1REAI and a PDISPLAY command is issued, the **not yet processed** queue entries are presented in the following way due to VSE/POWER's recovery:

1. queue entries, which were queued in the local RDR, LST and PUN queue before the node name change started, are now queued in the transmission (XMT) queue. The target node name (displayed after TO=) is the original local node name.
2. queue entries, which were queued in the transmission queue before the node name change started and whose destination node name was equal to the new local node name, are now queued in the local RDR, LST or PUN queue.

### Considerations for Shared Spooling

If VSE/POWER shares the queue and data file with other systems (operand SHARED=Q has been used in the POWER macro), special considerations insure data integrity. That means the local node name can be changed only when **no**

other system of the shared spooling complex is running. Otherwise the change of the node name is rejected, the network definition table is not loaded, and hence the networking function is not initialized. The following messages are issued (if, for example, the systems with the SYSID 2, 4 and 9 are still active):

```
1RE8I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 FAILED
      - ACTIVE SYSID(S) FOUND: 2,4,9
1RA5I PLOAD INVALID NETWORK DEFINITION TABLE ndtname, RC=0012
1Q08I UNABLE TO INITIALIZE NETWORKING FUNCTION, RC=0004
```

No information about the attempt to change the node name to nodeid2 is recorded internally. That means, when VSE/POWER is restarted, the original local node name nodeid1 remains unchanged. To perform the node name change in the above sample, you must first shutdown the systems with the SYSID 2,4, and 9 and the re-warm start the own system using the network definition table with the new local node name.

### Recovery from Shared System Abend during Node Name Change

If VSE/POWER is terminated abnormally (for example on SYSID 2) during the change of the local node name, VSE/POWER continues to change the local node name (to nodeid2) at the next start up of the **failing** system or any **other** system. The following messages are issued:

```
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 RE-INITIATED
1RB4I PLOAD NETWORK DEFINITION TABLE ndtname LOADED
1QB7I FULL QUEUE FILE RECOVERY IN PROGRESS FOR SYSID 2
1QB8I QUEUE FILE RECOVERY COMPLETED
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 IN PROGRESS
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 COMPLETED
```

If the starting system specifies an NDT with another node name than the new one (nodeid2), VSE/POWER continues to change the local node name to nodeid2 and completes the warm start, but does not initialize the networking function. The following messages are issued:

```
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 RE-INITIATED
1RA5I PLOAD INVALID NETWORK DEFINITION TABLE ndtname, RC=0010
1Q08I UNABLE TO INITIALIZE NETWORKING FUNCTION, RC=0004
1QB7I FULL QUEUE FILE RECOVERY IN PROGRESS FOR SYSID 2
1QB8I QUEUE FILE RECOVERY COMPLETED
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 IN PROGRESS
1RE7I CHANGE OF LOCAL NODE NAME FROM nodeid1 TO nodeid2 COMPLETED
```

If the networking function should be initialized, VSE/POWER must then be restarted using an NDT with node name nodeid2, or even with a nodeid3, which results in another change of the local node name.

## VTAM Considerations

SNA PNET application-to-application sessions are controlled by VTAM. Therefore, certain VTAM requirements must be met before a session can be established between two VSE/POWER PNET nodes, or between VSE/POWER PNET and any other supported networking system. Besides the requirements outlined in "VTAM Requirements" on page 11, the following requirements must be considered:

- VTAM must have the application names defined for the node being initialized, as well as for all nodes which are eligible for application-to-application sessions with other nodes in the network.
- To open the application interface from VSE/POWER to VTAM successfully, the corresponding VTAM sublibrary must be included in the phase search chain of the VSE/POWER partition.

## Administrative Considerations

- The VTAM network must be activated. This includes activation of the IBM 37xx NCPs (if no ICA is used) and SDLC links, and of all participating VTAM cross-domain resource managers. If VTAM is not available when the first PSTART command for an SNA node is given, VSE/POWER retries to open the VTAM interface at regular intervals until either a retry is successful or the request is deleted by means of a PSTOP command for the node.

When the VTAM session requirements are met, a session can be started with another eligible node by issuing a PSTART command with the appropriate operands specified (see *VSE/POWER Administration and Operation* for the correct format).

The VTAM RUSIZES (request unit) operand in the MODEENT macro should be specified as 0 for the PNET logmode table entry. This removes the restriction on the maximum RU size contained in the BIND operands. PNET requests VTAM to send RUs of maximal PNET buffer size to the other node (where PIUs are slightly larger than the PNET buffer size).

As long as a PNET buffer size between 300 and 2000 bytes is used (default is 1112 bytes), there is no problem with the other VTAM domain accepting that size of PIU if the defaults for MAXBFRU (contained in the VTAM LINE macro) and of the VTAM LFBUF buffer size are used. When increasing the PNET buffer size to its maximum of 32000 bytes, then the VTAM MAXBFRU operand and/or the LFBUF buffer size specification must be adjusted accordingly together with the corresponding NCP generation parameters UNITSZ and TRANSFR, provided an NCP is required.

UNITSZ and LFBUF buffersize must be the same for VTAM. Thus for VTAM, the product of MAXBFRU and UNITSZ/LFBUF buffer size must be larger than the PNET BUFSIZE parameter.

### Effect of VTAM Commands on PNET Nodes

Certain VTAM commands have an effect on PNET nodes communicating via SNA sessions. The most common VTAM commands and their actions are:

#### Z NET,QUICK

All SNA nodes are immediately terminated with message 1RB0I.

#### Z NET

All SNA nodes are terminated when all jobs or output currently being transmitted between the nodes have completed.

#### V NET,INACT or V NET,TERM

Both commands have the effect that the session between nodes is terminated. These commands may be useful when you cannot stop a node with the PSTOP command.

## TCP/IP Considerations

Besides the requirements outlined in "TCP/IP for VSE/ESA Setup and Interface Requirements" in *VSE/POWER Administration and Operation* and "TCP/IP Requirements" on page 11, there are no additional considerations. The MTU=maximum transmission unit size of the TCP/IP DEFINE LINK command applies *only* to the hardware characteristics of the addressed link type (e.g., Ethernet, Token Ring, or CTCA). There is *no* relation of the MTU to the PNET buffer size as specified by the BUFSIZE= operand of the VSE/POWER PNODE macro.

### Effect of TCP/IP Commands on PNET Nodes

Certain TCP/IP Operator Communication Exit commands have an effect on PNET nodes communicating via PNET TCP:

#### STOP LINKID=link-add-for-node-communication

The affected node is flagged with message 1RTQI, and the connection to this node is terminated with 1RB0I RC=0009, indicating "link error".

#### SHUTDOWN

All TCP and SSL nodes are terminated as described for STOP LINKID. The interface from VSE/POWER to the TCP/IP partition, namely the TD and SD subtasks, is also terminated with message 1RT8I. After the TCP/IP connection has been terminated, the TD and SD subtasks try to reestablish the TCP/IP connection by first issuing the 'SOCKET' socket call, which may lead to the following message

```
1RTJA TCP/IP: INITIALIZATION OF INTERFACE UNSUCCESSFUL, RETRY GOING ON
```

You can use the PINQUIRE NODE=own-nodeid command to track the status of VSE/POWER's interface to TCP/IP.

## Performance Considerations

The resources that PNET requires are:

- Processor storage (virtual and PFIXED storage)
- VSE/POWER partition Getvis storage related to TCP/IP support.
- Teleprocessing Line capacity
- Processing speed
- VTAM fixed and pageable buffers (for SNA sessions only)

You can influence the availability of these resources when you are planning the layout of the system. If the default values specified by the various macros (and the examples in this book) are used, PNET uses a relatively small amount of the resources. This should be sufficient for the initial installation and operation of PNET when network use is still quite low.

For PNET SNA, altering the performance characteristics for PNET requires a coordinated change of PNET and VTAM performance parameters. If you wish to improve your system's PNET performance, or if your system's resources are already severely restricted, make changes to the defaults or the use of the following parameters and resources:

- VTAM pacing
- Number and size of PNET buffers
- Number of VTAM buffers
- Number of PNET transmitters and receivers which are concurrently active.

For PNET TCP and PNET SSL, precautions have been taken to yield good performance:

- The TD and SD subtasks driving the SEND/RECEIVE requests in the VSE/POWER partition operate as a parallel workunit, so that the non-parallel system share is not increased for the Turbo Dispatcher.
- The standard number of buffers, namely 2 per transmitter and receiver of a TCP or SSL node, is extended by one additional one (altogether 5 buffers), which serves I/O read-ahead operations.
- The PNET buffer size can be extended up to 32,000 bytes, so that the XPCC-based exchange traffic of SEND/RECEIVE buffers between the VSE/POWER and TCP/IP partitions can be reduced.

## Administrative Considerations

When changing these parameters and resources, keep in mind that PNET is a teleprocessing application; here, response times are less critical than for an online application. In online applications (for example, a typical CICS transaction), a user waits for a response at a terminal. Therefore, PNET should not be allowed to acquire large amounts of resources; the default values given for the resources were set to achieve this.

This is a case where you might give the CICS or VTAM partition a higher priority than the VSE/POWER partition. See the NPC operand of the PSTART command.

### Transmitter Throughput

The specification of the numbers of PNET transmitters per node can affect throughput and performance. PNET CTC and PNET BSC need a minimum time of approx. 2.5 seconds per transmitted file, if only a single transmitter is used. This is caused by NJE-required request steps that have to be exchanged at the start and end of every job/output file to be transmitted. The following table gives an example of how the number of transmitters influences the performance for a PNET CTC line under VM. Forty small outputs were transmitted between two idling systems (no other activity), 5 tests were performed for each number of transmitters (1-6) and the average total transmission time is shown.

No. of transmitters	1	2	3	4	5	6
Time (seconds)	96	24.6	7.2	3.0	3.8	3.0

### VTAM Pacing

The specification of the VTAM pacing parameters affects the performance of PNET when SNA sessions are used for networking. By specifying a low value (we recommend a value of 3), use of both processor and real storage are kept at a minimum. Increasing the pacing value makes sense only if:

- The line used for the PNET session is not used to full capacity, and other applications which may be using the line do not have critical performance requirements.
- Both of the connected processors have enough processor storage available.

Increasing the pacing values require larger LFBUF and VPBUF specifications for VTAM. The combination of these values determines how many VTAM buffers are devoted to PNET requirements.

The pacing parameters must be specified in two places:

- In the VTAM APPL statement defining the local PNET application
- In the VTAM MODEENT macro for the appropriate BIND parameters.

See "Specifying Pacing" on page 83.

### Defining the Buffer Size

You can specify the size of the buffer your node is to use for communicating. For a BSC or CTCA line, the buffer size may vary between 300 and a maximum of 4,000 bytes. For an SNA link or a TCP or SSL node, the maximum buffer size is 32,000 bytes. To specify the buffer size, use the BUFSIZE operand in the PNODE generation macro. If you omit the operand, VSE/POWER uses a default buffer size of 4096 bytes for a TCP or SSL node, otherwise 1112 bytes for all other link types; for details on the PNODE macro see *VSE/POWER Administration and Operation*.



The buffer size to be used for the communication is agreed upon between the two communicating nodes during the sign-on process. If the values specified in the sign-on records are not equal, the smaller of the two values is used; the other system then adjusts to this value.

The larger the specified buffer size, the smaller the number of I/O movements (BSC and CTCA) or SEND/RECEIVE requests (SNA, TCP, and SSL) are required to transmit a given amount of data between two nodes. To support the larger buffers, it is necessary to have more storage available. In the case of a BSC or CTCA connection, this storage is the SETPFIX LIMIT storage. For an SNA link or a TCP or SSL node, the storage used for buffers comes from the partition GETVIS space. If storage is a problem, the buffers should be defined at their lowest allowed value, and the extra I/O's required will have to be accepted.

The storage used for the buffer pools is dynamically retrieved and, as soon as the transmitter or receiver task is finished, this storage is returned to the general VSE/POWER storage pool and can be used by other tasks.

### Number of Buffers

The number of buffers to be allocated to every transmitter and receiver task communicating with a particular node is defined in the MAXBUF operand of the PNODE macro. The default value of this operand for BSC, CTCA, and SNA links is three receiver and three transmitter buffers per activated task. Note that you may improve performance by assigning more buffers for a particular node. For example:

- If the transmission line speed between two nodes is high. Here, it may be of advantage to specify a larger number of receiver buffers on the node which will receive the mass of the transmissions.
- If large volume transmissions occur only in one direction. For example, a job transmitted from A to C consisting of a few statements, but the output returned from C to A consists of several hundred pages of report.

For SNA nodes, the following approximate relationship may be used to determine the MAXBUF number of transmitter and receiver buffers:

#### No. of transmitter buffers $\geq$

2 times the pacing value (outbound) defined by the VPACING operand in a VTAM B-BOOK.

#### No. of receiver buffers =

They are defined in the adjacent node and there should be at least as many as defined for the transmitter in the connected node.

For TCP and SSL nodes, only 2 receive and 2 transmitter buffers are allowed per task. Any other specification will result in a compiler MNOTE warning.

### Number of Transmitters and Receivers

PNET can use multiple transmitters when in communication with certain networking systems. The transmitters are all multiplexed on the same transmission line. Because of the multiplexed connection, and if the transmission line is already a "bottle neck" in the system, you cannot improve the overall transmission rate by simply activating more transmitters.

It is sometimes useful to activate further transmitters so that, when one transmitter is fully occupied sending large volume output, other smaller jobs or output have a chance to get transmitted via other transmitters.

### Network Operation with Shared Spooling

Any z/VSE system with shared spooling must have the same node name defined for all systems of the sharing complex. If an SNA session is to be used, a unique name must be specified for the APPLID operand for each of the sharing systems.

The following points apply when one of the sharing systems does not have PNET installed:

- The PNET specific parameters for the \* \$\$ JOB, \* \$\$ LST, and \* \$\$ PUN statements may be used under the following conditions:
  - The different sharing nodes require different application IDs if they are running PNET in an SNA session.
  - A destination node specified in a JECL statement is not checked for being known to the network, because the non-PNET system has no NDT.
- Jobs or output put into the spool files for transmission by the other sharing system does not activate a transmitter on the sharing system. This means that the job or output remains in the VSE/POWER spool files until a transmitter for the required node is activated from the system including PNET. This could happen automatically if something is added to the spool file on the PNET system which requires transmitting to the same node, or by operator intervention.

If PNET is installed on all sharing systems, all PNET functions are available with the following limitations:

- The operator of a system running the VSE/POWER Shared Spooling function can only communicate with or via systems directly linked to his system.
- Messages destined for the central operator, or for remote or cross-partition users of a remote node, can generally be received if the remote node is directly linked to the system which has issued the message.
- If more than 60 nodes are signed on at the same time, the jobs or output you put into the spool files for transmission by a sharing system do not automatically activate a transmitter on that system.

If any of the shared systems is running with the VSE Access Control function activated, VSE/POWER assigns that system a security "zone" through the POWER generation macro SECNODE operand. Any jobs arriving at the node that are using inherited VSE security authorization from a "parent" job (for example because an explicit security userid and password are not specified) will execute according to the following rules:

- The job will be selected to execute on a system in the same security zone (SECNODE operand) as that of the parent job unless a specific shared system (SYSID operand) was specified.
- If no specific shared system is required and no system has a matching SECNODE value, the job can be selected by any system for execution.
- In either of the above cases, if the executing system does not have a matching SECNODE value, the job will lose its security authorization at execution time.

### Change of a Job's Characteristics

This section describes what happens to the attributes of a job that is received via the network and still contains VSE/POWER \* \$\$ JOB statements.

The following gives a more detailed explanation of the different processing rules.

### Conversion of a \* \$\$ JOB Statement at Read-in Time

The attributes of a VSE/POWER job are defined by the values specified in a \* \$\$ JOB statement. When a job is sent via a network to a non-VSE/POWER node, the receiving node may not recognize the syntax and values of the VSE/POWER \* \$\$ JOB statement.

In order to pass a job's attributes to other systems, control records have been defined which are to be used by systems participating in an NJE-network. (The layout of these control records are described in *Network Job Entry Format and Protocols* ).

One of these control records is the job header control record. It contains all the attributes defined in the \* \$\$ JOB statement.

Normally, VSE/POWER processes a \* \$\$ JOB statement and moves the specified values into a job header control record when the job is passed to VSE/POWER for the first time. This happens if a job is read in 'locally' in one of the following ways:

- By a local reader
- By an RJE workstation
- From a tape
- Passed by a spool-access user.

Therefore, if VSE/POWER receives a job via the network from a VSE/POWER node, usually VSE/POWER does not receive a \* \$\$ JOB statement but only the job header control record.

### Conversion of a \* \$\$ JOB Statement at Receive-Time

In addition to the job header control record, a VSE/POWER job may also contain a \* \$\$ JOB statement if that job has been either

- Received from a non-VSE/POWER node via a network, or
- Produced using the DISP=I operand on the \* \$\$ JOB statement.

In these cases, the \* \$\$ JOB statement is processed when it is received via the network. This happens only if the first record is a \* \$\$ JOB statement and the receiving node is the final destination.

### Processing of a \* \$\$ JOB Statement

When a \* \$\$ JOB statement is processed at receive-time, all operands of the \* \$\$ JOB statement are processed and used to update the job's characteristics.

When an operand of the \* \$\$ JOB statement is not specified or has been specified incorrectly, a default value is assigned to the operand. This default value depends on where the job is received from.

1. When the job is received from a non-VSE node, some default values are retrieved from the job header control record and some default values are chosen by VSE/POWER (see the \* \$\$ JOB statement in *VSE/POWER Administration and Operation* for the default values). This is because there exist operands which are not known to the system running on the non-VSE node, and which are therefore not contained within the job header control record.

- a. For the following operands, the job is assigned a default value as received from the sending system:

**JNM** Job name

**BLDG** The programmer's building number

## Processing of a \* \$\$ JOB Statement

**DEPT** The programmer's department number

**FROM**

The user who submitted the job

**LDEST**

The node-id (and user-id) where to route the list output

**NETACCT**

The network account number

**PDEST**

The node-id (and user-id) where to route the punch output

**PRI** The priority

**PROGR**

The programmer's name

**PWD** A password

**ROOM**

The programmer's room number

**UINF** User information

**XDEST**

The node-id (and user-id) where to route the job for processing

Note that the sending system generally uses other keywords than the above described VSE/POWER operands.

b. All operands not listed above are unknown to a non-VSE system.

Consequently, for operands that are not listed above, the job is assigned the default value as described in *VSE/POWER Administration and Operation*.

2. When the job is received from a VSE node, all default values are those received from the sending system. Note, however, that such a job has been produced by using the DISP=I operand in the \* \$\$ PUN statement and that this job has inherited some default values from its producing job. For details, see the DISP=I description of the \* \$\$ PUN statement in *VSE/POWER Administration and Operation*.

### Note:

- Whenever an operand has been specified incorrectly, all its successive operands are ignored and the default values are taken.
- If a syntax error has been found, the job is placed in the hold state (H). Error messages are sent to:
  1. The local console operator and
  2. To the originator of the job or to the user who has been specified in the NTFY operand or the corresponding operand of a non-VSE system.

Part of the error message is the incorrect \* \$\$ JOB statement. If it contains the SEC= or PWD= parameter with a correct value length, the value will be masked with asterisks.

- If the receiving system is a backlevel system, operands known to the sending system may not be known to the receiving system and may, therefore, result in syntax errors.

### Processing Several \* \$\$ JOB Statements

Because a network-job<sup>1</sup> may contain more than one \* \$\$ JOB statement, VSE/POWER writes one VSE/POWER job to the spool file for each \* \$\$ JOB statement which is found within the network-job. This means that one network-job may result in several VSE/POWER jobs on the VSE/POWER spool file.

Only after the entire network-job has been received successfully, its VSE/POWER jobs are added to the spool file. Thus, any recovery problems are avoided when receiving of a network-job is terminated abnormally. In this way, the sender can resubmit the entire network-job without being concerned about the earlier network job.

#### Notes:

1. Different VSE/POWER jobs of one network-job do not necessarily have consecutive job numbers.
2. If the job name is invalid or missing on more than one \* \$\$ JOB statement, more than one VSE/POWER job may have the same job name (of the original network job, for example, RSCSnnnn).
3. If several \* \$\$ JOB statements are found within one network-job, each of these will result in one VSE/POWER job, no matter if for each of these jobs an \* \$\$ EOJ statement has been specified or not.
4. If an \* \$\$ EOJ statement is found, which is not immediately followed by a \* \$\$ JOB statement, the rest of the network-job is received as one VSE/POWER job which is assigned all the values of the original network-job (for example, jobname=RSCSnnnn, disposition=D, etc.).
5. The network exit routine may be called for more than one VSE/POWER job within one network-job.

### Non-Processing of the \* \$\$ JOB Statement

If the first record of the network-job is not a \* \$\$ JOB statement, or the next statement after a \* \$\$ EOJ statement is not a \* \$\$ JOB statement, the \* \$\$ JOB statement is not processed at the time the job is received. The \* \$\$ JOB statement is written as is to the spool file and is processed at the time the job is executed.

As long as the job is not executed, the characteristics of the job are those as received within the job header control record sent by the other system.

The \* \$\$ JOB statement is processed at execution time and some characteristics may change due to the specified values within the \* \$\$ JOB statement. For more details, see *VSE/POWER Administration and Operation*.

**Note:** If at execution time some characteristics have been changed, these changed characteristics may affect the job only as long as the job remains on the execution node. If the executed job is sent to another node, the job has its original characteristics at the receiving node. The characteristics changed due to the processed \* \$\$ JOB statement are not sent to the new node.

### Accounting

The receiver account record uses the original job name (for example, RSCSnnnn) in field NACNAME, although the received job is assigned a different name, namely

---

1. The term 'network-job' means a job received via the network according to the NJE network protocol (the network-job starts with a job header control record and ends with a job trailer record).

## Processing of a \* \$\$ JOB Statement

the one specified in the \* \$\$ JOB statement. The account information concerning the number of records for the original network-job and the VSE/POWER job spooled on disk may differ in the following way:

- The data record count within the receiver account record is updated according to the number of received records, that is, the \* \$\$ JOB statement (and any \* \$\$ JOB continuation statement) and the \* \$\$ EOJ statement are included in the accounting.
- The data record count of the VSE/POWER job on the spool file is updated according to the number of spooled records, that is, the \* \$\$ JOB statement (and any \* \$\$ JOB continuation statement) and the \* \$\$ EOJ statement are not included in the accounting.

## Connections to a Backlevel VSE/POWER System

If you send a job to a backlevel system the values of the operands of the \* \$\$ JOB statement which are not known to the backlevel receiving system are ignored by the receiving system.

The unknown values are not ignored, however, if the job has been produced with the DISP=I operand on the \* \$\$ JOB statement. In this case, the receiving backlevel system validates all operands of the \* \$\$ JOB statement and issues error messages for the unknown operands.

This is because of the following processing procedures:

A VSE/POWER job statement as seen by the user usually looks as follows:

```
* $$ JOB  
  . . . . . (other statements)  
* $$ EOJ
```

This indicates that the first statement of the job is a \* \$\$ JOB statement and the last statement is a \* \$\$ EOJ statement. In between any other type of statement may be included.

When this job is passed to VSE/POWER, VSE/POWER, in most cases, processes the \* \$\$ JOB statement and builds a network control record, called the job header control record. This record contains the values specified in the \* \$\$ JOB statement and is written to the VSE/POWER spool file, whereas the original \* \$\$ JOB statement is not written to the spool file and is no longer available.

As a result, when this job is sent to another system, the receiving system does not receive a \* \$\$ JOB statement, but only the network control record.

Whenever a new operand is introduced for the \* \$\$ JOB statement, the layout of the network control record is extended in order to contain the values for the newly defined operand. Therefore, if the receiving system is a backlevel system, the receiving system does not know any of the new operands nor any of the new fields within the network control record. The receiving system ignores the new values without any error or warning message and processes the job according to the values known to it.

The translation of a \* \$\$ JOB statement into a network control record usually takes place when, for example, the job is passed to VSE/POWER in one of the following ways.

1. The job is read in by a local reader.
2. The job is read in by an RJE workstation.

3. The job is read in from a tape.
4. The job is passed by a spool-access user.

The translation of a \* \$\$ JOB statement into a network control record does, however, not take place when, for example, the job that is passed to VSE/POWER has been produced by using the operand DISP=I in a \* \$\$ PUN statement.

In this case, the receiving system translates the \* \$\$ JOB statement into a network control record, and issues error messages if the \* \$\$ JOB statement contains operands or values for operands not yet known to it. The error messages (for example 1Q37I, 1R36I, and 1R37I) are issued on the operator console of the receiving system and to the originator of the job.

## Connections to RSCS

PNET can communicate with an external VM/RSCS via one of the RSCS NJE line drivers. The following points should be noted when communicating with RSCS:

- To make full use of the restart possibilities, a restart EXEC should be provided in RSCS for the node communicating with VSE/POWER PNET.
- When sending a job from a CMS user via RSCS to a PNET node, you must use the JOB option of the TAG command:

```
TAG DEV PUN node JOB
```

Otherwise, the job will enter the punch queue of the receiving PNET node.

- For commands given by a CMS user of a VM/RSCS node, the command authority of Figure 7 on page 41 will become effective. For more details, please refer to "Command Authorization" on page 38.
- Since RSCS uses only four digits for an FCB name, VSE/POWER adds, at printing time, a prefix according to the device type used. The prefixes used are described in *VSE/POWER Administration and Operation* under the FCB operand of the \* \$\$ LST JECL statement.
- To make use of the VM/CP distribution code specified for local printing or punching to VM/CP (DIST=... specified on \* \$\$ LST/PUN), use the new SET statement

```
SET RSCSR00M=DIST
```

to pass the DIST value now in field NJHGROOM via RSCS to VM/CP. Refer to *VSE/POWER Administration and Operation*.

For more information on RSCS, see *VM RSCS Networking V3R2 Planning and Installation*, SH24-5219.

## Connections to JES2 NJE

When creating a network containing JES2 NJE nodes, consider this:

For network definition and initialization:

- The connection used between PNET and JES2 must be defined to JES2 as a "predefined connection", using a CONNECT statement in the JES2 initialization deck.

The CONNECT statement must be specified with MEMBB=1, regardless of any system ID set by VSE/POWER. PNET appears as a 'single system complex' whether it runs on a shared VSE/POWER system or not.

For sending jobs to JES2 for running:

## Connections

- Any disposition specified in the VSE/POWER \* \$\$ JOB statement has an influence only on the final destination node for the output. Therefore, it is not possible to send a job to JES2 which is immediately put in a "HOLD" status before running, but it is possible to receive output from JES2 which is automatically placed in "HOLD" status.
- The \* \$\$ JOB statement can be used to specify default destinations for the print and punch output.

For receiving output that was produced on a JES2 node:

- Any output produced on a JES2 node and routed to a system with VSE/POWER is placed into the list or punch queue in the same class as used for the output while the job is running on the JES2 node.
- Since JES2 uses only four digits for an FCB name, VSE/POWER adds, at printing time, a prefix according to the device type used. The prefixes used are described in *VSE/POWER Administration and Operation* under the FCB operand of the \* \$\$ LST JECL statement.

**Note:** JES2 suppresses leading zeros in the remote ID. If, for example, a job is submitted from remote ID R005 and sent to JES2 for running, JES2 converts the remote ID into R5 and sends the output back. But R5 is not recognized as remote ID.

For sending output to JES2:

- JES2 recognizes the difference between print and punch output only by means of the class specification. If punch output for class B is produced and sent to JES2, but JES2 has class B as a list device, then the punch data is printed.
- Starting with JES2 version 4.1.: When sending output from VSE/POWER to JES2 for processing locally, you must specify the following for:

**\* \$\$ LST and/or \* \$\$ PUN statement**

DEST=(node-id,LOCAL)

**\* \$\$ JOB statement**

PDEST=(node-id,LOCAL), or LDEST=(node-id,LOCAL)

If output is sent to a JES2 node using the PALTER command, the user-id should be set to LOCAL.

### Multitrunk Support

JES2 offers "multitrunk support", which means several connections can be started between two nodes. This support is not offered by VSE/POWER for a single node but may be achieved by using the VSE/POWER shared spooling function for several z/VSE systems, grouped together as a system complex.

The shared spooling function allows up to nine z/VSE systems using VSE/POWER to share one single VSE/POWER spool file. For all z/VSE systems a network definition table (NDT) can be used which contains the same local nodeid and the same JES2 nodeid. On each z/VSE system the PSTART PNET,nodeid command may be used to start a connection to the JES2 node. Thus up to nine connections may be active at one time to the same JES2 node.

## Connections to JES3 NJE

When creating a network that includes JES3 NJE nodes, and when communicating with JES3 via a BSC line, you need to consider the following points:



For network definition and initialization:

- JES3 does not support the exchange of line and node passwords at connection start time. Therefore, do not specify a password in the applicable PLINE macro nor in the PSTART command, when starting the connection to the JES3 system.
- JES3 does not negotiate on the transmission buffer size. If JES3 receives a buffer size smaller than the own buffer size, it restarts the connection. Therefore, the buffer size specified in the PNODE macro must be identical to the JES3 buffer size.
- JES3 supports only one JOB and one OUTPUT receiver. If you issue a PACT command to activate a second transmitter, JES3 refuses permission for this second transmitter, and PNET sets this transmitter into drained status again.

For sending jobs from VSE/POWER to JES3 for running:

- The disposition specified in the VSE/POWER JECL statement \* \$\$ JOB has no influence on the JES3 system. Therefore, if you send a job to JES3, and if the job is to be in a "HOLD" status before running, include the JES3 statement `/*MAIN HOLD=YES` in the job stream.
- The class and priority specified in the VSE/POWER JECL statement \* \$\$ JOB are ignored by JES3. The specifications are replaced by the JES3 defaults.
- The VSE/POWER JECL statement \* \$\$ JOB can be used to specify the default destination for print and punch output.

For sending jobs from JES3 to VSE/POWER for running:

- Make use of the `//XMIT` JCL statement which allows to specify both a destination and a delimiter operand so that all statements between the `//XMIT` and the user-defined delimiter are sent to the execution node.

Example:

```
//JOB JOB PW19,'DEPT 53'
//X2 XMIT DEST=PWR,DLM=AA
      :
      :
(VSE/POWER job to be transmitted)
      :
      :
AA
//
```

For receiving output that was produced on a JES3 system:

- Punch output is placed into the punch queue with class 'B'. Print output is placed in the list queue with the same class as used for the output while the job is running on the JES3 system.
- Since JES3 uses only four digits for an FCB name, VSE/POWER adds, at printing time, a prefix according to the device type used. The prefixes used are described in *VSE/POWER Administration and Operation* under the FCB operand of the \* \$\$ LST JECL statement.

For sending output from VSE/POWER to JES3 for further processing:

- When output is produced on a VSE/POWER system and the output is routed to a JES3 system for processing on a local device, the target 'user-id' must be coded as 'LOCAL':  
\* \$\$ LST ...,DEST=(node-id,LOCAL)

## Connections

---

## Chapter 3. Diagnostic Aids

Problems may occur during the operation of the network. VSE/POWER provides the PNET line trace and console trace facilities to help diagnose problems. Also the various system messages should be noted.

---

### PNET Trace in Telecommunication Trace Area

This trace is also referred to as the PNET line trace because it must be enabled at PSTART time for a PNET 'line'. When errors occur on a certain BSC or CTCA line, initiate the trace with the following, for example:

```
PSTART PNET,NODEB,,010,,TRACE
```

An entry is then written to the in-core telecommunication trace area for each complete BSC or CTC line write/read request.

When failures occur on a certain SNA, TCP, or SSL connection, initiate the trace with the following, for example:

```
PSTART PNET,NODEX,,,,TRACE
```

An entry is written to the in-core telecommunication trace area for every

- completed SNA session SEND or RECEIVE request
- started, tested, completed TCP/IP socket call

For details on this trace and the layout of the trace entries, see Appendix E, "PNET Telecommunication Trace," on page 117.

---

### PNET Console Trace

PNET provides a console trace for BSC, CTCA, and TCP/IP connections. All I/O activity taking place on these connections is logged on the VSE operator's console. This function should be used only if serious problems occur and when other console traffic is not hindered by the logging process. To reduce the console traffic, the PNET console trace can also be directed to a printer device addressed as SYSLST.

The operator can request trace logging with the PSTART CNSLTR command. To terminate the console trace, PSTOP CNSLTR must be entered. For details on this trace, see Appendix D, "PNET BSC/CTCA/TCP/SSL Console Trace," on page 107.

---

### PNET TCP and PNET SSL System Messages

The various system messages should be noted. In addition to VSE/POWER messages, further diagnostic messages concerning TCP/IP connections may occur from:

- the VSE LE/C component (messages beginning with "CEExxxxx" or "EDCxxxxx") due to the EZASMI API interface used by VSE/POWER
- the TCP/IP component (messages beginning with "BSDxxxxx" or "SSLxxxxx", for example) due to the TD and SD subtasks used by EZASMI

## Problem Determination

Also, both EZASMI and TCP/IP for VSE/ESA offer their own tracing functions, which may produce console or SYSLST printer messages. For details, see Appendix D, "PNET BSC/CTCA/TCP/SSL Console Trace," on page 107.

---

## Problem Isolation in VSE/POWER Networking

The goal of problem isolation is to find the source of the failure. If one function or subcomponent fails and passes incorrect information to other components, those components may also fail. Before attempting in-depth problem analysis, collect all the possible information.

### Common Problems

The following lists commonly occurring problems. The explanations (symptom/causes/action) do not cover all possibilities; they are meant primarily as an aid to finding the cause of common problems.

#### 1. VSE/POWER networking does not initialize.

##### Symptom:

The VSE/POWER network function does not indicate successful initialization via message 1RB4I.

##### Possible causes:

- a) PNET= was not specified in the POWER generation that was used for the initialization.
- b) The NDT specified either in the PNET= operand of the POWER generation, or in the SET PNET operand of AUTOSTART, could not be found.
- c) A serious error was found in the NDT.

##### Action:

This may depend on any error message which has been issued on the console during the initialization phase of VSE/POWER. It may require a new VSE/POWER generation to be assembled and cataloged, or might mean that the NDT must be cataloged.

#### 2. Connection to other node does not complete.

##### Symptom:

No indication that connection is established (message 1RB3I), but message 1RC6I that connection is pending was issued.

##### Possible causes:

- a) The other node did not start connection or, if it is a non-PNET system, then the appropriate start commands for the network were not issued.
- b) The processor of the other node is not yet active.
- c) There is insufficient storage to activate the connection on the own system.
- d) The cross-domain resource manager (CDRM) is not active.
- e) The path is not activated.
- f) The major/minor SNA node is not properly activated.
- g) Incorrect specification of node name. For example, node A has specified a false node name for node B in the PSTART command.

##### Action:

Check the system console log for error messages and take corrective action. If SNA sessions are used, check that all required VTAM resources are activated.

3. Connection start causes error.

**Symptom:**

Sign-on is not completed (message 1RB3I not received).

**Possible causes:**

- a) Either the own operating system or the other node's operating system is overloaded and cannot respond quickly enough.
- b) The amount of real and/or GETVIS space allocated to the VSE/POWER partition is inadequate.
- c) VTAM on the own system is not active (only applicable for connections using SNA links).
- d) VSE/POWER has too low a priority and cannot acknowledge responses quickly enough.
- e) The transmission line has a high error rate.

**Action:**

The action depends on the cause determined above.

4. Other node rejects sign-on.

**Symptom:**

Connection is stopped due to a request from the other node (message 1RB0I), and/or the connection goes into restart mode (message 1RB8I).

**Possible causes:**

- a) The node password is omitted or does not match.
- b) The line password is omitted or does not match.
- c) An invalid buffer size is specified. The minimum that is allowed is 300 and the maximum is 4,000 bytes for a BSC or CTCA line or 32,000 bytes for an SNA or TCP link. The other node may check the value and reduce it to a value acceptable to that system.

**Action:**

Check with the operator of the other node to see whether he received any console messages during the attempt to start the connection.

5. Connection cannot be terminated.

**Symptom:**

A PSTOP command for a node or a PEND command is issued, but the node has never been signed off (message 1RB0I). If the connection is via an SNA link, then message 1RC4I may be issued.

**Possible causes:**

- a) VSE/POWER cannot write the network account record due to a full account file.
- b) VSE/POWER is short on DASD space and the receiver cannot write the received data set to disk (message 1Q38I).
- c) Available real and/or GETVIS space is insufficient.
- d) The other node terminated already without being able to communicate the fact.
- e) VTAM is in a "short on storage" condition.

**Action:**

- Check the system console log for error/warning messages and take corrective action.
- If the problem occurred with an SNA link, enter the VARY NET,INACT,ID=applid command in an attempt to get VTAM to deactivate the session.

## Problem Determination

- If the problem was caused by a shortage of disk space, use the POFFLOAD command to off-load some queue entries to tape for future processing.

### 6. Jobs or output appear to loop.

#### Symptom:

Jobs or output with the same original job number show up frequently in the XMT queue and are logged as being received.

#### Possible causes:

Incorrectly defined NDT. The NDT may have defined an incorrect ROUTE1 or ROUTE2 for a node. If alternate routing is used, extra care should be taken to ensure that loops are not defined within the network.

#### Action:

Check the network definitions for all nodes within the network and, especially, any alternate routings which might have been defined.

### 7. Transmitter starts but is automatically drained.

#### Symptom:

A transmitter is activated by means of a PACT command but is immediately drained again (message 1RA8I). When a job or output is made available in the XMT queue, the transmitter attempting to transmit this data is drained.

#### Possible causes:

- a) The other node is shutting down.
- b) The corresponding receiver on the other node is drained. For every active transmitter, the corresponding receiver must be available on the receiving node.

**Note:** If communicating with JES3, only JOB and OUT transmitters 1 are supported; an attempt to start any other transmitter causes the transmitter to be drained immediately.

#### Action:

If the error occurs when a further transmitter is activated, check with the operator of the connected node, whether he is able to support the additional receiver.

### 8. Transmitter does not start.

#### Symptom:

Job or output is in the XMT queue destined for the node for which a connection exists, but the data is not transmitted.

#### Possible causes:

- a) The transmitter is set to the drained status.
- b) The queue entry is in the HOLD state.
- c) The transmitter is waiting for confirmation from the receiving system, but this confirmation has never arrived.

#### Action:

Check the transmitters by a "PINQUIRE node-id" command and, if there is no transmitter in a "non-drained" status, use the PACT command to activate a transmitter of the correct type.

If a transmitter is available in an "inactive" status, check that the queue entry is in disposition D or K.

If the transmitter is still active, but has zero records to transmit, and this condition remains for several minutes, issue a PSTOP for the node, and then restart the connection.

**9. No response on commands.**

**Symptom:**

A command destined for another node was entered at the own node, but no reply messages are received and displayed.

**Possible causes:**

- a) Your node name is unknown at the node where the command was executed.
- b) The path to the target node was broken while the command was being transmitted.
- c) No path was available from the executing node to your node when the replies from the command were ready to be transmitted. This could be due to a line failure.
- d) The system was overloaded (no GETVIS space to hold a copy of the response message that is to be sent to the originating node).

**Action:**

Issue the command again. If there is still no reply, check with the operator of the executing system to ensure that your node is known to that node.

**10. VTAM error messages received.**

**Symptom:**

Some or all of the following messages may be received on a node using PNET SNA lines for network functions:

```
1RC0I    BUFFER(S) LOST ON LINK ... RC=0003 or RC=0004
1RD7I    LOSTERM EXIT SCHEDULED ... REASON LOST CODE=36
1RD4I    VTAM SETLOGON QUIESCE ... RC/FDB2=08,00
1RD5I    VTAM CLOSE FAILED ... RC=0088
1RD8I    VTAM OPNDST FAILED ... RC/FDB2=14/4B
IST1020I INSUFFICIENT STORAGE... FULL
```

**Possible causes:**

Probably the pacing operand was not specified for the APPLID in question, or there is a mismatch between the buffers acquired by PNET (too large a pacing value may have been specified) and the buffer space available in VTAM.

**Action:**

Check the specified pacing values and the VTAM buffer pool size. If message

```
IST1020I INSUFFICIENT STORAGE - DATA SPACE dspname FULL
```

is received, increase VSE/POWER and/or VTAM data space.

**11. CEE3322C message with USER-ABEND CODE 4082**

**Symptom:**

Some or all of the following messages may be displayed when PLOADing a network definition table containing a TCP node.

```
1RT7I    TCP/IP: INTERFACE STARTING, SOCKETCALL SOCKET ISSUED
CEE0374C CONDITION = CEE3200S TOKEN = 00030C80 59C3C5C5 00000000
        WHILE RUNNING PROGRAM SLIB_SOCKET WHICH STARTS AT 806C485
        AT THE TIME OF INTERRUPT
```

## Problem Determination

```
PSW      07CD0400 006F9868
GPR 0-3  00000002 0067EA68 80689590 806C48B0
GPR 4-7  006AC890 006BE3A4 006ABE48 00000001
GPR 8-B  0001078C FFFFFFFF FFFFFFFF 806C4850
GPR C-F  006F8CE0 0067EA00 806F8E7E 006F9868
FLT 0-2  448D760000000000 4040404040404040
FLT 4-6  4040404040404040 4040404040404040
CEE3322C EXECUTION ABNORMALLY TERMINATED WITH USER-ABEND CODE 4082
AND REASON CODE 0002.
0V15I REQUEST FROM SYSTEM SERVICE ROUTINE
0S00I SUB IPW$$TD CANCELED
0S07I PROBLEM PROGRAM PSW = 07CD0000 8020CBA8
0S30I DUMP STARTED. MEMBER=DF100000.DUMP IN SUBLIB=SYSDUMP.F1
1I51I DUMP COMPLETE
CEE3322C EXECUTION ABNORMALLY TERMINATED WITH USER-ABEND CODE 4082
AND REASON CODE 0002.
```

When the network definition table specified for the VSE/POWER startup contains one or more TCP nodes, a similar message sequence may appear on the console during initialization of VSE/POWER:

```
1Q20I AUTOSTART IN PROGRESS
1Q85I IT ,--- WAITING FOR GETVIS-24 STORAGE, 00007680 BYTES ...
1RT7I TCP/IP: INTERFACE STARTING, SOCKETCALL SOCKET ISSUED
CEE0374C CONDITION = CEE3200S TOKEN = 00030C80 59C3C5C5 00000000
        WHILE RUNNING PROGRAM SLIB_SOCKET WHICH STARTS AT 806C485
        AT THE TIME OF INTERRUPT
PSW      07CD0400 006F9868
GPR 0-3  00000002 0067EA68 80689590 806C48B0
GPR 4-7  006AC890 006BE3A4 006ABE48 00000001
GPR 8-B  0001078C FFFFFFFF FFFFFFFF 806C4850
GPR C-F  006F8CE0 0067EA00 806F8E7E 006F9868
FLT 0-2  448D760000000000 4040404040404040
FLT 4-6  4040404040404040 4040404040404040
CEE3322C EXECUTION ABNORMALLY TERMINATED WITH USER-ABEND CODE 4082
AND REASON CODE 0002.
0V15I REQUEST FROM SYSTEM SERVICE ROUTINE
0S00I SUB IPW$$TD CANCELED
0S07I PROBLEM PROGRAM PSW = 07CD0000 8020CBA8
0S30I DUMP STARTED. MEMBER=DF100000.DUMP IN SUBLIB=SYSDUMP.F1
1I51I DUMP COMPLETE
CEE3322C EXECUTION ABNORMALLY TERMINATED WITH USER-ABEND CODE 4082
AND REASON CODE 0002.
```

### Possible Causes:

The VSE/POWER partition Getvis area is too small.

### Action:

Increase the size of the partition Getvis area by increasing the ALLOC value.

## TCP Node Cannot Sign On

To get more information about the failure, enter the command

```
PSTART CNSLTR
```

which produces 1RTTI trace messages (see Appendix D, "PNET BSC/CTCA/TCP/SSL Console Trace," on page 107). To minimize the number of trace messages, use the operand 'PNETTCP' and start just the TCP node that cannot sign-on.

When the PSTART CNSLTR command is entered, the following messages are issued:



## Problem Determination

```
PSTART CNSLTR,,PNETTCP 14:24:08
1C39I COMMAND PASSED TO VSE/POWER 14:24:08
1R88I OK 14:24:08

1RTTI TD-1: IPW$$TD VERSION 07/26 (621100) RUNNING WITH IPW$$TS 14:24:08
VERSION 07/26 (5DC000) 14:24:08
1RTTI TD-2: SOK=0000 ACCEPT ,RC=+0000000,ERRNO=+0001103, 14:24:08
TS-RC=00,P 00000000 000 14:24:08
```

The trace message displaying information about the ACCEPT socket call with the character "P" after "TS-RC=00," shows that the passive connection is active and ready to process incoming CONNECT requests. If no messages are issued, make sure a network definition table (NDT) has been loaded with at least one TCP node.

### 1. 1RTTI with CONNECT and ERRNO=1127 After 2 Minutes

#### Symptom:

The following trace messages are issued when the CONNECT socket call does not complete successfully. After 2 minutes, the CONNECT socket call is terminated with ERRNO=1127 (=ETIMEDOUT), meaning that no remote node answered the CONNECT request. The connection is terminated by the CLOSE socket call. PNET tries again to establish the connection every two minutes, which produces the same messages.

```
PSTART PNET,POWER251 13:04:09
1C39I COMMAND PASSED TO VSE/POWER 13:04:09
1R88I OK 13:04:09

1RTTI LINE=TCP START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER251 13:04:09
1RTTI TD-1: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000, 13:04:09
TS-RC=00,A POWER251 08208000 0000 800000 13:04:09
1RC6I CONNECTION PENDING FOR NODE POWER251, TIME=13:04:09 13:04:09
1RTTI TD-3: SOK=0000 SOCKET ,RC=+00000001,ERRNO=+0000000, 13:04:09
TS-RC=00,A POWER251 08208000 0000 000000 13:04:09
1RTTI TD-1: SOK=0001 CONNECT ,RC=+0000003,E 13:04:09
TS-RC=00,A 009.001.001.001 0C208000 0000 800000 13:04:09
1RTTI TD-2: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0001103, 13:04:09
TS-RC=00,A POWER251 0C208000 0000 800000 13:04:09

.....
1RTTI TD-3: SOK=0001 CONNECT ,RC=-0000001,ERRNO=+0001127, 13:06:10
TS-RC=04,A POWER251 0C208000 0000 000000 13:06:10
1RTTI TD-1: SOK=0001 CLOSE ,RC=-0000001,ERRNO=+0001127, 13:06:10
TS-RC=00,A POWER251 2C308000 0800 800000 13:06:10
1RTTI TD-3: SOK=0001 CLOSE ,RC=+00000000,ERRNO=+0000000, 13:06:10
TS-RC=00,A POWER251 2C308000 0800 000000 13:06:10
```

The above message set shows only the important messages. In reality, more trace messages are issued in between:

- 1RTTI for the CONNECT socket call with ERRNO=1103 (=EINPROGRESS), is repeated to indicate that the connection is still being processed.
- 1RTTI for an ACCEPT socket call with ERRNO=1103 is issued to indicate that incoming CONNECT requests can be processed by a passive connection.

The following message set shows all messages.

```
1RTTI LINE=TCP START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER251 13:04:09
1RTTI TD-1: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000, 13:04:09
TS-RC=00,A POWER251 08208000 0000 800000 13:04:09
1RC6I CONNECTION PENDING FOR NODE POWER251, TIME=13:04:09 13:04:09
1RTTI TD-3: SOK=0000 SOCKET ,RC=+00000001,ERRNO=+0000000, 13:04:09
TS-RC=00,A POWER251 08208000 0000 000000 13:04:09
```

## Problem Determination

```
1RTTI TD-1: SOK=0001 CONNECT ,RC=+0000003,E 13:04:09
TS-RC=00,A 009.001.001.001 0C208000 0000 800000 13:04:09
1RTTI TD-2: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0001103, 13:04:09
TS-RC=00,A POWER251 0C208000 0000 800000 13:04:09
1RTTI TD-2: SOK=0000 ACCEPT ,RC=+0000000,ERRNO=+0001103, 13:04:24
TS-RC=00,P 00000000 0000 800000 13:04:24
1RTTI TD-2: SOK=0000 ACCEPT ,RC=+0000000,ERRNO=+0001103, 13:04:24
TS-RC=00,P 00000000 0000 800000 13:04:24
1RTTI TD-2: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0001103, 13:04:59
TS-RC=00,A POWER251 0C208000 0000 800000 13:04:59
1RTTI TD-2: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0001103, 13:04:59
TS-RC=00,A POWER251 0C208000 0000 800000 13:04:59
1RTTI TD-2: SOK=0000 ACCEPT ,RC=+0000000,ERRNO=+0001103, 13:05:14
TS-RC=00,P 00000000 0000 800000 13:05:14
1RTTI TD-2: SOK=0000 ACCEPT ,RC=+0000000,ERRNO=+0001103, 13:05:14
TS-RC=00,P 00000000 0000 800000 13:05:14
1RTTI TD-2: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0001103, 13:05:49
TS-RC=00,A POWER251 0C208000 0000 800000 13:05:49
1RTTI TD-2: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0001103, 13:05:49
TS-RC=00,A POWER251 0C208000 0000 800000 13:05:49
1RTTI TD-2: SOK=0000 ACCEPT ,RC=+0000000,ERRNO=+0001103, 13:06:04
TS-RC=00,P 00000000 0000 800000 13:06:04
1RTTI TD-2: SOK=0000 ACCEPT ,RC=+0000000,ERRNO=+0001103, 13:06:04
TS-RC=00,P 00000000 0000 800000 13:06:04
1RTTI TD-3: SOK=0001 CONNECT ,RC=-0000001,ERRNO=+0001127, 13:06:10
TS-RC=04,A POWER251 0C208000 0000 000000 13:06:10
1RTTI TD-1: SOK=0001 CLOSE ,RC=-0000001,ERRNO=+0001127, 13:06:10
TS-RC=00,A POWER251 2C308000 0800 800000 13:06:10
1RTTI TD-3: SOK=0001 CLOSE ,RC=+0000000,ERRNO=+0000000, 13:06:10
TS-RC=00,A POWER251 2C308000 0800 000000 13:06:10
```

### Possible causes:

- The IP address is wrong.
- The remote system is not running.
- The remote system is running, but TCP/IP is not running (TCP/IP for VSE/ESA in the case of a z/VSE system, usually running in partition F7).

### Action:

- Correct the IP address and load the corrected NDT, if necessary.
- Start the remote system.
- Start TCP/IP on the remote system (TCP/IP for VSE/ESA in the case of a z/VSE system, usually running in partition F7).

## 2. 1RTTI with CONNECT and ERRNO=1124 Immediately

### Symptom:

The following trace messages are issued when the CONNECT socket call completes immediately and unsuccessfully with ERRNO=1124 (=ENOTCONN). The connection is terminated by the CLOSE socket call. PNET tries again to establish the connection every two minutes.

```
PSTART PNET,POWER251 14:24:14
1C39I COMMAND PASSED TO VSE/POWER 14:24:14

1RTTI LINE=TCP START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER251 14:24:14
1RTTI TD-1: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000, 14:24:14
TS-RC=00,A POWER251 08208000 0000 800000 14:24:14
1RC6I CONNECTION PENDING FOR NODE POWER251, TIME=14:24:14 14:24:14
1RTTI TD-3: SOK=0000 SOCKET ,RC=+0000001,ERRNO=+0000000, 14:24:14
TS-RC=00,A POWER251 08208000 0000 000000 14:24:14
1RTTI TD-1: SOK=0001 CONNECT ,RC=+0000001,ERRNO=+0000000, 14:24:14
TS-RC=00,A 009.001.001.001 0C208000 0000 800000 14:24:14
1RTTI TD-2: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0001103, 14:24:14
TS-RC=00,A POWER251 0C208000 0000 800000 14:24:14
1RTTI TD-3: SOK=0001 CONNECT ,RC=-0000001,ERRNO=+0001124, 14:24:15
```

## Problem Determination

```
TS-RC=08,A POWER251      0C208000 0000 00000      14:24:15
1RTTI TD-1: SOK=0001 CLOSE      ,RC=-0000001,ERRNO=+0001124,      14:24:15
TS-RC=00,A POWER251      2C308000 0800 800000      14:24:15
1RTTI TD-3: SOK=0001 CLOSE      ,RC=+0000000,ERRNO=+0000000,      14:24:15
TS-RC=00,A POWER251      2C308000 0800 000000      14:24:15
```

If the remote system is a z/VSE system, the following message is issued by TCP/IP for z/VSE:

```
F7 0082 IPN214W TCP has rejected a connection destined for port 175 from
F7 0082 9.002.002.002,4132
```

### Possible causes:

Usually, the remote system and TCP/IP are running, and one of the following reasons can cause the problem:

- a. The remote TCP/IP application is not running (for example VSE/POWER).
- b. The port number of the remote TCP/IP application is incorrect.
- c. The TCP/IP application is running but cannot act as a server (VSE/POWER has issued message 1RTYI). If the remote system starts the connection to your local node and your local system can act as a server, the two nodes may nevertheless sign on successfully.

### Action:

- a. Start the remote TCP/IP application (for example VSE/POWER).
- b. Correct the port number of the remote TCP/IP application and load your corrected NDT, if necessary.
- c. Cause the remote TCP/IP application to act as a server. If the remote system is a z/VSE system, VSE/POWER has issued message 1RTYI. Use the PSTOP TCPIP command to stop the TCP/IP communication and restart it by loading an NDT with at least one TCP node.

### 3. 1RTTI with RECEIVE and ERRNO=1124 or RC=+0000000 Immediately

#### Symptom:

The following trace messages are issued when the CONNECT socket call completes immediately and successfully, as does the SEND socket call (sending an OPEN control record), whereas the socket call RECEIVE (getting a reply to the OPEN control record) fails with ERRNO=1124 (=ENOTCONN). The connection is terminated by the CLOSE socket call. PNET retries to establish the connection every two minutes. Sometimes the socket call RECEIVE completes with RC=+0000000 (and ERRNO=+0000000).

```
PSTART PNET,POWER251      15:27:08
1C39I COMMAND PASSED TO VSE/POWER      15:27:08

1RTTI LINE=TCP START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER251      15:27:08
1RTTI TD-1: SOK=0000 SOCKET      ,RC=+0000000,ERRNO=+0000000,      15:27:08
TS-RC=00,A POWER251      08208000 0000 800000      15:27:08
1RC6I CONNECTION PENDING FOR NODE POWER251, TIME=15:27:08      15:27:08
1RTTI TD-3: SOK=0000 SOCKET      ,RC=+0000001,ERRNO=+0000000,      15:27:08
TS-RC=00,A POWER251      08208000 0000 000000      15:27:08
1RTTI TD-1: SOK=0001 CONNECT      ,RC=+0000001,ERRNO=+0000000,      15:27:08
TS-RC=00,A 009.001.001.001 0C208000 0000 800000      15:27:08
1RTTI TD-2: SOK=0001 CONNECT      ,RC=+0000000,ERRNO=+0001103,      15:27:08
TS-RC=00,A POWER251      0C208000 0000 800000      15:27:08
1RTTI TD-3: SOK=0001 CONNECT      ,RC=+0000000,ERRNO=+0000000,      15:27:08
TS-RC=00,A POWER251      0C208000 0000 000000      15:27:08
1RTTI TD-1: SOK=0001 SEND      ,RC=+0000033,ERRNO=+0000000,      15:27:08
TS-RC=00,A POWER251      OPEN      0      15:27:08
1RTTI TD-S: D6D7C5D5 40404040 D7D6E6C5 D9F2F6F2 09020202 D7D6E6C5      15:27:08
D9F2F5F1 09010101 00000000      15:27:08
1RTTI TD-2: SOK=0001 SEND      ,RC=+0000000,ERRNO=+0001103,      15:27:08
```

## Problem Determination

```
TS-RC=00,A POWER251      0C208000 0000 008000      15:27:08
1RTTI TD-3: SOK=0001 SEND      ,RC=+0000033,ERRNO=+0000000,      15:27:09
TS-RC=00,A POWER251      0C208000 0000 000000      15:27:09
1RTTI TD-1: SOK=0001 RECEIVE    ,RC=+0000033,ERRNO=+0000000,      15:27:09
TS-RC=00,A POWER251      0C208000 0000 800000      15:27:09
1RTTI TD-2: SOK=0001 RECEIVE    ,RC=+0000000,ERRNO=+0001103,      15:27:09
TS-RC=00,A POWER251      0C208000 0000 800000      15:27:09
1RTTI TD-3: SOK=0001 RECEIVE    ,RC=-0000001,ERRNO=+0001124,      15:27:09
TS-RC=08,A POWER251      00000072 E000 8C0000      15:27:09
1RTTI TD-1: SOK=0001 CLOSE      ,RC=-0000001,ERRNO=+0001124,      15:27:09
TS-RC=00,A POWER251      2C308000 0800 800000      15:27:09
1RTTI TD-3: SOK=0001 CLOSE      ,RC=+0000000,ERRNO=+0000000,      15:27:09
TS-RC=00,A POWER251      2C308000 0800 000000      15:27:09
```

If the remote system is a z/VSE system, the following messages are issued by VSE/POWER:

```
1RTTI TD-2: SOK=0000 ACCEPT      ,RC=+0000000,ERRNO=+0001103,      15:27:08
TS-RC=00,P      00000000 0000 800000      15:27:08
1RTTI TD-3: SOK=0000 ACCEPT      ,RC=+0000010,ERRNO=+0000000,      15:27:08
TS-RC=00,P      00000000 0000 000000      15:27:08
1RTTI TD-1: SOK=0010 GETHOSTBYADDR,RC=+0000010,ERRNO=+0000000,      15:27:08
TS-RC=00,P 009.002.002.002 00000000 0000 800000      15:27:08
1RTTI TD-3: SOK=0010 GETHOSTBYADDR,RC=-0000001,ERRNO=+0000000,      15:27:08
TS-RC=00,P 009.002.002.002 00000000 0000 800000      15:27:08
1RTTI TD-1: SOK=0010 RECEIVE    ,RC=+0000033,ERRNO=+0000000,      15:27:08
TS-RC=00,P POWER262      00000000 0000 800000      15:27:08
1RTTI TD-2: SOK=0010 RECEIVE    ,RC=+0000000,ERRNO=+0001103,      15:27:08
TS-RC=00,P POWER262      00000000 0000 800000      15:27:08
1RTTI TD-3: SOK=0010 RECEIVE    ,RC=+0000033,ERRNO=+0000000,      15:27:09
TS-RC=00,P POWER262      OPEN 0      15:27:09
1RTTI TD-R: D6D7C5D5 40404040 D7D6E6C5 D9F2F6F2 09020202 D7D6E6C5      15:27:08
D9F2F5F1 09010101 00000000      15:27:08
1RTHI TCP/IP: NODE POWER262 AWAITING CONNECTION      15:27:08
1RTTI TD-1: SOK=0010 CLOSE      ,RC=+0000033,ERRNO=+0000000,      15:27:09
TS-RC=00,P POWER262      2010000C 0800 800000      15:27:09
1RTTI TD-3: SOK=0010 CLOSE      ,RC=+0000000,ERRNO=+0000000,      15:27:09
TS-RC=00,P POWER262      2010000C 0800 800000      15:27:09
1RTTI TD-1: SOK=0000 ACCEPT      ,RC=+0000000,ERRNO=+0000000,      15:27:09
TS-RC=00,P      00000000 0000 800000      15:27:09
1RTTI TD-2: SOK=0000 ACCEPT      ,RC=+0000000,ERRNO=+0001103,      15:27:09
TS-RC=00,P      00000000 0000 800000      15:27:09
```

The CLOSE socket call is issued because a PSTART command has not yet been issued on the remote system for your node.

### Possible causes:

- a. The connection to your local node has not been started at the remote system (for example, the 'PSTART PNET,nodeid' command has not been issued on a z/VSE system).

### Action:

- a. At the remote system, start the connection to your local node (for example issue the 'PSTART PNET,nodeid' command on a z/VSE system).

## SSL Node Cannot Sign On

To get more information about the failure, enter the command

```
PSTART CNSLTR
```

which produces 1RTTI trace messages (see Appendix D, "PNET BSC/CTCA/TCP/SSL Console Trace," on page 107). To minimize the number of trace messages, use the operand 'PNETSSL' and start just the SSL node that cannot sign on.

When the PSTART CNSLTR command is entered, the following messages are issued:

```
PSTART CNSLTR, ,PNETSSL 13:17:28
1C39I COMMAND PASSED TO VSE/POWER 13:17:28
1R88I OK 13:17:28
1RTTI SD-1: IPW$$$SD VERSION 07/26 (5DF500) RUNNING WITH IPW$$$ 13:17:28
VERSION 07/26 (5E4E00) 13:17:28
1RTTI SD-1: SOK=0002 SELECT-READ ,RC=+0000000,ERRNO=+0000000, 13:17:28
SS-RC=00,P 00000000 0000 400000 13:17:28
1RTTI SD-3: SOK=0002 SELECT-READ ,RC=+0000000,ERRNO=+0000000, 13:18:01
SS-RC=00,P 00000000 0000 400000 13:18:01
1RTTI SD-2: SOK=0002 ACCEPT ,RC=+0000000,ERRNO=+0000000, 13:18:01
SS-RC=02,P 00000000 0000 400000 13:18:01
```

Since the SD subtask issues synchronous socket calls, the SELECT-READ socket call is issued to test that a following synchronous socket call will not block the communication. Therefore, the trace messages often display a SELECT-READ or SELECT-WRITE socket call before displaying a READ, WRITE, or ACCEPT socket call.

The trace message displaying information about the ACCEPT socket call, with the character "P" after "SS-RC=02," shows that the passive connection is active and ready to process incoming CONNECT requests. If no messages are issued, make sure a network definition table (NDT) has been loaded with at least one SSL node.

### 1. 1RTTI with SEND and ERRNO=1124 After 2 Minutes

#### Symptom:

The following trace messages are issued when the CONNECT socket call does not complete successfully. After 2 minutes, the SELECT-WRITE socket call is posted with an exception condition and the following SEND socket call is terminated with ERRNO=1124 (=ENOTCONN), meaning that no remote node answered the CONNECT request. The connection is terminated by the CLOSE socket call. PNET retries to establish the connection every two minutes, which produces the same messages.

```
PSTART PNET,POWER264 07:42:51
1C39I COMMAND PASSED TO VSE/POWER 07:42:51
1R88I OK 07:42:51

1RTTI LINE=SSL START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER264 07:42:51
1RTTI SD-1: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264 08208000 0000 000000 07:42:51
1RC6I CONNECTION PENDING FOR NODE POWER264, TIME= 7:42:51 07:42:51
1RTTI SD-3: SOK=0000 SOCKET ,RC=+0000001,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264 08208000 0000 000000 07:42:51
1RTTI SD-1: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264 0C208000 0000 000000 07:42:51
1RTTI SD-3: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264 0C208000 0000 000000 07:42:51
1RTTI SD-1: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A 009.004.004.004 0C208000 0000 000000 07:42:51
1RTTI SD-3: SOK=0001 CONNECT ,RC=-0000001,ERRNO=+0001103, 07:42:51
SS-RC=00,A POWER264 0C208000 0000 000000 07:42:51
1RTTI SD-1: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0001103, 07:42:51
SS-RC=00,A POWER264 0C208000 0000 000000 07:42:51
1RTTI SD-3: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0000000, 07:42:51
```

## Problem Determination

```

SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
1RTTI SD-1: SOK=0001 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
.....

1RTTI SD-3: SOK=0001 SELECT-WRITE ,RC=+0000001,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264          0C208000 0000 000000          07:44:52
1RTTI SD-1: SOK=0001 SEND      ,RC=+0000033,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264          OPEN SSL 0 0 000000          07:44:52
1RTTI SD-S: D6D7C5D5 40E2E2D3 D7D6E6C5 D9F2F6F2 09020202 D7D6E6C5 07:44:52
D9F2F6F4 09040404 00000000          07:44:52
1RTTI SD-3: SOK=0001 SEND      ,RC=-0000001,ERRNO=+0001124, 07:44:52
SS-RC=08,A POWER264          0C208000 0000 000000          07:44:52
1RTTI SD-1: SOK=0001 CLOSE      ,RC=+0000000,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264          2C308000 0800 000000          07:44:52
1RTTI SD-3: SOK=0001 CLOSE      ,RC=+0000000,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264          2C308000 0800 000000          07:44:52

```

The above message set shows only the important messages. In reality, additional trace messages are issued between:

- 1RTTI for the SELECT-WRITE socket call with RC=+0000000 (indicating connection not ready for I/O) is repeated to indicate that the connection is still processed.
- 1RTTI for an ACCEPT socket call with RC=+0000000 is issued to indicate that incoming CONNECT requests can be processed by a passive connection.

The following message set below shows all messages.

```

1RTTI LINE=SSL START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER264 07:42:51
1RTTI SD-1: SOK=0000 SOCKET      ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264          08208000 0000 000000          07:42:51
1RC6I CONNECTION PENDING FOR NODE POWER264, TIME= 7:42:51          07:42:51
1RTTI SD-3: SOK=0000 SOCKET      ,RC=+0000001,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264          08208000 0000 000000          07:42:51
1RTTI SD-1: SOK=0001 IOCTL      ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
1RTTI SD-3: SOK=0001 IOCTL      ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
1RTTI SD-1: SOK=0001 CONNECT    ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A 009.004.004.004 0C208000 0000 000000          07:42:51
1RTTI SD-3: SOK=0001 CONNECT    ,RC=-0000001,ERRNO=+0001103, 07:42:51
SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
1RTTI SD-1: SOK=0001 IOCTL      ,RC=+0000000,ERRNO=+0001103, 07:42:51
SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
1RTTI SD-3: SOK=0001 IOCTL      ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
1RTTI SD-1: SOK=0001 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=00,A POWER264          0C208000 0000 000000          07:42:51
1RTTI SD-2: SOK=0001 SEND      ,RC=+0000000,ERRNO=+0000000, 07:42:51
SS-RC=02,A POWER264          0C208000 0000 004000          07:42:51
1RTTI SD-3: SOK=0000 SELECT-READ ,RC=+0000000,ERRNO=+0000000, 07:43:17
SS-RC=00,P                    00000000 0000 400000          07:43:17
1RTTI SD-2: SOK=0000 ACCEPT      ,RC=+0000000,ERRNO=+0000000, 07:43:17
SS-RC=02,P                    00000000 0000 400000          07:43:17
1RTTI SD-1: SOK=0001 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 07:43:41
SS-RC=00,A POWER264          0C208000 0000 000000          07:43:41
1RTTI SD-3: SOK=0001 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 07:43:41
SS-RC=00,A POWER264          0C208000 0000 000000          07:43:41
1RTTI SD-1: SOK=0000 SELECT-READ ,RC=+0000000,ERRNO=+0000000, 07:43:51
SS-RC=00,P                    00000000 0000 400000          07:43:51
1RTTI SD-3: SOK=0000 SELECT-READ ,RC=+0000000,ERRNO=+0000000, 07:43:51
SS-RC=00,P                    00000000 0000 400000          07:43:51
1RTTI SD-2: SOK=0000 ACCEPT      ,RC=+0000000,ERRNO=+0000000, 07:44:24

```

## Problem Determination

```
SS-RC=02,P                00000000 0000 400000                07:44:24
1RTTI SD-1: SOK=0000 SELECT-READ ,RC=+0000000,ERRNO=+0000000, 07:44:25
SS-RC=00,P                00000000 0000 400000                07:44:25
1RTTI SD-3: SOK=0001 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 07:44:31
SS-RC=00,A POWER264      0C208000 0000 000000                07:44:31
1RTTI SD-2: SOK=0001 SEND ,RC=+0000000,ERRNO=+0000000, 07:44:31
SS-RC=02,A POWER264      0C208000 0000 004000                07:44:31
1RTTI SD-1: SOK=0001 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 07:44:32
SS-RC=00,A POWER264      0C208000 0000 000000                07:44:32
1RTTI SD-2: SOK=0001 SEND ,RC=+0000000,ERRNO=+0000000, 07:44:32
SS-RC=02,A POWER264      0C208000 0000 004000                07:44:32
1RTTI SD-3: SOK=0001 SELECT-WRITE ,RC=+00000001,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264      0C208000 0000 000000                07:44:52
1RTTI SD-1: SOK=0001 SEND ,RC=+00000033,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264      OPEN SSL 0 0 000000                07:44:52
1RTTI SD-S: D6D7C5D5 40E2E2D3 D7D6E6C5 D9F2F6F2 09020202 D7D6E6C5 07:44:52
D9F2F6F4 09040404 00000000                07:44:52
1RTTI SD-3: SOK=0001 SEND ,RC=-0000001,ERRNO=+0001124, 07:44:52
SS-RC=08,A POWER264 0C208000 0000 000000 07:44:52
1RTTI SD-1: SOK=0001 CLOSE ,RC=+0000000,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264      2C308000 0800 000000                07:44:52
1RTTI SD-3: SOK=0001 CLOSE ,RC=+0000000,ERRNO=+0000000, 07:44:52
SS-RC=00,A POWER264      2C308000 0800 000000                07:44:52
```

### Possible causes:

- a. The IP address is wrong.
- b. The remote system is not running.
- c. The remote system is running, but TCP/IP is not running (TCP/IP for VSE/ESA in case of a z/VSE system, usually running in partition F7).

### Action:

- a. Correct the IP address and load the corrected NDT, if necessary.
- b. Start the remote system.
- c. Start TCP/IP on the remote system (TCP/IP for VSE/ESA in case of a z/VSE system, usually running in partition F7).

## 2. 1RTTI with SEND and ERRNO=1124 Immediately

### Symptom:

The following trace messages are issued when the SEND socket call completes immediately and unsuccessfully with ERRNO=1124 (=ENOTCONN). The connection is terminated by the CLOSE socket call. PNET tries again to establish the connection every two minutes.

```
PSTART PNET,POWER264 07:54:38
1C39I COMMAND PASSED TO VSE/POWER 07:54:38

1RTTI LINE=SSL START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER264 07:54:38
1RTTI SD-1: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000, 07:54:38
SS-RC=00,A POWER264      08208000 0000 000000                07:54:38
1RC6I CONNECTION PENDING FOR NODE POWER264, TIME= 7:54:38 07:54:38
1RTTI SD-3: SOK=0000 SOCKET ,RC=+00000001,ERRNO=+0000000, 07:54:38
SS-RC=00,A POWER264      08208000 0000 000000                07:54:38
1RTTI SD-1: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0000000, 07:54:38
SS-RC=00,A POWER264      0C208000 0000 000000                07:54:38
1RTTI SD-3: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0000000, 07:54:38
SS-RC=00,A POWER264      0C208000 0000 000000                07:54:38
1RTTI SD-1: SOK=0001 CONNECT ,RC=+0000000,ERRNO=+0000000, 07:54:38
SS-RC=00,A 009.004.004.004 0C208000 0000 000000                07:54:38
1RTTI SD-3: SOK=0001 CONNECT ,RC=-0000001,ERRNO=+0001103, 07:54:38
SS-RC=00,A POWER264      0C208000 0000 000000                07:54:38
1RTTI SD-1: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0001103, 07:54:38
SS-RC=00,A POWER264      0C208000 0000 000000                07:54:38
1RTTI SD-3: SOK=0001 IOCTL ,RC=+0000000,ERRNO=+0000000, 07:54:38
SS-RC=00,A POWER264      0C208000 0000 000000                07:54:38
1RTTI SD-1: SOK=0001 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 07:54:38
SS-RC=00,A POWER264      0C208000 0000 000000                07:54:38
```

## Problem Determination

```
1RTTI SD-2: SOK=0001 SEND ,RC=+0000000,ERRNO=+0000000, 07:54:38
SS-RC=02,A POWER264 0C208000 0000 004000 07:54:38
1RTTI SD-3: SOK=0001 SELECT-WRITE ,RC=+0000001,ERRNO=+0000000, 07:54:39
SS-RC=00,A POWER264 0C208000 0000 0000000 07:54:39
1RTTI SD-1: SOK=0001 SEND ,RC=+00000033,ERRNO=+0000000, 07:54:39
SS-RC=00,A POWER264 OPEN SSL 0 0 000000 07:54:39
1RTTI SD-S: D6D7C5D5 40E2E2D3 D7D6E6C5 D9F2F6F2 09A49B89 D7D6E6C5 07:54:39
D9F2F6F4 09A49B8B 00000000 07:54:39
1RTTI SD-3: SOK=0001 SEND ,RC=-0000001,ERRNO=+0001124, 07:54:39
SS-RC=08,A POWER264 0C208000 0000 000000 07:54:39
1RTTI SD-1: SOK=0001 CLOSE ,RC=+0000000,ERRNO=+0000000, 07:54:39
SS-RC=00,A POWER264 2C308000 0800 000000 07:54:39
1RTTI SD-3: SOK=0001 CLOSE ,RC=+0000000,ERRNO=+0000000, 07:54:39
SS-RC=00,A POWER264 2C308000 0800 000000 07:54:39
```

If the remote system is a z/VSE system, the following message is issued by TCP/IP for VSE/ESA:

```
F7 0082 IPN214W TCP has rejected a connection destined for port 2252 from
F7 0082 9.002.002.002,4119
```

### Possible causes:

Usually, the remote system and TCP/IP are running, and one of the following reasons can cause the problem:

- The remote TCP/IP application is not running (for example VSE/POWER).
- The port number of the remote TCP/IP application is incorrect.
- The TCP/IP application is running but cannot act as a server (VSE/POWER has issued message 1RTYI). If the remote system starts the connection to your local node and your local system can act as a server, the two nodes may nevertheless sign on successfully.

### Action:

- Start the remote TCP/IP application (for example VSE/POWER).
- Correct the port number of the remote TCP/IP application and load your corrected NDT, if necessary.
- Cause the remote TCP/IP application to act as a server. If the remote system is a z/VSE system, VSE/POWER has issued message 1RTYI. Use the 'PSTOP TCPSSL' command to stop the TCP/IP communication and restart it by loading an NDT with at least one SSL node.

### 3. 1RTTI with RECEIVE RC=+0000000 or RC=-0000001 and ERRNO=+0001124 Immediately

The following trace messages are issued when the CONNECT socket call completes immediately and successfully, as does the SEND socket call (sending an OPEN control record), whereas the socket call RECEIVE (getting a reply to the OPEN control record) fails with RC=+0000000. The connection is terminated by the CLOSE socket call. PNET tries again to establish the connection every two minutes. Sometimes the socket call RECEIVE completes with RC=-0000001 and ERRNO=+0001124.

```
PSTART PNET,POWER261 13:35:52
1C39I COMMAND PASSED TO VSE/POWER 13:35:52

1RTTI LINE=SSL START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER261 13:35:52
1RTTI SD-1: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,A POWER261 08208000 0000 000000 13:35:52
1RC6I CONNECTION PENDING FOR NODE POWER261, TIME=13:35:52 13:35:52
1RTTI SD-3: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,A POWER261 08208000 0000 000000 13:35:52
1RTTI SD-1: SOK=0000 IOCTL ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,A POWER261 0C208000 0000 000000 13:35:52
1RTTI SD-3: SOK=0000 IOCTL ,RC=+0000000,ERRNO=+0000000, 13:35:52
```



## Problem Determination

```

SS-RC=00,A POWER261          0C208000 0000 000000          13:35:52
1RTTI SD-1: SOK=0000 CONNECT      ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,A 009.004.001.001 0C208000 0000 000000          13:35:52
1RTTI SD-3: SOK=0000 CONNECT      ,RC=-0000001,ERRNO=+0001103, 13:35:52
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:52
1RTTI SD-1: SOK=0000 IOCTL        ,RC=+0000000,ERRNO=+0001103, 13:35:52
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:52
1RTTI SD-3: SOK=0000 IOCTL        ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:52
1RTTI SD-1: SOK=0000 SELECT-WRITE ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:52
1RTTI SD-2: SOK=0000 SEND         ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=02,A POWER261          0C208000 0000 004000          13:35:52
1RTTI SD-3: SOK=0000 SELECT-WRITE ,RC=+0000001,ERRNO=+0000000, 13:35:53
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:53
1RTTI SD-1: SOK=0000 SEND         ,RC=+0000033,ERRNO=+0000000, 13:35:53
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:53
1RTTI SD-S: D6D7C5D5 40E2E2D3 D7D6E6C5 D9F2F6F2 09020202 D7D6E6C5
D9F2F6F1 09010101 00000000          13:35:53
1RTTI SD-3: SOK=0000 SEND         ,RC=+0000033,ERRNO=+0000000, 13:35:53
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:53
1RTTI SD-3: SOK=0002 SELECT-READ  ,RC=+0000001,ERRNO=+0000000, 13:35:54
SS-RC=00,P                    00000000 0000 400000          13:35:54
1RTTI SD-X: 00000005 00000000 00000000 00000000 00000000 ARRAY-0: 00000001
00000000 00000000 00000000          13:35:54
1RTTI SD-1: SOK=0000 RECEIVE      ,RC=+0000033,ERRNO=+0000000, 13:35:54
SS-RC=00,A POWER261          0C208000 0000 000000          13:35:54
1RTTI SD-3: SOK=0000 RECEIVE      ,RC=+0000000,ERRNO=+0000000, 13:35:54
SS-RC=00,A POWER261          00000072 E800 8C0000          13:35:54
1RTTI SD-1: SOK=0000 CLOSE        ,RC=+0000000,ERRNO=+0000000, 13:35:54
SS-RC=00,A POWER261          2C308000 0800 000000          13:35:54
1RTTI SD-3: SOK=0000 CLOSE        ,RC=+0000000,ERRNO=+0000000, 13:35:54
SS-RC=00,A POWER261          2C308000 0800 000000          13:35:54

```

If the remote system is a z/VSE system, the following messages are issued by VSE/POWER:

```

1RTTI SD-3: SOK=0002 SELECT-READ  ,RC=+0000001,ERRNO=+0000000, 13:35:52
SS-RC=00,P                    00000000 0000 400000          13:35:52
1RTTI SD-X: 00000004 00000000 00000000 00000000 00000000 ARRAY-0: 00000004
00000000 00000000 00000000          13:35:52
1RTTI SD-1: SOK=0002 ACCEPT        ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,P                    00000000 0000 000000          13:35:52
1RTTI SD-3: SOK=0002 ACCEPT        ,RC=+0000003,ERRNO=+0000000, 13:35:52
SS-RC=00,P                    00000000 0000 000000          13:35:52
1RTTI SD-1: SOK=0003 GETHOSTBYADDR,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=00,P 009.002.002.002 00000000 0000 000000          13:35:52
1RTTI SD-3: SOK=0003 GETHOSTBYADDR,RC=-0000001,ERRNO=+0000000, 13:35:52
SS-RC=00,P 009.002.002.002 00000000 0000 000000          13:35:52
1RTTI SD-2: SOK=0003 RECEIVE       ,RC=+0000000,ERRNO=+0000000, 13:35:52
SS-RC=02,P POWER262          00000000 0000 400000          13:35:52
1RTTI SD-3: SOK=0003 SELECT-READ  ,RC=+0000001,ERRNO=+0000000, 13:35:53
SS-RC=00,P POWER262          00000000 0000 400000          13:35:53
1RTTI SD-X: 0000000C 00000000 00000000 00000000 00000000 ARRAY-0: 00000008
00000000 00000000 00000000          13:35:53
1RTTI SD-1: SOK=0003 RECEIVE       ,RC=+0000033,ERRNO=+0000000, 13:35:53
SS-RC=00,P POWER262          00000000 0000 000000          13:35:53
1RTTI SD-3: SOK=0003 RECEIVE       ,RC=+0000033,ERRNO=+0000000, 13:35:53
SS-RC=00,P POWER262          00000000 0000 000000          13:35:53
1RTTI SD-R: D6D7C5D5 40E2E2D3 D7D6E6C5 D9F2F6F2 09020202 D7D6E6C5
D9F2F6F1 09010101 00000000          13:35:53
1RTHI TCP SSL: NODE POWER262 AWAITING CONNECTION 13:35:53
1RTTI SD-1: SOK=0003 CLOSE        ,RC=+0000000,ERRNO=+0000000, 13:35:53
SS-RC=00,P POWER262          20100004 0800 000000          13:35:53
1RTTI SD-3: SOK=0003 CLOSE        ,RC=+0000000,ERRNO=+0000000, 13:35:53
SS-RC=00,P POWER262          20100004 0800 000000          13:35:53

```

**Possible causes:**

## Problem Determination

- a. The connection to your local node has not been started at the remote system (for example, the 'PSTART PNET,nodeid' command has not been issued on a z/VSE system).

### Action:

- a. At the remote system, start the connection to your local node (for example, issue the 'PSTART PNET,nodeid' command on a z/VSE system).

## 4. 1RT2I with Failing SSL-INIT or SSL-SOK-INIT

### Symptom:

The socket calls SSL-INIT and SSL-SOK-INIT are used to initialize the general SSL feature and the SSL feature for each connection. Many kinds of failures may occur, which produce various return codes and error numbers.

When the SSL feature for a connection is initialized, one node starts the initialization and uses therefore the private key and reads the member within the keyring library. The node which starts this initialization is the one, which acts as a server. This means the socket call ACCEPT issued by the server has been posted due to the CONNECT request from the remote node. If the private key can be accessed by the node acting as a server, the connection can be established successfully, although the private key may not be accessible by the node acting as the client. Since both nodes can act as a server or a client, it may happen that once the connection is established successfully and at other times the connection can not be established depending on which node acts as server. Usually the server is the node which tried to start the connection first (where the PSTART command has been entered first).

### Possible causes:

One of the following reasons might be the cause for the failure:

- a. The keyring library is not defined to TCP/IP for VSE/ESA.
- b. The private key is not defined to TCP/IP for VSE/ESA.
- c. The keyring library does not exist.
- d. The private key does not exist.
- e. The keyring library is not defined to VSE/POWER.
- f. The private key is not defined to VSE/POWER.

### Action:

- a. Check whether the keyring library is defined to TCP/IP for VSE/ESA. The keyring library is defined by a z/VSE library member named \$SOCKOPT.PHASE. Such a member can be created by step 1 of the job SKSSLKEY distributed by z/VSE in ICCF library 59. If necessary, define a keyring library by using the operands SSLIBN and SSLSUBN.
- b. Check whether the correct private key is defined to TCP/IP for VSE/ESA. The private key is defined by a z/VSE library member named \$SOCKOPT.PHASE. Such a member can be created by the step 1 of the job SKSSLKEY distributed by z/VSE in ICCF library 59. If necessary, define a private key by using the operand SSLMEMN.
- c. Define the keyring library as a z/VSE sublibrary using the DEFINE command of the z/VSE librarian which is invoked by EXEC LIBR.
- d. Define the private key. You may use the step 2 of the job SKSSLKEY distributed by z/VSE in ICCF library 59.

- e. Define the keyring library to VSE/POWER by using the operand KEYRING for the local node when creating a network definition table (NDT). If necessary, create a new NDT and load it by using the VSE/POWER command PLOAD PNET,ndtname.
- f. Define the private key to VSE/POWER by using the operand DNAME for the local node or for the remote node when creating a network definition table (NDT). If necessary, create a new NDT and load it by using the VSE/POWER command PLOAD PNET,ndtname.

See also chapter "Configuring Your z/VSE Host for SSL" in the *z/VSE e-business Connectors User's Guide*.

## Problem Determination

---

## Appendix A. Examples of Defining VTAM BOOKs and Pacing

The examples in this appendix show how to:

- Define a network's configuration to VTAM (on 4.2 level); see "Defining B-BOOKs" below.
- Control the number of VTAM buffers allowed at a given point in time; see "Specifying Pacing" on page 83.

You will need to tailor the examples to your own installation environment. Likewise, you will need to include continuation characters where applicable.

---

### Defining B-BOOKs

The example applies to the network described in Figure 2 on page 5. It assumes that the SDLC connection is between two systems with ICAs.

With respect to the sequential bring-up of the VTAM resources, it is recommended to stick to the suggested sequence of resource definitions in the following VTAM configuration list "B-BOOK".

VSE/POWER PNET SNA does not support parallel sessions between its own application node and other NJE nodes. Therefore, specify PARSESS=NO in the VTAM Application Major Node explicitly, or use the default specification, which is PARSESS=NO.

#### Node B B-BOOK

```
*****
* ATCSTRNB- VTAM START LIST FOR NODEB *
*****
SSCPID=20, C
HOSTSA=20, C
SSCPNAME=SSCPB020, C
HOSTPU=NODEB020, C
NOPROMPT, C
NETID=OURNET01, C
MAXSUBA=255, C
CONFIG=NB, C
IOINT=0, C
SGALIMIT=0, C
BSBUF=(28,,,1), C
CRPLBUF=(60,,,1), C
LFBUF=(70,,,11), C
IOBUF=(70,288,,,11), C
LPBUF=(12,,,6), C
SFBUF=(20,,,20), C
SPBUF=(210,,,32), C
XDBUF=(6,,,1)
*****
* ATCCONNB- VTAM CONFIG LIST FOR NODEB *
*****
MAJNODEB, C
PATHNB, C
CDRMNB, C
CDRSCNB, C
APPLISTB
```

## VTAM B-BOOKS

```
*****
* CDRMNB-   CDRM FOR NODEB                               *
*****
CDRMNB  VBUILD TYPE=CDRM
BBBBB   CDRM  SUBAREA=20,ELEMENT=1,ISTATUS=ACTIVE
DDDDD   CDRM  SUBAREA=22,ELEMENT=1,ISTATUS=INACTIVE

*****
* CDRSCNB  CDRSC FOR NODEB                               *
*****
CDRSCNB VBUILD TYPE=CDRSC
VTAMD   CDRSC CDRM=DDDDD

*****
* APPLISTB THE APPLICATION LIST USED IN NODEB           *
*****
          VBUILD TYPE=APPL
VTAMB   APPL  AUTH=(PASS,ACQ),VPACING=3,                C
          MODETAB=POWRTAB,DLOGMOD=PWRHOST1

*****
* MAJNODEB MAJOR NODE DEFINITION FOR NODEB             *
*****
MAJNODEB VBUILD TYPE=CA
GROUPNB  GROUP LNCTL=SDLC,DIAL=NO
LINE35   LINE  ADDRESS=035,PAUSE=0.2,RETRIES=10
PU22     PU    PUTYPE=5,SUBAREA=22

*****
* PATHNB   PATH DEFINITION FOR NODEB                   *
*****
PATHTOND PATH  DESTSA=22,ER0=(22,1),VR0=0
```

## Node D B-BOOK

```
*****
* ATCSTRND- VTAM START LIST FOR NODED                   *
*****
SSCPID=22,                                             C
HOSTSA=22,                                             C
SSCPNAME=SSCPD022,                                    C
HOSTPU=NODED022,                                     C
NOPROMPT,                                             C
NETID=OURNET01,                                       C
MAXSUBA=255,                                          C
CONFIG=ND,                                            C
IOINT=0,                                              C
SGALIMIT=0,                                          C
BSBUF=(28,,,1),                                       C
CRPLBUF=(60,,,1),                                     C
LFBUF=(70,,,11),                                      C
IOBUF=(70,288,,,11),                                  C
LPBUF=(12,,,6),                                       C
SFBUF=(20,,,20),                                      C
SPBUF=(210,,,32),                                    C
XDBUF=(6,,,1)

*****
* ATCCONND- VTAM CONFIG LIST FOR NODED                 *
*****
MAJNODED,                                             C
PATHND,                                              C
CDRMND,                                              C
CDRSCND,                                             C
APPLISTD

*****
* CDRMND-   CDRM FOR NODED                               *
*****
```

```

CDRMND  VBUILD TYPE=CDRM
BBBBB   CDRM  SUBAREA=20,ELEMENT=1,ISTATUS=INACTIVE
DDDDD   CDRM  SUBAREA=22,ELEMENT=1,ISTATUS=ACTIVE
*****
* CDRSCND  CDRSC FOR NODED                                     *
*****
CDRSCND  VBUILD TYPE=CDRSC
VTAMB    CDRSC  CDRM=BBBBB
*****
* APPLISTD THE APPLICATION LIST FOR NODED.                     *
*****
          VBUILD TYPE=APPL
VTAMD    APPL  AUTH=(PASS,ACQ),VPACING=3,                       C
          MODETAB=POWRTAB,DLOGMOD=PWRHOST1
*****
* MAJNODED MAJOR NODE DEFINITION FOR NODED                     *
*****
MAJNODED VBUILD TYPE=CA
GROUPND  GROUP LNCTL=SDLC,DIAL=NO
LINE35   LINE  ADDRESS=035,PAUSE=0.2,RETRIES=10
PU20     PU    PUTYPE=5,SUBAREA=20
*****
* PATHND   PATH DEFINITION FOR NODED                           *
*****
PATHTONB PATH  DESTSA=20,ER0=(20,1),VR0=0

```

## Specifying Pacing

To prevent PNET from filling all the VTAM buffers while transmitting jobs or output, use the pacing facility of VTAM. It controls the number of buffers which VTAM allows to flow between the sending and receiving nodes at any particular point in time for a session.

Figure 8 shows how, for PNET, the pacing parameters may be specified to VTAM when selecting session parameters. You need only to specify the pacing parameters in the MODEENT table; VSE/POWER generates all other required session parameters.

```

PWRHOST1  APPL  AUTH=(PASS,ACQ),                               C
          DLOGMOD=PWRHOST1,                                   C
          VPACING=3,                                          C
          MODETAB=POWRTAB

```

Figure 8. Example of VTAM Definition for Pacing

Figure 9 shows how to define the logon mode table (MODETAB) for use with PNET. You can define the table together with the above pacing specification.

## VTAM Pacing

```
          PUNCH ' PHASE POWRTAB,* '
POWRTAB  MODETAB
IBMPower MODEENT LOGMODE=PWRHOST1,          C
          PSNDPAC=3,                          C
          SRCVPAC=3,                          C
          SSNDPAC=3,                          C
          FMPROF=X'03',                       C
          TSPROF=X'03',                       C
          PRIPROT=X'72',                      C
          SECPROT=X'72'
        MODEEND
        END
```

*Figure 9. Example of Log-on Mode Table (MODETAB) Definition*



---

## Appendix B. Sample of a PNET Receiver Exit Routine

The sample of the PNET receiver exit routine, which is shown in this appendix, is delivered to you as an A-BOOK in PRD1.MACLIB under the name of NETEXAMP.A. If you extend this example further, you may need macros that are only available with the optional code material.

The sample includes:

1. The set of statements that causes the source code of the PNET receiver exit routine to be assembled and catalogued.
2. The source code of the routine example itself.

---

### Control Statements for Assembly and Cataloging

```
* $$ JOB JNM=NETEXRUN,CLASS=A,DISP=D
// JOB NETEXRUN
// OPTION CATAL
// LIBDEF *,SEARCH=(PRD1.MACLIB)
// LIBDEF *,CATALOG=...
*
* PROVIDE FOR ... CATALOG SUBLIBRARY FOR NETEXAMP
*
// EXEC ASSEMBLY,SIZE=100K
    COPY NETEXAMP
    END
/*
// EXEC LNKEDT
/&
* $$ EOJ
```

---

### Programming Example Source Code

#### Programming Interface Information

```
TITLE ' NETEXAMP'
PUNCH ' PHASE NETEXAMP,*'
*****
*****
*
*   NETWORK   RECEIVER   EXIT   ROUTINE
*
*****
*****
SPACE 2
* INPUT:      VARIOUS REGISTERS WITH CONTENTS AS DEFINED FOR THE
*             NETEXIT INTERFACE; SEE 'REGISTER USAGE' BELOW.
*             IN PARTICULAR REGISTER 3 ADDRESSES A WORKAREA
*             OF 50 BYTES, SINCE THE EXIT IS ASSUMED TO HAVE
*             BEEN LOADED WITH A WORKAREA SIZE OF 50 BYTES.
*
SPACE 2
* FUNCTION:   THIS EXIT ROUTINE CHECKS RECEIVED NETWORK INPUT FOR ANY
*             NETWORK ACCOUNT NUMBER AND FLUSHES THE ENTRY, IF NONE
*             IS PROVIDED.
*             IF THE ENTRY HAS TO BE PROCESSED ON OUR OWN NODE,
*             (AND NOT JUST BE STORED FOR FORWARDING TO OTHER NODES)*
*             - A JOB ENTRY IS ONLY ACCEPTED, WHEN IT COMES FROM
*             A CERTAIN GROUP OF ORIGIN NODES
*             - AN OUTPUT ENTRY MAY ENTER THE OWN NODE ONLY,
*             AFTER ITS PRINTING/PUNCHING PRIORITY HAS BEEN
```

## Receiver Exit Example

```

*           ADJUSTED ACCORDING TO A LOCAL RATIO OF OUTPUT      *
*           AMOUNT TO PROCESSING PRIORITY.                      *
*           SPACE 2                                             *
* OUTPUT:   RETURN TO VSE/POWER WITH BELOW RETURN CODES PROVIDED *
*           IN REGISTER 15:                                     *
*           X'00' - NORMAL RETURN, PROCESS NON MODIFIED RECORD *
*           - NORMAL RETURN, PROCESS UPDATED RECORD           *
*           X'10' - FLUSH NETWORKING INPUT                     *
*           SPACE 2                                             *
* LOCATION: THIS RECEIVER EXIT PHASE IS LOCATED WITHIN THE    *
*           - PAGEABLE AREA OF THE VSE/POWER PARTITION WHEN   *
*           LOADED AT VSE/POWER INITIALIZATION TIME, OR IN THE *
*           - GETVIS AREA OF THE VSE/POWER PARTITION WHEN     *
*           LOADED AFTER INITIALIZATION BY THE 'PLOAD' COMMAND.*
*           FOR DEBUGGING PURPOSES LOCATE THE PNET RECEIVER EXIT *
*           IN STORAGE BY ITS STORAGE DESCRIPTOR 'PNET-EXIT' AND BY *
*           THE POINTER 'PNCBAUE' IN THE PNET MASTER CONTROL BLOCK. *
*           THE LATTER IS ANCHORED VIA THE                      *
*           FIELD 'CAPN' OF THE COMMON ADDRESS TABLE OF VSE/POWER. *
*           SPACE 2                                             *
* REMARKS:  FUNCTIONALLY THIS CODE IS A FANCY EXAMPLE!        *
*           YET IT PROVIDES A CODING SKELETON THAT SHOWS, HOW THE *
*           INFORMATION PASSED TO THE EXIT MAY BE USED TO ACCESS *
*           CONTROL DATA BOTH IN VSE/POWER CONTROL BLOCKS AND IN *
*           RECORDS BEING RECEIVED.                             *
*           FIND DSECTS OF OTHER VSE/POWER CONTROL BLOCKS ON THE *
*           OPTIONAL MATERIALS TAPE. REFER TO THEIR DESCRIPTION *
*           IN THE OPTIONAL DOCUMENTATION.                       *
*           FOR DETAILED CONTENTS OF THE NETWORK CONTROL RECORDS AND *
*           THE USAGE BY DIFFERENT NETWORKING COMPONENTS SEE:  *
*           'NETWORK JOB ENTRY FORMATS AND PROTOCOLS'          *
*           SC23-0070                                           *
*           EJECT                                              *
* REGISTER USAGE:                                             *
*           R0  RECORD ADDRESS PASSED BY VSE/POWER             *
*           R1  RECORD LENGTH PASSED BY VSE/POWER             *
*           R2  RECORD CONTROL BYTE (RCB) AND SUB-RCB (SRCB)   *
*           IN THE TWO LOW ORDER BYTES, PASSED BY VSE/POWER   *
*           R3  ADDRESS OF THE WORKAREA PASSED BY VSE/POWER    *
*           R4-R8 WORK REGISTERS                                *
*           R9  ADDRESS OF THE WORKAREA IN PNET EXIT           *
*           RA  ADDRESS OF COMMON ADDRESS TABLE (CAT) AS PASSED *
*           RB  ADDRESS OF TASK CONTROL BLOCK (TCB) AS PASSED *
*           RC  USED FOR VSE/POWER PURPOSES                     *
*           RD  UNUSED                                          *
*           RE  RETURN ADDRESS TO VSE/POWER                    *
*           RF  EXIT BASE REGISTER AND RETURN CODE TO VSE/POWER *
*           SPACE 2                                             *
NETEXAMP CSECT ,           ESTABLISH MAIN CONTROL SECTION
*           SPACE 1
*           FOLLOWING ADDRESSABILITY IS ASSUMED AT ENTRY:
*           SPACE 1
*           USING *,RF           BASE REG ESTABLISHED BY VSE/POWER
*           USING PADS,RA        ESTABLISH ADDRESSABILITY OF CAT
*           USING TCDS,RB        ESTABLISH ADDRESSABILITY OF TCB
*           B REX000            BUMP OVER STORAGE DESCRIPTOR
*           SPACE 1
*           DC CL12'PNET-EXIT'   DEFINE STORAGE DESCRIPTOR PNET-EXIT
*           SPACE 1
*****
*           INITIAL HOUSE-KEEPING, SAVE INFORMATION IN THE WORKAREA *
*           ABOUT ENTRY BEING RECEIVED                               *
*****
*           SPACE 1
*           THIS PNET EXIT IS CALLED BY VSE/POWER FOR ALL RECEIVERS *
*           CURRENTLY ACTIVE. REENTRANCY IS ENSURED BY USING THE WORKAREA *
*           OF 50 BYTES, WITH WHICH EVERY RECEIVER IS EQUIPPED.

```

## Receiver Exit Example

```

*      THIS EXAMPLE TRIES TO SHOW, HOW THE WORKAREA MAY BE USED TO
*      HOLD INFORMATION ABOUT THE RECORDS BELONGING TO ONE JOB OR
*      OUTPUT ENTRY BEING RECEIVED.
*      ALSO THE EXAMPLE CHECKS, WHETHER THE REQUIRED WORK AREA SIZE
*      OF 50 BYTES HAS BEEN SPECIFIED FOR THIS EXIT. IF NOT, WE DO
*      NOT TRUST THE PASSED WORKAREA; INSTEAD WE INFORM THE CENTRAL
*      OPERATOR BY A WARNING MESSAGE.
      SPACE 1
REX000 DS   0H                      START OF PROGRAM
      SPACE 2
*      WHENEVER A WORK AREA IS USED, THE LENGTH OF THIS AREA MUST BE
*      VERIFIED EACH TIME THE EXIT GAINS CONTROL, BECAUSE
*      - THE EXIT MIGHT HAVE BEEN LOADED BY OPERATOR WITH A WRONG
*      LENGTH OF ITS WORK AREA
*      - A NEW VERSION OF THE EXIT MIGHT HAVE BEEN LOADED AND THE
*      PREVIOUSLY DEFINED WORK AREA IS STILL USED UNTIL THE NEXT
*      QUEUE ENTRY BOUNDARY IS REACHED.
      SPACE 2
      LTR  R9,R3                      WORK AREA SPECIFIED ?
      BZ   REX010                      ..NO, WARN OPER. AND FLUSH ENTRY
      USING WORKAREA,R9                MAKE WORK AREA ADDRESSABLE
      SPACE 1
      L    R4,WORKLEN                  GET SPECIFIED WORK AREA SIZE
      LA   R5,WORKLN                   GET REQU. SIZE OF WORK AREA
      CR   R4,R5                       WORK AREA TOO SHORT ?
      BNL  REX020                      ..NO, CONTINUE EXIT
      SPACE 1
REX010 DS   0H
      DROP R4                          RELEASE ADDRESSABILITY
      SPACE 1
*      RESPECT THAT MACRO IPW$WTO DESTROYS REGISTER R0-R3
      SPACE 1
      LA   R7,REXMSG2                  GET ADDRESS OF REXMSG2
      BAL  R6,SUBWTO                    ISSUE REXMSG2
      SPACE 1
      LA   R7,REXMSG3                  GET ADDRESS OF REXMSG3
      BAL  R6,SUBWTO                    ISSUE REXMSG3
      SPACE 1
      B    REX180                      GO AND FLUSH THE CURRENT ENTRY
      SPACE 2
REX020 DS   0H
      LR   R9,R3                      COPY ADDRESS OF WORKAREA
      USING WORKAREA,R9                MAKE WORKAREA ADDRESSABLE
      STM  R0,R1,WORKRADR              SAVE RECORD ADDRESS AND ITS LENGTH
      STCM R2,3,WORKRCB                SAVE RCB AND SRCB
      CLI  WORKTID,0                   BEGIN OF ENTRY ?
      BNE  REX050                      ..NO, BRANCH TO CONTINUE
      MVC  WORKTID,TCSD                 SAVE TASK-ID IN WORKAREA
      L    R4,TCENCB                   GET POINTER TO NODE CONTROL BLOCK
      USING NCBDS,R4                   ESTABLISH ADDRESSABILITY TO NCB
      MVC  WORKNODE,NCBNAME             MAKE NODE NAME PART OF WORK AREA ID
      DROP R4                          RELEASE ADDRESSABILITY
      SPACE 2
*      THE FOLLOWING CODE SHOWS, HOW THE CENTRAL OPERATOR CAN
*      BE INFORMED EASILY BY MESSAGE SERVICES:
*      WRITE A MESSAGE TO CONSOLE, WHEN CALLED FOR VERY
*      FIRST ENTRY AT ALL. RESPECT PARALLEL USAGE OF
*      CODE BY MANY RECEIVERS!
*      NOTE: MESSAGE SERVICES DESTROYS REGISTERS 0 - 3
      SPACE 1
      TS   REXFLG1                      DID WE ALREADY ISSUE MESSAGE ?
      BC   4,REX050                      ..YES, DO NOT DO IT AGAIN
      SPACE 1
      LA   R7,REXMSG1                  GET ADDRESS OF REXMSG1
      BAL  R6,SUBWTO                    ISSUE REXMSG1
      SPACE 2
*****

```

## Receiver Exit Example

```

*           ANALYZE SUB-RECORD CONTROL BYTE (SRCB)           *
*           FOR RECORD TYPE BEING PASSED CURRENTLY           *
*****
SPACE 1
REX050 DS  0H
        CLI  WORKSRCB,X'C0'      IS THIS A JOB HEADER RECORD ?
        BE   REX100              ..YES, BRANCH TO HANDLE IT
        CLI  WORKSRCB,X'D0'      IS THIS A JOB TRAILER RECORD ?
        BE   REX200              ..YES, BRANCH TO HANDLE IT
        CLI  WORKSRCB,X'E0'      IS THIS A DATASET HEADER RECORD ?
        BE   REX300              ..YES, BRANCH TO HANDLE IT
SPACE 1
B       REX400                  ..NO, PROCESS JOB TYPE RECORD
EJECT
*****
*           PROCESS JOB HEADER RECORD                         *
*****
SPACE 2
*       BOTH JOB AND OUTPUT ENTRIES ARE PRECEDED BY A JOB HEADER *
*       CONTROL RECORD AND ARE TERMINATED BY A JOB TRAILER CONTROL *
*       RECORD.                                                 *
*       FOR JOB TYPE ENTRIES A DATA SET HEADER CONTROL RECORD MIGHT *
*       APPEAR BETWEEN NORMAL DATA RECORDS, TO INDICATE BY A *
*       'RECORD CHARACTERISTICS CHANGE SECTION', THAT A DATA LENGTH *
*       OTHER THAN 80 BYTES IS TRANSMITTED.                   *
*       FOR OUTPUT ENTRIES AT LEAST ONE DATA SET HEADER CONTROL *
*       RECORD FOLLOWS THE JOB HEADER RECORD TO CONTAIN THE *
*       OUTPUT ATTRIBUTES.                                     *
SPACE 2
*       THE JOB HEADER RECORD IS EXAMINED FOR A VALID ACCOUNT NUMBER.
*       SINCE VSE/POWER 4.1 THIS ACCOUNT NUMBER MAY BE SPECIFIED
*       USING THE 'NETACCT=' OPERAND OF THE * $$ JOB STATEMENT.
*       IF NO ACCOUNT NUMBER IS SPECIFIED, THIS CODE ASSUMES, THAT
*       THE 'USER=' VSE/POWER JECL OPERAND MIGHT HAVE BEEN
*       SPECIFIED TO PROVIDE AN E.G. ACCOUNT NUMBER THEREIN. THE
*       JOB HEADER RECORD IS THEN CHECKED FOR THE PRESENCE
*       OF THE OPTIONAL VSE/POWER SECTION.
*       IF THE VSE/POWER SECTION EXISTS, THE FIRST 8 BYTES OF THE
*       USER FIELD CONTAIN THE ACCOUNT NUMBER IN CHARACTER FORMAT.
*       IF HOWEVER THE FIELD CONTAINS BLANK OR NO VSE/POWER
*       SECTION EXISTS IN THE JOB HEADER RECORD, THE JOB OR
*       OUTPUT ENTRY IS FLUSHED.
SPACE 1
REX100 DS  0H
        L    R4,WORKRADR          GET ADDRESS OF RECORD PASSED
        USING NJHDSCT,R4          ESTABLISH ADDRESSABILITY
        CLI  NJHGACCT,C' '        ANY ACCOUNT NUMBER PRESENT ?
        BH   REX140              ..YES, CONTINUE
SPACE 2
*       TRY TO LOCATE A VSE/POWER SUBSYSTEM SECTION
SPACE 1
LR     R0,R4                      COPY JOB HEADER RECORD ADDRESS
AH     R0,NJHLEN                  POINT BEHIND RECORD
LA     R1,NJHG                    POINT TO GENERAL SECTION
SPACE 1
REX120 DS  0H
        CLI  NJHGTYPE-NJHG(R1),NTYPPOWR VSE/POWER SECTION FOUND ?
        BE   REX130              ..YES, BRANCH
        AH   R1,NJHLEN-NJHG(R1)   POINT TO NEXT SECTION
        CR   R1,R0                DID WE REACH END OF RECORD ?
        BL   REX120              ..NO, TRY THIS SECTION
        B    REX180              ..YES, NO VSE/POWER SECTION PRESENT,
*                               GO AND FLUSH THE ENTRY
SPACE 1
REX130 DS  0H
        CLI  NJHPUSER-NJHP(R1),C' ' ACC. NO IN POWER SECTION ?
        BNH  REX180              ..NO, GO AND FLUSH ENTRY

```

## Receiver Exit Example

```

MVC  NJHGACCT,NJHPUSER-NJHP(R1)  COPY ACCOUNT NUMBER
SPACE 1
REX140 DS  0H
MVC  WORKACCT,NJHGACCT  SAVE ACCOUNT NO IN WORK AREA
SPACE 2
*
*  ACCEPT ANY OUTPUT TYPE ENTRY BEING RECEIVED.
*  IDENTIFY OUTPUT RECEIVER FROM
*      - EITHER RECORD CONTROL BYTE (AS BELOW)
*      - OR FROM FIELD 'TCRM' OF TASK-ID, WHICH CONTAINS
*          C'JOB' IN CASE OF JOB RECEIVER OR
*          C'OUT' IN CASE OF OUTPUT RECEIVER
SPACE 1
TM  WORKRCB,X'09'      IS OUTPUT BEING RECEIVED ?
BO  REXDONE           ..YES, BRANCH AND ACCEPT ENTRY
SPACE 2
*
*  IF A JOB IS BEING RECEIVED WHICH IS DESTINED FOR THE LOCAL
*  NODE FOR EXECUTION, THE ORIGINATING NODE IS CHECKED
*  WHETHER THE JOB IS ACCEPTED FROM THIS NODE OR NOT.
*  FOR ALLOWED NODES SEE TABLE 'REXTAB2'; THE OWN NODE
*  NAME IS OBTAINED FROM THE 'MRNN' FIELD OF THE DMB.
SPACE 1
L   R1,CAQC           POINT TO DISK MANAGEMENT BLOCK
USING QCDS,R1        ESTABLISH DMB ADDRESSABILITY
CLC MRNN,NJHGXEQN    IS JOB DESTINED FOR LOCAL SYSTEM
BNE REXDONE          ..NO, DON'T CHECK ORG NODE NAME,
*                   ACCEPT JOB AS STORE/FORWARD NODE;
*                   SUCH JOBS MIGHT NOW BE FLAGGED IN
*                   THE WORK AREA FOR OTHER ACTIONS.
DROP R1              RELEASE ADDRESSABILITY
LA  R1,REXTAB2       ADDR TABLE OF VALID NODE NAMES
SPACE 1
REX150 DS  0H
CLC  NJHGORG,0(R1)    MATCHING NODE NAME ?
BE   REXDONE          ..YES, BRANCH AND ACCEPT JOB
*                   FOR PROCESSING ON OWN NODE
LA   R1,L'NJHGORG(R1) BUMP TO NEXT NODE ENTRY
CLI  0(R1),STOP       END OF TABLE REACHED ?
BNE  REX150          ..NO, CONTINUE LOOPING THROUGH
SPACE 1
REX180 DS  0H
LA   RF,X'10'        SET FLUSH RETURN CODE
BR   RE              RETURN TO VSE/POWER
DROP R4              RELEASE ADDRESSAB. OF JOB HDR
EJECT
*****
*                   PROCESS JOB TRAILER RECORD
*                   *****
SPACE 1
*
*  NO ACTION CODE IS PROVIDED HERE. BUT AS THE PRESENT CODE HAS
*  ACCEPTED ONLY ENTRIES FROM THE NETWORK WITH A NON BLANK
*  NETWORK ACCOUNT NUMBER WHICH HAS BEEN SAVED IN THE EXIT
*  WORK AREA, ONE COULD DO ACCOUNTING EVALUATIONS ON ENTRY
*  LEVEL - ACCORDING TO STORE AND FORWARD ENTRIES OR ENTRIES
*  HAVING BEEN PROCESSED ON OUR OWN NODE, WHEN THE JOB TRAILER
*  RECORD IS PASSED.
REX200 DS  0H
B   REXDONE          AND BRANCH TO EXIT ROUTINE
EJECT
*****
*                   PROCESS DATASET HEADER RECORD
*                   *****
SPACE 1
*
*  NOTE: SEE DESCRIPTION OF JOB HEADER TO FIND OUT, WHERE A
*  DATASET HEADER RECORD MAY APPEAR IN AN ENTRY BEING
*  RECEIVED.
*
*  ACTIONS TAKEN:

```

## Receiver Exit Example

```

*      1) BYPASS A 'RECORD CHARACTERISTICS CHANGE SECTION' OF JOBS.
*      2) SELECT ONLY OUTPUT ENTRY DESTINED TO OUR LOCAL SYSTEM.
*      3) DEMONSTRATE HOW TO FIND THE 'OUTPUT PROCESSING SECTION' AND
*          HOW TO LOCATE THEREIN A CERTAIN 'OUTPUT PROCESSING TEXT UNIT'.
*      4) DEPENDING ON THE ACTUAL LINE COUNT THE OUTPUT PRIORITY,
*          STORED IN THE VSE/POWER SECTION OF THE DATA SET HEADER RECORD,
*          IS REDUCED IF THE SPECIFIED PRIORITY IS HIGHER THAN THE
*          MAXIMUM ALLOWED PRIORITY.
*
*      NOTE: THE GENERAL SECTION OF THE DATASET HEADER RECORD IS
*            ALWAYS THE FIRST SECTION IN THE RECORD AND CAN THEREFORE
*            BE ADDRESSED IMMEDIATELY. ALL OTHER SECTIONS MUST BE
*            SEARCHED AND CAN BE IN ANY SEQUENCE.
*
*      SPACE 1
*      1) BYPASS A RECORD CHARACTERISTICS CHANGE SECTION
REX300 DS    0H
        L    R4,WORKRADR      GET ADDRESS OF RECORD PASSED
        USING NDHSECT,R4      ESTABLISH ADDRESSABILITY
        TM   NDHGMOD,NDHC$MOD  IS THIS JUST A CHAR CHANGE SECTION ?
        BO   REXDONE          ..YES, BRANCH TO ACCEPT RECORD,
*                               WHEN DSHR FOUND IN JOB TYPE ENTRY
*      2) SELECT OUTPUT DESTINED TO OWN NODE
        L    R1,CAQC          POINT TO DISK MANAGEMENT BLOCK
        USING QCDS,R1        ESTABLISH ADDRESSABILITY
        CLC  MRNN,NDHGNODE    IS OUTPUT DESTINED FOR LOCAL SYSTEM
        BNE  REXDONE          ..NO, GO AND ACCEPT OUTPUT AS
*                               AS STORE AND FORWARD NODE;
*                               SUCH OUTPUT MIGHT NOW BE FLAGGED
*                               IN THE WORK AREA FOR OTHER ACTIONS.
*                               RELEASE ADDRESSABILITY OF DMB
        DROP R1
        SPACE 3
*      3) THE FOLLOWING CODE LOCATES THE 'OUTPUT PROCESSING SECTION' AND
*          THEREIN A CERTAIN 'OUTPUT PROCESSING TEXT UNIT' (OPTU), FOR
*          EXAMPLE OPTU WITH KEY X'0026', WHICH STANDS FOR 'ROOM' ACC.
*          TO NJE 'FORMATS AND PROTOCOLS' SC23-0070-03.
*
*      FIND A DESIRED SECTION WITHIN DSHR AS IDENTIFIED BY SECTION TYPE
*      FIELD, NAMELY THE OUTPUT PROCESSING SECTION - TYPE = X'89'. FOR
*      THAT PURPOSE SYMBOLS ARE REFERRED AS GENERATED BY THE DSECT MACRO:
*      IPW$DNR JHR=YES,JTR=YES,DHR=YES,OUT=YES ...
        SPACE 1
        LR   R0,R4            COPY ADDR(DSHR) TO R0
        AH   R0,NDHLEN        POINT R0 BEHIND DSHR
        LA   R1,NDHG          POINT R1 TO BEGIN OF DSHR SECTIONS,
*                               WHERE GENERAL SECTION IS THE FIRST
        SPACE 1
*      RESPECTING THAT ALL SECTIONS OF THE DATASET HEADER START WITH
*      2 BYTES - SECTION LENGTH
*      1 BYTE - SECTION TYPE
*      WE USE SYMBOLS OF THE OUTPUT PROCESSING SECTION (NDHS) TO
*      ADDRESS THE 'LENGTH' AND 'TYPE' PER LOCATED SECTION.
NEXTSECT DS    0H
        CLI  NDHSTYPE-NDHS(R1),NTYPGDS  OUTPUT PROCESSING SECTION ?
        BE   OUTSECT          ..YES, GO FOR OPTB AND OPTU'S
        AH   R1,NDHSLEN-NDHS(R1)  POINT TO NEXT SECTION IN DSHR
        CR   R1,R0            DID WE REACH END OF DSHR ?
        BL   NEXTSECT        ..NO, GO AND CHECK NEXT SECTION
        B    NOTFOUND        ELSE GO AND STOP ALL SEARCHING
        SPACE 2
*      BYPASS STANDARD HEADER (OPTB) OF OUTPUT PROCESSING SECTION AND
*      SEARCH THROUGH THE REST OF THIS SECTION FOR THE OPTU WITH THE
*      DESIRED KEY - ASSUME X'0026'.
*      IN GENERAL AN OPTU STEMS FROM A KEYWORD SPECIFICATION,
*      E.G. 'ROOM=XYZ', WHICH IS CODED AS FOLLOWS:
*      2 BYTES - KEY          (E.G. X'0026')
*      2 BYTES - VALUECOUNT (E.G. X'0001')
*      2 BYTES - VALUELENGTH (E.G. X'0003')

```

## Receiver Exit Example

```

*           N BYTES - VALUE           (E.G. C'XYZ')
* BUT AN OPTU MAY ALSO STEM FROM 'KEYWORD=(VALUE1,VALUE2)'. THEN THE
* 2-BYTE VALUECOUNT CONTAINS X'0002' (OR MORE), FOLLOWED BY REPEATED
* VALUELENGTH AND VALUE FIELDS.
*
* NOTE THE FOLLOWING EXCEPTIONS:
* 1) 'VALUECOUNT' MAY CONTAIN X'0000', THEN NO DATA ELEMENTS
* FOLLOW (NO VALUELENGTH AND VALUE FIELDS), BUT
* THE NEXT OPTU STARTS RIGHT AFTER 'VALUECOUNT'.
* 2) 'VALUELENGTH' MAY CONTAIN X'0000', THEN A NULL VALUE FOLLOWS
* THAT MEANS THERE IS NO VALUE FIELD AT ALL.
* THEN EITHER THE NEXT DATA ELEMENT OR THE NEXT
* OPTU STARTS RIGHT AFTER 'VALUELENGTH'.
*
*           SPACE 1
OUTSECT DS    0H
        USING NDHS,R1           MAKE OUTP. PROC. SECT. ADDRESSABLE
        LR   R0,R1             COPY SECTION ADDRESS TO R0
        AH  R0,NDHSLEN        POINT R0 BEYOND THIS SECTION
        LA   R1,NDHSTEXT      POINT R1 TO BEGIN OF OPTU'S
        DROP R1               RELEASE SECTION ADDRESSABILITY
        SPACE 2
MOREOPTU DS    0H
        CLC  0(2,R1),=X'0026'  OPTU FOUND WITH DESIRED KEY ?
        BE   KEYFOUND          ..YES, GO AND PROCESS KEY-OPTU
        SPACE 1
* BUMP OVER CURRENT NON-MATCHING OPTU WITH ONE OR MORE VALUES
        LH  R2,2(R1)          PICK UP VALUECOUNT
        LA  R1,4(R1)          POINT R1 TO VALUELENGTH
        LTR R2,R2             VALUECOUNT ALL ZERO ?
        BZ  LASTOPTU          ..YES, GO AND CHECK MORE OPTU'S
        SPACE 1
MOREVAL  DS    0H
        LH  R3,0(R1)          PICK UP VALUE LENGTH IN R3
        LA  R1,2(R1)          POINT R1 TO VALUE
        AR  R1,R3             POINT R1 BEHIND VALUE, EVEN
*                               WHEN IT IS A 'NULL' VALUE
        BCT R2,MOREVAL        GO AND REPEAT IF MORE VALUES
        SPACE 1
* CHECK IF THERE ARE MORE OPTU'S IN THE OUTPUT PROCESSING SECTION
LASTOPTU DS    0H
        CR  R1,R0             REACHED END OF OUTP. PROC. SECT.?
        BL  MOREOPTU          ..NO, GO FOR NEXT OPTU
        B   NOTFOUND          ELSE STOP SEARCHING
        SPACE 2
* PICK UP AND PROCESS THE DESIRED VALUE FROM THE OPTU ADDRESSED BY R1
        SPACE 1
KEYFOUND DS    0H
* WRITE YOUR OWN CODE ACCORDING TO 'BUMP OVER CURRENT OPTU'
* TO ADDRESS THE FIRST (OR SUBSEQUENT) VALUE(S) OF THE OPTU
* ADDRESSED BY REGISTER 1.
* THEN DECIDE IF STEP 4) SHOULD BE DONE OR IF IMMEDIATE RETURN
* TO VSE/POWER VIA LABEL 'REXDONE' IS DESIRED.
        SPACE 2
* HANDLE 'NOT FOUND' CONDITION
        SPACE 1
NOTFOUND DS    0H
* WRITE YOUR OWN CODE TO DECIDE IF STEP 4) SHOULD BE DONE OR IF
* IMMEDIATE RETURN TO VSE/POWER VIA LABEL 'REXDONE' IS DESIRED.
        SPACE 3
* 4) NOW ADJUST OUTPUT PRIORITY ACCORDING TO LOCAL RATIO
* OF OUTPUT AMOUNT TO PRINTING/PUNCHING PRIORITY
        SPACE 1
        L   R5,NDHGNREC        GET TOTAL RECORD COUNT
        LA  R1,REXTAB1        ADDRESS PRIORITY TABLE
        SPACE 1
REX310  DS    0H
        C   R5,0(R1)          IS RECORD COUNT IN THIS RANGE ?

```

## Receiver Exit Example

```

BNH  REX320          ..YES, CONTINUE
LA   R1,8(R1)       ADVANCE TO NEXT ENTRY IN TABLE
B    REX310         AND LOOP THROUGH
SPACE 1
REX320 DS  0H
SR   R5,R5          CLEAR WORK REGISTER
IC   R5,4(R1)       PICK UP MAX PRIORITY
LR   R0,R4          COPY ADDR OF DATASET HEADER RECORD
AH   R0,NDHLEN      POINT BEHIND RECORD
LA   R1,NDHG        POINT TO GENERAL SECTION
SPACE 1
REX330 DS  0H
CLI  NDHGTYPE-NDHG(R1),NTYPPWR  VSE/POWER SECTION ?
BE   REX340         ..YES, BRANCH
AH   R1,NDHGLEN-NDHG(R1) POINT TO NEXT SECTION
CR   R1,R0          DID WE REACH END OF RECORD ?
BL   REX330         ..NO, TRY THIS SECTION
B    REXDONE        ..YES, NO VSE/POWER SECTION PRESENT
*                               DON'T INFLUENCE PRIORITY OF ENTRY
SPACE 1
REX340 DS  0H
CLM  R5,1,NDHPPRIO-NDHP(R1) PRIORITY WITHIN VALID RANGE ?
BH   REXDONE        ..YES, BRANCH AND LEAVE PRIORITY
STC  R5,NDHPPRIO-NDHP(R1) STORE NEW (REDUCED) PRIORITY
*                               BY UPDATING DSHR CONTROL RECORD
B    REXDONE        RETURN TO VSE/POWER
DROP R4             RELEASE ADDRESSABILITY OF DSHR
EJECT
*****
*          PROCESS JOB DATA RECORD          *
*****
SPACE 2
*   THE RECORD STARTS WITH ONE OF THE FOLLOWING:
*   '/' - VSE OR MVS JECL STATEMENT
*   '/*' - END OF INPUT OR JES2/JES3 CONTROL STATEMENT
*   '/&' - END OF JOB
*   '* ' - COMMENT OR VSE/POWER JECL STATEMENT
*   THIS EXAMPLE DOES NOT TAKE ANY ACTION AND RETURNS TO
*   VSE/POWER WITH THE 'PROCESS RECORD' RETURN CODE.
SPACE 1
REX400 DS  0H
SPACE 1
*   MORE TO COME HERE ACCORDING TO SPECIFIC NEEDS
SPACE 2
*****
*          RETURN TO VSE/POWER          *
*****
SPACE 1
REXDONE DS  0H          COMMON EXIT
SR   RF,RF            SET UP RETURN CODE
BR   RE              RETURN TO CALLER
EJECT
*****
*   SUBROUTINE TO ISSUE MESSAGE ON OPERATOR CONSOLE
*
*   USAGE OF VSE/POWER MACRO IPW$WTO:
*   THIS MACRO, TOGETHER WITH SOME INDICATIONS SET IN THE TCB,
*   CAN BE USED TO DISPLAY A MESSAGE ON THE CENTRAL OPERATOR
*   CONSOLE.
*   THE USAGE OF THIS MACRO DESTROYS THE REGISTER R0 - R3.
*   THE MESSAGE TO BE DISPLAYED SHOULD CONTAIN ALPHAMERIC
*   CHARACTERS ONLY.
*
*****
SPACE 2
SUBWTO DS  0H
STCM  R7,7,TCMW+1    PASS MESSAGE ADDRESS

```



## Receiver Exit Example

```

MVI   TCMW,X'00'           CLEAR OPTION BYTE
OI    TCMW,TCPCOP         FORCE MESSAGE TO CONSOLE
OI    TCMW,TCDNMM        SUPPRESS MESSAGE MODIFICATION
SPACE
IPW$WTO TCMW             ISSUE MESSAGE
SPACE
MVI   TCMW,X'00'           CLEAR OPTION BYTE
BR    R6                 RETURN TO CALLER
SPACE 5
DROP  R9,RA,RB           RELEASE ADDRESSABILTY
EJECT

*****
*
*           D E F I N I T I O N S
*
*****
SPACE 1
REXFLG1 DC X'00'           PNET EXIT FLAG BYTE 1
SPACE 1
REXTAB1 DC F'100',C'9',AL3(0)  PRIORITY TABLE BASED ON CARD/
DC F'200',C'8',AL3(0)        LINE COUNT, OFFERING MAXIMUM
DC F'400',C'7',AL3(0)        PRINTING/PUNCHING PRIORITY
DC F'600',C'6',AL3(0)        WHICH OUTPUT ENTRY IS
DC F'800',C'5',AL3(0)        ALLOWED TO HAVE AND OFFERING
DC F'1200',C'4',AL3(0)       DUMMY ADDRESS, THAT MIGHT BE
DC F'2000',C'3',AL3(0)       USED TO ADDRESS A SPECIFIC
DC F'4000',C'2',AL3(0)       PROCESSING ROUTINE.
DC X'7FFFFFFF',C'1',AL3(0)
SPACE 1
REXTAB2 DC CL8'DENVER'        TABLE OF NODE NAMES, FROM
DC CL8'LONDON'              WHICH WE ARE WILLING TO
DC CL8'NEWYORK'             ACCEPT A JOB, THAT SHOULD BE
DC CL8'HOUSTON'             PROCESSED ON OUR OWN NODE
DC CL8'RALEIGH'
DC AL1(STOP)                LOGICAL TABLE END OF X'FF'
SPACE 2
*****
*           M E S S A G E S
*
*****
SPACE 1
REXMSG1 DC AL1(REXMSG1L-REXMSG1-1)  LENGTH OF MESSAGE
DC C'CONTROL PASSED TO NETEXIT FOR THE VERY FIRST ENTRY'
REXMSG1L EQU *
SPACE 1
REXMSG2 DC AL1(REXMSG2L-REXMSG2-1)  LENGTH OF MESSAGE
DC C'INCORRECT SIZE OF WORKAREA GIVEN, 50 BYTES NEEDED'
REXMSG2L EQU *
SPACE 1
REXMSG3 DC AL1(REXMSG3L-REXMSG3-1)  LENGTH OF MESSAGE
DC C'DISABLE NETEXIT USING PVAR, RE-LOAD IT BY PLOAD'
REXMSG3L EQU *
SPACE 2
*****
*           L A Y O U T   O F   W O R K   A R E A
*
*****
SPACE 1
WORKAREA DSECT ,           LAYOUT OF WORK AREA
WORKLEN DS F              LENGTH OF WORKAREA AS REQUESTED
*                          AND PASSED BY VSE/POWER
WORKTID DS CL16           TASK IDENTIFIER
WORKNODE DS CL8           NODE NAME OF SERVING TASK
WORKRADR DS F             ADDRESS OF RECORD
WORKRLN DS F             LENGTH OF RECORD
WORKRCB DS X             RCB OF RECORD JUST PASSED
WORKSRCB DS X            SRCB OF RECORD JUST PASSED
WORKACCT DS CL8          ACCOUNT NUMBER OF JOB
WORKFILL DS CL4          UNUSED BYTES

```

## Receiver Exit Example

```
WORKLN  EQU  *-WORKAREA          LENGTH OF WORK AREA
SPACE 2
*****
*          LAYOUT OF VSE/POWER CONTROL BLOCKS          *
*          AND NETWORK CONTROL RECORDS                  *
*****
SPACE 1
IPW$EQU ,          DEFINE REGISTER EQUATES AS
*                USED IN VSE/POWER CODING
SPACE 2
IPW$DPA ,          LAYOUT OF COMMON ADDRESS TABLE
*                (CAT) ALSO CALLED PERMANENT AREA
*                OF VSE/POWER.
SPACE 2
IPW$DTC NET=YES   LAYOUT OF TASK CONTROL BLOCK
SPACE 2
IPW$DQC ,          LAYOUT OF DISK MANAGEMENT BLOCK
SPACE 2
IPW$DQR ,          LAYOUT OF INTERNAL QUEUE RECORD
SPACE 2           ADDRESSED BY TCQV OF TCB
IPW$DPN ,          LAYOUT OF PNET MASTER CTL. BLOCK
SPACE 2
IPW$DNR JHR=YES, JTR=YES, DHR=YES, OUT=YES NETWORK CONTROL RECS
SPACE 2
IPW$DNC ,          LAYOUT OF NODE CONTROL BLOCK
SPACE 3
*                END
*                NOT REQUIRED FOR '.A' COPY BOOK
```

End of Programming Interface Information

---

## Appendix C. Sample of a PNET Transmitter Exit Routine

The sample of the PNET transmitter exit routine, which is shown in this appendix, is delivered to you as an A-BOOK in PRD1.MACLIB under the name of XMTEXAMP.A. If you extend this example further, you may need macros that are only available with the optional code material.

The sample includes:

1. The set of statements that causes the source code of the PNET transmitter exit routine to be assembled and catalogued.
2. The source code of the routine example itself.

---

### Control Statements for Assembly and Cataloging

```
* $$ JOB JNM=XMTEXRUN,CLASS=A,DISP=D
// JOB XMTEXRUN
// OPTION CATAL
// LIBDEF *,SEARCH=(PRD1.MACLIB)
// LIBDEF *,CATALOG=...
*
* PROVIDE FOR ... CATALOG SUBLIBRARY FOR XMTEXAMP
*
// EXEC ASSEMBLY,SIZE=100K
    COPY XMTEXAMP
    END
/*
// EXEC LNKEDT
/&
* $$ EOJ
```

---

### Programming Example Source Code

```
----- Programming Interface Information -----
                TITLE ' XMTEXAMP '
                PUNCH ' PHASE XMTEXAMP,* '
*
*****
*****
*
*   NETWORK TRANSMITTER EXIT ROUTINE
*
*****
*****
                SPACE 2
*   FUNCTION:
*   THIS EXIT ROUTINE PROVIDES AN EXAMPLE OF HOW A TRANSMITTER
*   EXIT CAN BE USED TO
*
*       - INSERT A DATA RECORD INTO THE DATA STREAM
*       - MODIFY A NETWORK CONTROL RECORD
*       - PROCESS A DATA RECORD
*       - PROCESS A CHANGED NETWORK CONTROL RECORD
*       - STOP THE TRANSMISSION OF A QUEUE ENTRY
*
*   DURING THE TRANSMISSION OF A QUEUE ENTRY TO ANOTHER NODE.
*
*   IT ALSO SHOWS THE USE OF THE TRANSMITTER EXIT WORK AREA,
*   THE PARAMETER LIST AND HOW MESSAGES CAN BE WRITTEN TO THE
*****
```

## Transmitter Exit Example

```
* OPERATOR CONSOLE USING VSE/POWER SERVICES. *
* *
* SUPPOSE THE FOLLOWING NETWORKING ENVIRONMENT: *
* *
*           X----->A----->B *
* *
* THE TRANSMITTER EXIT IS LOADED AT LOCAL NODE X. *
* NODE X IS CONNECTED TO NODE A. *
* NODE A IS CONNECTED TO NODE B. *
* *
* THIS TRANSMITTER EXIT ROUTINE ENSURES THAT ONLY JOBS CAN LEAVE *
* THE LOCAL NODE X. *
* OUTPUT IS NOT ALLOWED TO LEAVE NODE X AND WILL BE QUEUED AGAIN *
* IN THE XMT QUEUE WITH THE CORRESPONDING 'HOLD' DISPOSITION. *
* *
* ASSUME THAT SEVERAL JOBS WHICH RESIDE IN THE XMT QUEUE OF NODE X *
* ARE TO BE TRANSMITTED TO NODE A AND B. FOR JOBS WHOSE JOB HEADER *
* SECTION CONTAINS NO USER SECTION AND WHOSE FINAL DESTINATION IS *
* NODE B, THE EXIT ROUTINE PROVIDES A USER SECTION TO THEIR JOB *
* HEADER RECORD. IN ADDITION, THE VSE/POWER JOB DISPOSITION IS *
* SET TO 'L' AND A COMMENT RECORD IS INSERTED AFTER THE JOB HEADER *
* RECORD. *
* JOBS WHOSE FINAL DESTINATION IS NOT NODE B ARE NOT TOUCHED *
* BY THE EXIT. *
* *
* SINCE THE EXIT ROUTINE REQUIRES A WORK AREA , IT IS VERIFIED *
* THAT THE WORK AREA WHICH IS SPECIFIED IN THE PLOAD COMMAND OR *
* IN THE POWER MACRO IS LARGE ENOUGH. *
* IF A USER SECTION IS TO BE APPENDED TO THE JOB HEADER RECORD *
* THE EXIT ROUTINE VERIFIES IF THE WORK AREA IS LARGE ENOUGH TO *
* CONTAIN A COPY THE JOB HEADER RECORD AND THE USER SECTION. *
* IF IT IS NOT LARGE ENOUGH THE TRANSMISSION OF THE QUEUE ENTRY *
* IS STOPPED BY SETTING THE RETURN CODE TO 'FLUSH HOLD'. *
* A MESSAGE IS WRITTEN TO THE OPERATOR CONSOLE, INDICATING THAT *
* THE EXIT MUST BE LOADED AGAIN WITH THE REQUIRED WORK AREA SIZE. *
*   SPACE 2
* INPUT: *
* - PARAMETER LIST, WHOSE LAYOUT IS DEFINED IN MACRO IPW$DTX *
* - SEVERAL REGISTERS WHICH ADDRESS VSE/POWER CONTROL BLOCKS. *
*   SEE REGISTER USAGE BELOW FOR MORE DETAILS. *
*   SPACE 2
* OUTPUT: *
* - PARAMETER LIST. *
*   THIS EXIT RETURNS TO VSE/POWER WITH RETURN CODES SHOWN BELOW. *
* *
*   TEXRC: X'00' - PROCESS RECORD *
*           X'08' - INSERT DATA RECORD *
*           X'14' - PROCESS MODIFIED NETWORK CONTROL RECORD *
*           X'18' - FLUSH QUEUE ENTRY WITH 'HOLD' *
*   SPACE 2
* LOCATION: *
*   THIS TRANSMITTER EXIT ROUTINE IS LOCATED WITHIN THE *
* *
*   - PAGEABLE AREA OF THE VSE/POWER PARTITION WHEN *
*   LOADED AT VSE/POWER INITIALIZATION TIME BY THE *
*   POWER MACRO, OR IN THE *
*   - GETVIS AREA OF THE VSE/POWER PARTITION WHEN *
*   LOADED AFTER INITIALIZATION BY THE 'PLOAD' COMMAND. *
* *
*   FOR DEBUGGING PURPOSES LOCATE THE PNET EXIT IN STORAGE *
*   BY ITS STORAGE DESCRIPTOR 'TRANSMITTER EXIT' AND BY THE *
*   POINTER 'PNCBTUE' OF THE PNET MASTER CONTROL BLOCK. *
*   THE LATTER IS ANCHORED VIA THE FIELD 'CAPN' IN THE *
*   THE COMMON ADDRESS TABLE (CAT) OF VSE/POWER. *
*   SPACE 2
```

## Transmitter Exit Example

```

* REMARKS:
* THIS CODE IS AN EXAMPLE TO SHOW, HOW THE INFORMATION
* PASSED TO THE EXIT MAY BE USED TO ACCESS CONTROL DATA
* BOTH IN VSE/POWER CONTROL BLOCKS AND IN RECORDS
* BEING TRANSMITTED.
* YOU CAN FIND DSECTS OF OTHER VSE/POWER CONTROL BLOCKS ON THE
* OPTIONAL MATERIALS TAPE. PLEASE REFER TO THEIR DESCRIPTION
* IN THE OPTIONAL DOCUMENTATION.
* FOR DETAILED CONTENTS OF THE NETWORK CONTROL RECORDS AND
* THE USAGE BY DIFFERENT NETWORKING COMPONENTS SEE:
* 'NETWORK JOB ENTRY FORMATS AND PROTOCOLS' SC23-0070
*
* REGISTER USAGE:
* R0 WORK REGISTER
* R1 ADDRESS OF PARAMETER LIST AT ENTRY
* R2-R7 WORK REGISTERS
* R8 ADDRESS OF PARAMETER LIST DURING EXIT PROCESSING
* R9 ADDRESS OF XMTEXIT WORK AREA
* RA ADDRESS OF COMMON ADDRESS TABLE (CAT)
* RB ADDRESS OF TRANSMITTER TASK CONTROL BLOCK (TCB)
* RC USED FOR VSE/POWER PURPOSES
* RD NOT USED
* RE RETURN ADDRESS TO VSE/POWER
* RF EXIT BASE REGISTER ESTABLISHED BY VSE/POWER
*
*****
EJECT
* ESTABLISH ADDRESSABILITY OF BASE REGISTER AND VSE/POWER
* CONTROL BLOCKS. DEFINE STORAGE DESCRIPTOR FOR BETTER LOCATING
* THIS EXIT IN STORAGE.
SPACE 1
XMTEXAMP CSECT ESTABL. MAIN CONTROL SECTION
SPACE 1
USING *,RF ESTABL. BASE REGISTER
USING PADS,RA ESTABL. ADDRESSABILITY OF CAT
USING TCDS,RB ESTABL. ADDRESSABILITY OF TCB
B TEXSTART BRANCH OVER STORAGE DESCRIPTOR
SPACE
DC CL16'TRANSMITTER EXIT' DEFINE STORAGE DESCRIPTOR
SPACE 2
* SINCE THIS EXAMPLE USES VSE/POWER MACRO IPW$WTO TO WRITE
* MESSAGES TO THE OPERATOR CONSOLE AND THE CONTENTS OF REGISTERS
* R0,R1,R2,R3 ARE DESTROYED BY THIS MACRO, THE ADDRESS OF THE
* PARAMETER LIST IS SAVED INTO REGISTER 8
SPACE 1
TEXSTART DS 0H START OF ROUTINE
LR R8,R1 SAVE ADDR. OF PARAMETER LIST
USING TEXDS,R8 ADDRESS PARAMETER LIST
SPACE 2
* WHENEVER A WORK AREA IS USED, THE LENGTH OF THIS AREA MUST BE
* VERIFIED EACH TIME THE EXIT GAINS CONTROL, BECAUSE
* - THE EXIT MIGHT HAVE BEEN LOADED BY OPERATOR WITH A WRONG
* LENGTH OF ITS WORK AREA
* - A NEW VERSION OF THE EXIT MIGHT HAVE BEEN LOADED AND THE
* PREVIOUSLY DEFINED WORK AREA IS STILL USED UNTIL THE NEXT
* QUEUE ENTRY BOUNDARY IS REACHED.
SPACE 2
* SINCE A WORK AREA IS REQUIRED BY THIS ROUTINE, VERIFY IF ONE
* IS SPECIFIED. IF NOT, THE TRANSMISSION IS STOPPED.
SPACE 1
ICM R9,15,TEXWA WORK AREA SPECIFIED ?
BZ TEXMS1 ..NO, ISSUE MESSAGE AND
* FLUSH TRANSMISSION
SPACE 2
* THE WORK AREA EXISTS AND CAN NOW BE ADDRESSED. TEST IF
* THE WORK AREA IS LARGE ENOUGH. IF NOT, THE TRANSMISSION
* IS STOPPED.

```

## Transmitter Exit Example

```

SPACE 1
USING WADSECT,R9           MAKE WORK AREA ADDRESSABLE
L   R2,WALNGTH           GET SPECIFIED WORK AREA SIZE
LA  R3,WALEN             GET REQU. SIZE OF WORK AREA
CR  R2,R3                WORK AREA TOO SHORT ?
BL  TEXMS2                ..YES, GO TO ISSUE MESSAGE
SPACE 2
*   OUTPUT DATA MUST NOT LEAVE LOCAL NODE X AND THE TRANSMISSION
*   OF THE QUEUE ENTRY IS STOPPED BY SETTING RETURN CODE
*   'FLUSH HOLD'
SPACE 1
TEX050 DS   0H
      CLI  WAFLG3,X'00'    ENTERED FOR THE FIRST TIME ?
      BNE  TEX100         ..NO, BYPASS JOB TEST
      MVI  WAFLG3,X'01'    SET FIRST TIME BYTE
      TM   TEX0T,TEOJOB    JOB TYPE DATA STREAM ?
      BO   TEX100         ..YES, GO AHEAD
      B    TEXMS4         GO TO ISSUE MESSAGE
SPACE 2
*   PROCESS JOB TYPE DATA STREAM. USE THE EQUATES
*   OF THE PARAMETER LIST. DECIDE WHETHER A NETWORK CONTROL
*   RECORD OR A JOB DATA RECORD IS PASSED TO THIS EXIT
SPACE 1
TEX100 DS   0H
      TM   TEXRT,TERJHR    JOB HEADER RECORD ?
      BO   TEXJHR         ..YES, BRANCH TO HANDLE IT
      TM   TEXRT,TERJTR    JOB TRAILER RECORD ?
      BO   TEXJTR         ..YES, BRANCH TO HANDLE IT
      TM   TEXRT,TERDSHR   DATA SET HEADER RECORD ?
      BO   TEXDSHR        ..YES, BRANCH TO HANDLE IT
SPACE 1
      B    TEXDATA        PROCESS JOB DATA RECORD
SPACE 3
*****
*   PROCESS JOB HEADER RECORD (JHR)   *
*****
SPACE 1
TEXJHR DS   0H
SPACE 2
*   SINCE IT IS NECESSARY TO IDENTIFY THE MESSAGES WRITTEN TO THE
*   OPERATOR CONSOLE, THE MESSAGE PREFIX IS SET UP WITH
*   THE TASK ID, THE NAME AND THE FINAL DESTINATION OF THE
*   CURRENTLY TRANSMITTED JOB. THE INFORMATION IS RETRIEVED FROM
*   THE GENERAL SECTION OF THE JHR WHICH IS THE FIRST SECTION
SPACE 1
MVC  WAMES(MESL+1),MES    MOVE MESSAGE INTO WORK AREA
MVC  WAMTID,TCTI         INSERT TASK ID INTO WORK AREA
L    R7,TEXRV            LOAD JHR ADDR. INTO R7
USING NJHDSECT,R7        ADDRESS JHR
MVC  WAMQEN,NJHGJNAM     GET NAME OF QUEUE ENTRY
MVC  WAMEX,NJHGXEQN     GET EXEC. NODE NAME
MVC  WAEXND,WAMEX       COPY EXECUTION NODE NAME
CLC  WAEXND,NODEB       JOB FOR NODE B ?
BNE  TEXPRC             ..NO, PROCESS JOB DATA
SPACE 2
*   SET UP REGISTER 1 WITH THE ACTUAL SECTION ADDRESS OF THE JHR
*   AND LET REGISTER 0 POINT TO THE END OF THE JHR
SPACE 1
LR   R0,R7              COPY ADDRESS OF JHR
AH   R0,NJHLEN          POINT BEHIND JHR
LA   R1,NJHG            GET FIRST SECTION OF JHR
DROP R7                 RELEASE JHR ADDRESSABILITY
SPACE 2
*   VERIFY WHETHER A USER SECTION IS ALREADY APPENDED TO
*   THE JHR AND IDENTIFY DIFFERENT KINDS OF JHR SECTIONS.
*   SINCE THE VARIOUS SECTIONS OF THE JHR CAN OCCUR IN AN
*   ARBITRARY SEQUENCE, EXCEPT FOR THE GENERAL SECTION

```

## Transmitter Exit Example

```

*      WHICH IS THE FIRST, THE JOB HEADER RECORD IS
*      EXAMINED FROM SECTION TO SECTION.
*      REGISTER 1 CONTAINS ADDRESS OF NEXT JHR SECTION AND
*      REGISTER 0 POINTS BEHIND THE END OF THE JHR
SPACE 1
JHRLOOP DS  0H          LOOP THROUGH JHR SECTIONS
        CR   R1,R0      END OF JHR REACHED ?
        BNL  JHREND     ..YES, LEAVE LOOP
SPACE 2
*      DETERMINE KIND OF JHR SECTION USING EQUATES OF IPW$DNR
SPACE 1
        CLI  NJHGTYPE-NJHG(R1),NTYPPWR POWER SECTION ?
        BE   PRCPW      ..YES, BRANCH AND PROCESS IT
        TM   NJHGTYPE-NJHG(R1),NTYPUSER USER SECTION ?
        BO   PRCUS      ..YES, BRANCH AND PROCESS IT
        B    CONT       OTHERWISE PREPARE LOOP
*
*      CONTINUATION
SPACE 2
PRCUS  DS  0H          PROCESS USER SECTION
        MVI  WAFLG2,X'01' INDICATE USER SECTION FOUND
        B    CONT       PREPARE LOOP CONTINUATION
SPACE 2
PRCPW  DS  0H          PROCESS VSE/POWER SECTION
        LA   R2,NJHPDISP-NJHP(R1) GET ADDRESS OF JOB DISPOS.
        ST   R2,WAPWRPT   SAVE ADDRESS OF JOB DISP.
        L    R3,TEXRV     GET JHR ADDRESS
        SR   R2,R3        COMPUTE OFFSET OF JOB DISP.
        ST   R2,WAPWROF   SAVE OFFSET OF JOB DISP.
SPACE 2
CONT   DS  0H          PREPARE LOOP CONTINUATION
        AH   R1,NJHGLN-NJHG(R1) GET NEXT SECTION ADDRESS
        B    JHRLOOP     CONTINUE LOOP
SPACE 2
JHREND DS  0H
        CLI  WAFLG2,X'01' USER SECTION ALREADY PRESENT?
        BE   JHRUPD     ..YES, BRANCH AND UPDATE JHR
SPACE 2
*      SINCE A USER SECTION DID NOT APPEAR AND DESTINATION
*      ADDRESS IS NODE B, A USER SECTION WILL BE APPENDED AT THE
*      END OF THE JHR. FIRST, VERIFY WHETHER THE WORK AREA IS
*      LARGE ENOUGH TO CONTAIN THE EXTENDED JHR.
SPACE 1
        L    R2,TEXRL     GET LENGTH OF JHR
        LA   R3,NJHULL    GET LENGTH OF USER SECTION
        AR   R2,R3        COMPUTE NEW JHR LENGTH
        ST   R2,WAJHRL    SAVE NEW LENGTH OF JHR
        LA   R3,WAJHRLN   GET LENGTH OF JHR W.A.
        CR   R2,R3        ENOUGH SPACE IN W.A. ?
        BH   TEXMS3      ..NO, BRANCH TO FLUSH
SPACE 2
*      SINCE THE LENGTH OF THE JHR WILL BE ALTERED, THE OLD
*      JHR HAS TO BE COPIED INTO THE WORK AREA
SPACE 1
        L    R2,TEXRV     GET FROM ADDRESS OF JHR
        L    R3,TEXRL     GET FROM LENGTH OF JHR
        LA   R4,WANJHR    GET TO ADDRESS OF WORK AREA
        LR   R5,R3        COPY LENGTH OF JHR
        MVCL R4,R2        COPY JHR INTO WORK AREA
SPACE 2
*      APPEND USER SECTION TO JHR IN WORK AREA. RESPECT THAT R4
*      POINTS TO FIRST STORAGE LOCATION JUST BEHIND JHR DUE TO
*      THE PRECEDING MVCL INSTRUCTION
SPACE 1
        LA   R2,NJHULN    GET ADDRESS OF USER SECTION
        LA   R3,NJHULL    GET LENGTH OF USER SECTION
        LR   R5,R3        COPY LENGTH OF USER SECTION
        MVCL R4,R2        MOVE USER SECTION INTO W.A.

```

## Transmitter Exit Example

```

SPACE 2
*   UPDATE LENGTH OF JHR WITHIN THE COPY AND SET VSE/POWER
*   JOB DISPOSITION TO 'L'
SPACE 1
LA   R3,WANJHR           GET ADDRESS OF WORK AREA
L    R2,WAPWROF          GET OFFSET OF JOB DISPOS.
AR   R3,R2               COMPUTE ADDRESS OF JOB
*                               DISPOSITION FIELD IN WORK AREA
MVI  0(R3),C'L'         SET VSE/POWER JOB DISP. TO 'L'
L    R2,WAJHRL           GET NEW LENGTH OF JHR
STH  R2,WANJHR           SET NEW LENGTH OF JHR
SPACE 2
*   INDICATE THAT AT RECEIPT OF THE NEXT RECORD, A RECORD
*   MUST BE INSERTED.
*   UPDATE PARAMETER LIST; SINCE LENGTH AND ADDRESS OF THE
*   JHR WERE ALTERED, SET UP RETURN CODE 'MODIFY'
SPACE 1
MVI  WAFLG,X'01'        INDICATE: RECORD TO INSERT
LA   R2,WANJHR          GET WORK AREA ADDRESS
ST   R2,TEXRV           SAVE INTO PARAMETER LIST
L    R3,WAJHRL           GET LENGTH OF NEW JHR
ST   R3,TEXRL           SAVE INTO PARAMETER LIST
MVI  TEXRC,TERMOD       SET RETURN CODE: MODIFY
B    TEXEND              GO TO END OF ROUTINE
SPACE 2
*   A USER SECTION IS ALREADY CONTAINED IN THE JHR. THE
*   VSE/POWER JOB DISPOSITION IS SET TO 'L' WITHIN THE
*   VSE/POWER SECTION OF THE JOB HEADER RECORD
SPACE 1
JHRUPD DS  0H            UPDATE TOTAL LENGTH OF JHR
L      R2,WAPWRPT        GET ADDRESS OF JOB DISP.
MVI   0(R2),C'L'         SET VSE/POWER JOB DISP. TO 'L'
SPACE 2
*   SINCE LENGTH OR ADDRESS OF THE JHR WAS NOT CHANGED
*   NORMAL PROCESSING IS INDICATED IN THE PARAMETER LIST
SPACE 1
B     TEXPRC              GO TO NORMAL PROCESSING
SPACE 3
*****
*   PROCESS DATA SET HEADER RECORD (DSHR)
*****
SPACE 1
TEXDSHR DS  0H
SPACE 2
*   SINCE A DSHR MAY APPEAR AFTER A JHR, DETERMINE
*   IF THE PRECEDING JHR HAD BEEN MODIFIED AND IF A COMMENT
*   RECORD HAD BEEN INSERTED ALREADY
SPACE 1
CLI  WAFLG,X'00'        NOTHING ALTERED IN JHR OR
*                               RECORD ALREADY INSERTED ?
BE   TEXPRC              ..YES, PROCESS DSHR
BAL  R5,INSUB           BRANCH TO INSERTION ROUTINE
B    TEXEND              GO TO END OF ROUTINE
SPACE 3
*****
*   PROCESS JOB TRAILER RECORD
*****
SPACE 1
TEXJTR DS  0H
SPACE 2
*   INSERT YOUR CODE IF YOU WANT TO DO SOMETHING
*   WITH A JOB TRAILER RECORD
SPACE 1
B    TEXPRC              PROCESS RECORD
SPACE 3
*****
*   PROCESS JOB DATA RECORD
*

```



## Transmitter Exit Example

```

*****
SPACE 1
TEXDATA DS 0H
SPACE 2
* SINCE A JOB RECORD NORMALLY APPEARS AFTER A JHR, DETERMINE
* IF THE PRECEDING JHR HAD BEEN MODIFIED AND A COMMENT RECORD
* HAD BEEN INSERTED ALREADY
SPACE 1
CLI WAFLG,X'00' NOTHING ALTERED IN JHR OR
* RECORD ALREADY INSERTED ?
BE TEXPRC ..YES, PROCESS DATA RECORD
BAL R5,INSUB BRANCH TO INSERTION ROUTINE
B TEXEND GO TO END OF ROUTINE
SPACE 2
*****
* PROCESS CURRENT RECORD *
*****
SPACE 1
TEXPRC DS 0H
MVI TEXRC,TEROK SET RETURN CODE: PROCESS
B TEXEND GO TO END OF ROUTINE
SPACE 2
*****
* ISSUE MESSAGE TO INDICATE WORK AREA MISSING. TRANSMISSION *
* OF THE CURRENTLY PROCESSED QUEUE ENTRY IS STOPPED. *
*****
SPACE 1
TEXMS1 DS 0H
SPACE 1
LA R7,TXMSG4 GET ADDRESS OF TXMSG4
BAL R6,SUBWTO ISSUE TXMSG4
SPACE 2
LA R7,TXMSG2 GET ADDRESS OF TXMSG2
BAL R6,SUBWTO ISSUE TXMSG2
SPACE 2
LA R7,TXMSG3 GET ADDRESS OF TXMSG3
BAL R6,SUBWTO ISSUE TXMSG3
SPACE 1
MVI TEXRC,TERFLSH SET RETURN CODE: FLUSH HOLD
B TEXEND GO TO END OF ROUTINE
SPACE 2
*****
* ISSUE MESSAGE TO INDICATE WRONG SIZE OF WORK AREA. *
* TRANSMISSION OF THE CURRENTLY PROCESSED QUEUE ENTRY IS *
* STOPPED. *
*****
SPACE 1
TEXMS2 DS 0H
SPACE 1
LA R7,TXMSG1 GET ADDRESS OF TXMSG1
BAL R6,SUBWTO ISSUE TXMSG1
SPACE 2
LA R7,TXMSG2 GET ADDRESS OF TXMSG2
BAL R6,SUBWTO ISSUE TXMSG2
SPACE 2
LA R7,TXMSG3 GET ADDRESS OF TXMSG3
BAL R6,SUBWTO ISSUE TXMSG3
SPACE 1
MVI TEXRC,TERFLSH SET RETURN CODE: FLUSH HOLD
B TEXEND GO TO END OF ROUTINE
SPACE 2
*****
* A JHR APPEARED BUT THE WORK AREA IS NOT LARGE ENOUGH TO *
* CONTAIN A COPY OF THE EXTENDED JHR. *
* THE OPERATOR IS INFORMED AND THE TRANSMISSION OF THE *
* CURRENTLY PROCESSED QUEUE ENTRY IS STOPPED. *
* THE DSECT OF THE WORK AREA MUST BE EXTENDED TO CONTAIN THE *

```

## Transmitter Exit Example

```

*          DATA OF THE EXTENDED JOB HEADER RECORD.          *
*          THE EXIT MUST BE ASSEMBLED ANEW AND LOADED AGAIN WITH THE *
*          APPROPRIATE WORK AREA SIZE                          *
*****
SPACE 1
TEXMS3  DS   0H
        MVC  WAMFIELD,TEXMSG4      MOVE MESS. INTO MESSAGE FIELD
        L    R1,WAJHRL             GET LENGTH OF NEW JHR
        LA   R2,WALN              GET LENGTH OF WORK AREA IN
*                                     FRONT OF JHR SAVE AREA
        AR   R1,R2                 COMPUTE REQUIRED W.A. LENGTH
        CVD  R1,WAOUTF             CONVERT SIZE TO PACKED DECIMAL
        UNPK WAMFIELD+24(5),WAOUTF MAKE IT PRINTABLE
        OI   WAMFIELD+28,X'F0'     AND REMOVE SIGN
        MVI  TEXRC,TERFLSH        SET RETURN CODE: FLUSH HOLD
        SPACE 1
        LA   R7,WAMES             GET ADDRESS OF UPDATED TEXMSG4
        BAL  R6,SUBWTO            ISSUE TEXMSG4
        SPACE 2
        B    TEXEND              GO TO END OF ROUTINE
        SPACE 2
*****
*          TRANSMISSION OF OUTPUT DATA IS NOT ALLOWED. A MESSAGE IS *
*          WRITTEN TO THE OPERATOR CONSOLE AND THE TRANSMISSION OF *
*          THE CURRENTLY PROCESSED QUEUE ENTRY IS STOPPED        *
*****
SPACE 1
TEXMS4  DS   0H
        SPACE 1
        LA   R7,TEXMSG7          GET ADDRESS OF TEXMSG7
        BAL  R6,SUBWTO            ISSUE TEXMSG7
        SPACE 2
        MVI  TEXRC,TERFLSH        SET RETURN CODE: FLUSH HOLD
        B    TEXEND              GO TO END OF ROUTINE
        SPACE 2
*****
*          SUBROUTINE FOR INSERTION OF DATA RECORDS. THIS ROUTINE IS *
*          CALLED WHEN A DATA RECORD IS TO BE INSERTED AFTER A *
*          PRECEDING JOB HEADER RECORD. REGISTER 5 POINTS BACK TO *
*          THE CALLING ROUTINE.                                  *
*****
SPACE 1
INSUB   DS   0H
        SPACE 2
*          A JOB DATA RECORD IS TO BE INSERTED. GET ADDRESS AND LENGTH *
*          OF THIS RECORD AND UPDATE PARAMETER LIST              *
        SPACE 1
        LA   R2,TEXIREC          GET ADDRESS OF INSERT. RECORD
        ST   R2,TEXRV            SET RECORD ADDRESS
        MVC  TEXRL,TEXIRECL      SET RECORD LENGTH
        MVI  TEXCC,X'00'        SET CARRIAGE CONTROL
        MVI  TEXRT,TERNCD       SET RECORD TYPE
        MVI  TEXDT,TEDJRNC      SET DATA RECORD TYPE
        MVI  TEXRC,TERINS       SET RETURN CODE: INSERT
        MVI  WAFLG,X'00'        INDICATE: INSERTION DONE
        SPACE 2
*          PREPARATION FOR MESSAGE TEXMSG9
        SPACE 1
        MVC  WAMFIELD,TEXMSG9    INSERT MESSAGE
        SPACE 1
        LA   R7,WAMES           GET ADDRESS OF UPDATED TEXMSG9
        BAL  R6,SUBWTO          ISSUE TEXMSG9
        SPACE 1
        BR   R5                 RETURN TO CALLER
        SPACE 2
*****
*          LEAVE TRANSMITTER EXIT                                *

```

## Transmitter Exit Example

```

*****
      SPACE 2
TEXEND  DS   0H
        BR   RE           RETURN TO CALLER
        SPACE
        EJECT
*****
*      SUBROUTINE TO ISSUE MESSAGE ON OPERATOR CONSOLE      *
*                                                           *
*  USAGE OF VSE/POWER MACRO IPW$WTO:                      *
*  THIS MACRO, TOGETHER WITH SOME INDICATIONS SET IN THE TCB, *
*  CAN BE USED TO DISPLAY A MESSAGE ON THE CENTRAL OPERATOR *
*  CONSOLE.                                                *
*  THE USAGE OF THIS MACRO DESTROYS THE REGISTER R0 - R3.  *
*  THE MESSAGE TO BE DISPLAYED SHOULD CONTAIN ALPHAMERIC  *
*  CHARACTERS ONLY.                                       *
*                                                           *
*****
      SPACE 2
SUBWTO  DS   0H
        STCM R7,7,TCMW+1      PASS MESSAGE ADDRESS
        MVI  TCMW,X'00'      CLEAR OPTION BYTE
        OI  TCMW,TCPCOP      FORCE MESSAGE TO CONSOLE
        OI  TCMW,TCDNMM      SUPPRESS MESSAGE MODIFICATION
        SPACE
        IPW$WTO TCMW        ISSUE MESSAGE
        SPACE
        MVI  TCMW,X'00'      CLEAR OPTION BYTE
        BR   R6             RETURN TO CALLER
        SPACE 5
        DROP R8,R9,RA,RB    RELEASE ADDRESSABILITY
*                               OF PARAMETER LIST, WORK AREA,
*                               CAT AND TCB
        EJECT
*****
*
*      DATA DEFINITIONS
*
*****
      SPACE 2
*****
*
*      MESSAGES
*
*  GENERAL FORM:
*
*  TASK-ID|TO-NODE|ENTRYNAME: .....
*  |<---4--1---8--1---8---3>|<-----40----->|
*  |<-----25----->|<-----MESSAGE----->|
*  |<---MESSAGE PREFIX--->|<-----MESSAGE----->|
*
*****
      SPACE 1
*      LAYOUT OF MESSAGE PREFIX
      SPACE 1
MES     DC   AL1(MESL)      LENGTH OF MESSAGE
MTID    DC   CL4' '        TASK ID
        DC   C'|'          SEPARATOR
        DC   CL8' '        EXECUTION NODE ID
        DC   C'|'          SEPARATOR
        DC   CL8' '        QUEUE ENTRY NAME
        DC   CL3': '      SEPARATOR
        DC   CL40' '      MESSAGE FIELD
MESL    EQU  *-MTID       LENGTH OF MESSAGE DATA
        DS   0F
TXMSG1  DC   AL1(TXMSG1-TXMSG1-1)
TEXMSG1 DC   CL40'XMTEXIT: WORK AREA SIZE NOT CORRECT !'

```

## Transmitter Exit Example

```
TXMSGL1 EQU *
TXMSG2 DC AL1(TXMSGL2-TXMSG2-1)
TEXMSG2 DC CL40'XMTEXIT: 670 BYTES OF WORK AREA NEEDED'
TXMSGL2 EQU *
TXMSG3 DC AL1(TXMSGL3-TXMSG3-1)
TEXMSG3 DC CL40'XMTEXIT: LOAD XMTEXIT AGAIN USING PLOAD'
TXMSGL3 EQU *
TXMSG4 DC AL1(TXMSGL4-TXMSG4-1)
TEXMSG4 DC CL40'XMTEXIT: WORK AREA NOT SPECIFIED !'
TXMSGL4 EQU *
TXMSG7 DC AL1(TXMSGL7-TXMSG7-1)
TEXMSG7 DC CL40'XMTEXIT: XMISSION OF OUTPUT NOT ALLOWED'
TXMSGL7 EQU *
TXMSG9 DC AL1(TXMSGL9-TXMSG9-1)
TEXMSG9 DC CL40'COMMENT RECORD INSERTED AFTER JHR'
TXMSGL9 EQU *
TXMSGA DC AL1(TXMSGLA-TXMSGA-1)
TEXMSGA DC CL40'NEW WORK AREA REQUIRED (NNNNN BYTES) ! '
TXMSGLA EQU *
SPACE 2
*****
*
*           U S E R   S E C T I O N
*           OF THE JOB HEADER RECORD
*
*****
SPACE 1
NJHULN DC AL2(NJHULL)          LENGTH OF USER SECTION
DS 0AL2                      USER SECTION FLAG
DC AL1(NTYPU)                 ID FOR USER SECTION
DC AL1(NJH$M)                 MODIFIER
NTYPU EQU B'11000000'         VALUE FOR USERSECTION ID
NJH$M EQU B'00000000'         VALUE FOR MODIFIER
NJHUDATA DC C'+-----+
DC C'|
DC C'|         >> DEPENDS ON YOUR CHOICE <<
DC C'|         >> DEPENDS ON YOUR CHOICE <<
DC C'|
DC C'+-----+
NJHULL EQU *-NJHULN          LENGTH OF USER SECTION
SPACE 2
*****
*
*           L O C A L   C O N S T A N T S
*
*****
SPACE 1
DS 0F
NODEB DC CL8'JEC1'           NAME OF NODE B
TEXIREC DC CL80'* >>>> USER SECTION APPENDED TO JOB HEADER <<<<'
TEXIRECL DC A(L'TEXIREC)     LENGTH OF TEXIREC
SPACE 2
EJECT
*****
*
*           L A Y O U T   O F   T H E
*           T R A N S M I T T E R   E X I T   W O R K   A R E A
*
*****
SPACE 1
WADSECT DSECT              BEGIN OF WORK AREA
WALNGTH DS F                LENGTH OF WA
WAJHRL DS F                LENGTH OF EXTENDED WORK AREA
WAFLG DS X                 RECORD INSERTION INDICATOR
WAFLG2 DS X                 USER SECTION FLAG BYTE
WAFLG3 DS X                 FIRST TIME FLAG BYTE
DS X                       NOT USED
```

## Transmitter Exit Example

```
WAEXND DS CL8 NAME OF EXECUTION NODE
SPACE 2
* FIELDS WHICH CONTAIN THE POINTER AND THE OFFSET OF THE
* JOB DISPOSITION FIELD WITHIN THE VSE/POWER SECTION OF THE JHR
SPACE 1
WAPWRPT DS F POINTER TO VSE/POWER JOB DISP.
WAPWROF DS F OFFSET OF VSE/POWER JOB DISP.
SPACE 2
* LAYOUT OF MESSAGE TO BE ISSUED
SPACE 1
WAMES DS AL1 LENGTH OF MESSAGE
WAMTID DS CL4 TASK ID
DS C SEPARATOR
WAMEX DS CL8 EXECUTION NODE ID
DS C SEPARATOR
WAMQEN DS CL8 QUEUE ENTRY NAME
DS CL3 SEPARATOR
WAMFIELD DS CL40 MESSAGE FIELD
WAOUTF DS D FIELD FOR DATA CONVERSION
WALN EQU *-WADSECT LENGTH OF WORK AREA IN FRONT OF *
JHR SAVE AREA

SPACE 2
* THE FOLLOWING SPACE IS USED TO CONTAIN ALL POSSIBLE SECTIONS
* OF THE JOB HEADER RECORD.
SPACE 1
WANJHR DS 0F BEGIN OF JHR SAVE AREA
DS 4X ..SPACE FOR PREFIX SECTION
DS 0F ..BEGIN OF JHR SECTIONS
ORG ++562 ..SPACE FOR JHR SECTIONS
WAJHRLN EQU *-WANJHR LENGTH OF JHR WORK AREA
WALEN EQU *-WADSECT LENGTH OF ENTIRE WORK AREA
EJECT

*****
* LAYOUT OF VSE/POWER CONTROL BLOCKS AND NETWORK CONTROL RECORDS *
*****

SPACE 2
IPW$DTX TRANSMITTER EXIT PARAMETER LIST
SPACE 2
IPW$EQU , REGISTER EQUATES
SPACE 2
IPW$DPA , VSE/POWER PERMANENT AREA (CAT)
SPACE 2
IPW$DTC NET=YES VSE/POWER TASK CONTROL BLOCK
SPACE 2
IPW$DQC , VSE/POWER DISK MANAGEMENT BLOCK
SPACE 2
IPW$DQR , LAYOUT OF SPOOL QUEUE RECORD
* (ADDRESSED BY TCB.TCQV)

SPACE 2
IPW$DPN , VSE/POWER PNET MAST.CNTRL.BLOCK
SPACE 2
IPW$DNR JHR=YES,JTR=YES,DHR=YES,OUT=YES NETWORK CONTROL RECORDS
SPACE 2
IPW$DNC , VSE/POWER NODE CONTROL BLOCK
SPACE 2
* END NOT REQ. FOR '.A' COPY BOOK
```

End of Programming Interface Information

## Transmitter Exit Example

---

## Appendix D. PNET BSC/CTCA/TCP/SSL Console Trace

The console trace can be used to record the order of I/O events that take place between the two nodes using a BSC or CTCA line or a TCP/IP connection. It can then be used to check whether one or both nodes are following the protocol.

**Note:** The console trace can optionally be written directly to a printer by specifying the printer address 'cuu' in the PSTART CNSLTR command (see *VSE/POWER Administration and Operation* and "Console Trace Commands" on page 108).

For a TCP/IP connection the implementation produces console trace messages in two formats, the normal format for line I/O OUT:/IN: (LINE=TCP is displayed) and the format for TCP/IP socket communication (see "Additional TCP Console Trace Message" on page 108). Also the components EZASMI API and TCP/IP for VSE/ESA may issue their own console trace messages (see "Additional TCP Component Console Trace Messages" on page 115).

The following information is recorded on the central operator's console at I/O start

```
1RTTI LINE=xxx START-IO OUT:BSC=aaaa BCB=bb FCS=ffff RCB=rrrr nodename
```

followed by the console line at I/O completion

```
1RTTI LINE=xxx CCB=cccc IN: BSC=aaaa BCB=bb FCS=ffff RCB=rrrr nodename
```

where:

**LINE=xxx**

For a BSC or CTCA connection indicates the two-byte 'cuu' device address in binary format. Otherwise, displays 'TCP' or 'SSL', indicating that the TCP/IP connection is traced.

**CCB=cccc**

are the CSW status bits at I/O completion.

**SENSE=ssss**

is the two-byte sense information as returned from the device.

**Note:** SENSE is optional. It is displayed only for a BSC connection when the CSW status bits indicate 'Unit Check'.

**OUT or IN**

is the direction of the buffer.

**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
- X'323D'- SYN NAK

**BCB=bb**

is the block control byte either sent or received.

## PNET BSC/CTCA/TCP/SSL Console Trace

**FCS=ffff**

are the two bytes of the function control sequence field.

**RCB=rrrr**

is the record control byte (RCB) and subrecord control byte (SRCB) of the first record in the buffer sent or received.

**nodename**

is the nodeid of the connection being served.

**Note:** The trace lines produced in this format represent typical BSC write/read command sequences or the corresponding CTC or TCP or SSL (which is initiated from the CTC layer) sense-write/control-read command sequences. Non-typical command sequences not accompanied by IN/OUT data buffers may appear at:

- line start, and may read for a CTC line, for example:

```
1RTTI LINE=E1D START-IO OUT: SPECIAL! SEQUENCE, CCW OPCODE=14 POWER251
```

In this case, the operation code of the first (chained) CCW is displayed.

- line termination, and may read for a CTC line, for example:

```
1RTTI LINE=E1D CCB=0C00 IN: LAST CCW NOT READ, CCW OPCODE=01 POWER251
```

In this case, the operation code of the last (chained) CCW is displayed.

---

## Console Trace Commands

The console trace can be initiated while VSE/POWER is running, by means of the two command formats:

```
PSTART CNSLTR[, ,PNETxxx]  
PSTART CNSLTR, cuu[, PNETxxx]
```

where:

**cuu** is the printer address to be assigned to SYSLST. The trace will then appear on SYSLST instead of on the console. Also the trace messages of the EZASMI API and of the TCP/IP for VSE/ESA program for the VSE/POWER partition will be directed to SYSLST as well, provided that the SYSLST printer is assigned to the VSE/POWER partition. See "Routing TCP Console Trace Messages to SYSLST" on page 114 for more details.

**PNETxxx**

Indicates the specified type of PNET connection to be traced. The types are "PNETBSC", "PNETCTC", "PNETTCP", and "PNETSSL".

To terminate the console trace and remove potential assignment of SYSLST, enter the following command:

```
PSTOP CNSLTR
```

---

## Additional TCP Console Trace Message

For PNET connections with a TCP or SSL node, another console trace message format is presented, in addition to the line I/O OUT:/IN: message, to display TCP/IP information.

Whereas the line I/O OUT:/IN: console trace message is issued by the VSE/POWER maintask, the TCP or SSL console trace messages are issued by the TD or SD subtask running as a separate z/VSE subtask. Therefore, these messages



## Additional TCP Console Trace Message

are displayed on the console with a z/VSE subtask identifier that is different from the task identifier of the VSE/POWER maintask.

The TCP and SSL console trace messages have the following layout:

```
1RTTI xD-d: ....(rest of trace information) .....
```

where xD identifies the module (TD for IPW\$\$TD and SD for IPW\$\$SD) and d is the step within the module that issues the trace message.

The step identifier d is

1. a numeric character when a socket call is being processed
2. an alphabetic character when
  - a. data has been received or sent via a socket call
  - b. an I/O request is processed
  - c. a particularly unusual event occurred

### Numeric Step Identifier

The following message format is used to trace processing of a socket call:

```
1RTTI TD-d: SOK=nnnn socketcall ,RC=srrrrrrr,ERRNO=seeeeeee, TS-RC=tt,mode
      node-id          status-info
or
1RTTI SD-d: SOK=nnnn socketcall ,RC=srrrrrrr,ERRNO=seeeeeee, TS-RC=tt,mode
      node-id          status-info
```

where:

**d** is the 'step identifier'.

Whenever the step identifier has the value 1, 2, 3, or 4, the message was issued to trace a socket call. The meaning is as follows:

- '1' means the socket call will be issued immediately
- '2' means the socket call was issued but was not immediately completed (the ECB has not yet been posted)
- '3' means the socket call has been completed (the ECB has been posted)
- '4' means that, when issuing the socket call, an error occurred that terminates the socket call immediately

#### **SOK=nnnn**

stands for socket call descriptor and is followed by a value which has been assigned by the TCP/IP component. One value is assigned for each TCP/IP connection, which means each TCP type node gets its own socket descriptor number. One socket descriptor (usually 0) is assigned for a socket used for incoming connection requests (processed in 'passive mode').

#### **socketcall**

is the name of the socket call which is processed. The following socket calls are used and may be traced:

- INITAPI
- TERMAPI
- SOCKET
- GETHOSTID
- BIND
- LISTEN

## Additional TCP Console Trace Message

- CONNECT
- ACCEPT
- SEND
- RECEIVE
- CLOSE
- CANCEL
- GETHOSTBYADDR
- GETHOSTBYNAME
- SELECT-READ
- SELECT-WRITE
- SSL-INIT
- SSL-UNINIT
- SSL-GETDNAME
- SSL-FREEMEM
- SSL-SOK-INIT
- SSL-SOK-READ
- SSL-SOK-WRITE
- SSL-SOK-CLOSE
- SSL-SOK-RESET
- SSL-GETCIPHER
- IOCTL

### **RC=srrrrrrr**

the return code for the socket call in decimal format as received by the TCP/IP product, where 's' stands for the sign, which is either "+" or "-" or omitted.

In case of step identifier '1' the following has to be considered:

- The values have not been reset and may contain a value of the previous socket call.
- If it is a SEND or RECEIVE socket call, the data length used for the socket call. In case of succesful SEND or RECEIVE, the RC value is equal to the number of bytes sent or received and the ERRNO value is zero. These values are set by TCP/IP for VSE/ESA.
- If it is a CANCEL socket call, RC contains the address of the request to be cancelled.

If the sign is omitted, rrrrrr displays a hexadecimal value which is:

1. the IP-address of the local node for the socket call GETHOSTID
2. a pointer to a control block for the socket calls SSL-GETDNAME and SSL-SOK-INIT.

### **ERRNO=seeeeeee**

the error number for the socket call as received by TCP/IP for VSE/ESA, where 's' stands for the sign, which is either + or - or blank. In case of step identifier '1' the following has to be considered:

- The values have not been reset and may contain a value of the previous socket call.

### **TS-RC=tt**

The return code passed from module IPW\$\$TS to module IPW\$TD.

### **SS-RC=tt**

The return code passed from module IPW\$\$SS to module IPW\$\$SD.

**mode** identifies the TCP/IP processing mode, which distinguishes who actually issued a CONNECT or who issued the ACCEPT. Values are:

## Additional TCP Console Trace Message

- C** means local node is acting as client (after the initial contact has been completed).
- A** means connection is in initial state and is processing in TCP/IP 'active mode'.
- P** means connection is in initial state and is processing in TCP/IP 'passive mode'
- S** means local node is acting as server (after the initial contact has been completed)

### node-id

The following may be displayed:

- the nodeid of the connection being processed.
- the 15 character dotted-decimal IP address if the nodeid is not yet known for the connection.
- blanks if neither a node-id nor an IP-address is available.

### status-info

the following may be displayed:

- Any hexadecimal characters for most socket calls.

#### s1s2s3s4 s5

5 status bytes concerning the TCP/IP connection (starting at ITPST1 within the TCP/IP workarea of the Node Control Block)

#### s6s7s8

- 3 status bytes concerning the socket calls:
  - RECEIVE (ITPSCST1) or any socket call other than one of the following
  - SEND (ITPSCSS1)
  - CANCEL (ITPSCCS1)
- If the initial contact is not yet complete:
  - If it is a SEND request and step identifier '1': data to be sent
  - If it is a RECEIVE request and step identifier '3': data which has been received.
- If it is a CANCEL socket call and step identifier '1', the following is displayed:
  - 'SEND' if a SEND request is to be cancelled.
  - 'RECV OR OTH' if the request to be cancelled is not a SEND.
  - an address of a workarea which could not be identified. This is probably an error.

## Alphabetic Step Identifier

When 'd' is alphabetic, the following message formats are possible:

### Step identifier 'C'

CTC I/O data.

Data within the TCP/IP receive buffer, which will be moved into the CTC I/O read buffer, for example:

```
1RTTI TD-C: RECEIVED FROM NODE node-id: 00000006 1002808F CF000000
00000000 PB=00000000
```

## Additional TCP Console Trace Message

### Notes:

1. The first 4 bytes contain the TCP/IP specific TTR, which contains the length of the CTC I/O data. The CTC I/O data itself starts with the fifth byte.
2. The BCB in the TCP/IP receive buffer may be different from the BCB in the CTC I/O buffer that is displayed later on by the PNET line driver using keyword 'BCB='.
3. After the constant PB= the number of not yet processed bytes in the TCP/IP buffer is displayed.

### Step identifier 'F'

Wait-a-bit situation.

This happens if the VSE/POWER maintask sends an indication to the remote node that the local node can no longer receive data. This indication is signaled by 2 FCS bytes suspending either all or selected receivers. In both cases, no further RECEIVE socket call is issued until the maintask resets the FCS bytes to signal that data can be received again. Usually this happens if no receive buffers are available (see the MAXBUF operand of the PNODE macro in *VSE/POWER Administration and Operation*). In most cases, this wait-a-bit situation is resolved within seconds.

Such a message can appear as follows:

```
1RTTI TD-F: FCS SENT TO NODE node-id: 87CF
```

### Step identifier 'I'

Initial trace message.

Whenever a trace is started, this message appears once to identify the versions of the modules taking part in the TCP/IP subtask, for example:

```
1RTTI TD-I: IPW$TD VERSION 07/12 RUNNING WITH IPW$TS VERSION 07/12
```

### Step identifier 'M'

Moved data within TCP/IP buffer.

Data has been moved from the TCP/IP buffer to the start of the TCP/IP buffer. This happens if the RECEIVE socket call did not receive all data and another RECEIVE has to be issued to get the rest of one TCP/IP data record, for example:

```
1RTTI TD-M: MOVED FROM NODE node-id: 00000009 00000000 00000009 00000000
```

**Note:** 16 bytes are displayed regardless of how many bytes have been moved. In most cases, TCP/IP specific control information (TTB and TTR) has been moved, which usually means just 12 or 4 bytes. Thus, the first bytes may be the same as some of the last bytes as in the above sample.

### Step identifier 'P'

CCB with completion conditions.

Whenever an I/O request initiated on the CTC layer has been completed by data to be returned from the TCP layer to the CTC layer, the CCB (Command Control Block) is displayed with the I/O completion conditions as simulated by the TCP layer, such as:

## Additional TCP Console Trace Message

```
1RTTI TD-P: POWER254: CCB=0FFA9400 0C000000 00511AB8 LCCW=00511AD8
      ST=60010080 48010000 8C207000 0000 07
```

In addition, the following information is displayed:

1. After the constant LCCW=, the address of the last processed CCW is displayed. This address is the address of the last CCW incremented by 8 and therefore points behind the CCW.
2. After the constant ST=, the following status information is displayed:
  - a. 8 general status bytes concerning the connection (starting at NCBTYP within the NCB)
  - b. 6 status bytes concerning the TCP/IP connection (starting at NCBTTPST1 within the NCB)
  - c. 1 sense byte which is passed for the SENSE CCW and is very important for the initial stand-alone SENSE CCW.

### Step identifier 'R'

Data of RECEIVE socket call.

In addition to the trace message with a numeric step identifier describing the RECEIVE socket call, the received data is displayed in an extra message, for example:

```
1RTTI TD-R: D5C1D240 40404040 D7D6E6C5 D9F2F5F4 09A49B44 C2D6C5E5
      D4C3E3F1 09A4BA4B 02F1D9E3
```

### Step identifier 'S'

Data of SEND socket call.

In addition to the trace message with a numeric step identifier describing the SEND socket call, the data to be sent is displayed in an extra message, for example:

```
1RTTI TD-S: D5C1D240 40404040 D7D6E6C5 D9F2F5F4 09A49B44 C2D6C5E5 D4C3E3F1
      09A4BA4B 02F1D9E3
```

### Step identifier 'W'

Up to 4 CCWs of the start I/O request.

Whenever an I/O request has been initiated on the CTC layer and passed to the TCP layer, all CCWs (Command Control Words) with their operation codes and pointers to the data buffers are displayed, such as:

```
1RTTI: TD-W: CCW'S FOR NODE node-id: 1441070000B60000001 015694B460000006
      07000000600000001 02573C3420001000
```

In some cases (when starting or stopping a connection) the I/O request consists of fewer than 4 CCWs. Nevertheless, four groups of 32 characters in storage are always displayed.

### Step identifier 'X'

Arrays of socket call SELECT-READ

When a socket call SELECT-READ completes successfully with a return code > 0, the first parts of the read arrays are displayed. After the constant 'SD-X:' the first

## Additional TCP Console Trace Message

fullwords of the array used as input parameter are displayed. After the constant 'ARRAY-O:' the first fullwords of the array used as output parameter are displayed.

Each socket descriptor is represented by a bit in a bit string. The bit strings are grouped together in 32-bit fullwords: the first fullword contains the bits for the sockets 0-31, the second fullword the bits for sockets 31-63, the third fullword the bits for sockets 64-95, the fourth fullword the bits for sockets 96-127. Within each fullword, the bits are numbered from right to left.

In the sample below the first fullword contains 00000003. This means that the sockets 0 (rightmost bit) and 1 (second bit from right) are used by the socket call SELECT as input. Since the first fullword of the output array contains 00000002, only socket 1 is ready to be processed.

```
1RTTI SD-X: 00000003 00000000 00000000 00000000 ARRAY-O: 00000002
00000000 00000000 00000000
```

### Sample console log

The following group of trace messages resulted from a sense-write-data/control-read-data I/O requests to the TCP node 'POWER253':

```
F1 0001 1RTTI LINE=TCP START-IO OUT:BSC=1002 BCB=81 FCS=8FCF RCB=9A80 POWER253
F1 0101 1RTTI TD-W: CCW'S FOR NODE POWER253: 1441070B60000001
0160703460000025 0700000060000001 025F103420007D00
F1 0101 1RTTI TD-1: SOK=0928 SEND ,RC=+0000053,ERRNO=+0000000,
TS-RC=00,A POWER253 842C8000 0000 808000
F1 0101 1RTTI TD-S: 00000035 00000000 00000025 1002858F CF9A80CC 80770003
D7D6E6C5 D9F2F5F3 A900CCD7
F1 0101 1RTTI TD-2: SOK=0928 SEND ,RC=+0000000,ERRNO=+0001103,
TS-RC=00,A POWER253 842C8000 0000 808000
F1 0101 1RTTI TD-3: SOK=0928 SEND ,RC=+0000053,ERRNO=+0000000,
TS-RC=00,A POWER253 84AC8000 0000 800000
F1 0101 1RTTI TD-P: POWER253: CCB=7CFA9400 0C000000 00410738 LCCW=00410758
ST=60000080 86010000 842C6000 0000 07
F1 0001 1RTTI LINE=TCP CCB=0C00 IN: BSC=1002 BCB=82 FCS=8FCF RCB=0000 POWER253
F1 0001 1RTTI LINE=TCP START-IO OUT:BSC=1002 BCB=82 FCS=8FCF RCB=0000 POWER253
F1 0101 1RTTI TD-W: CCW'S FOR NODE POWER253: 1441070B60000001
0150223460000006 0700000060000001 025F103420007D00
F1 0101 1RTTI TD-3: SOK=0928 RECEIVE ,RC=+0001083,ERRNO=+0000000,
TS-RC=00,A POWER253 842C8000 0000 000000
F1 0101 1RTTI TD-R: 0000043B 00000000 0000042B 1002868F CF9A80CD 00000420
D7D6E6C5 D9F2F5F4 40A800CE
F1 0101 1RTTI TD-C: RECEIVED FROM NODE POWER253: 0000042B 1002868F CF9A80CD
00000420 PB=00000000
F1 0101 1RTTI TD-P: POWER253: CCB=78D59400 0C000000 00410738 LCCW=00410758
ST=60000080 86010000 842C5000 0000 07
F1 0001 1RTTI LINE=TCP CCB=0C00 IN: BSC=1002 BCB=83 FCS=8FCF RCB=9A80 POWER253
:
```

**Note:** The layout of the above messages may be changed for future enhancements.

## Routing TCP Console Trace Messages to SYSLST

If TCP console trace messages are routed to SYSLST, the following has to be observed:

- The console trace messages do not contain the z/VSE subtask identifier.
- The console trace messages contain some time information at the end, a readable character part and a hexadecimal part, for example:

## Additional TCP Console Trace Message

```
1RTTI TD-1: SOK=0000 SOCKET ,RC=+0000000,ERRNO=+0000000,TS-RC=00,P
          00000000 00 800000 12:53:35 D0E5C3 4E22
```

The hexadecimal part (D0E5C3 4E22 in the example) consists of two groups. The first group contains the second, third and fourth bytes of the output of the STCK instruction, and the second group contains the fifth and sixth bytes. These values can be used to identify the trace entry within the storage trace area, which corresponds to the trace message and contains some more trace information.

- In addition to the console trace messages, other messages issued by the TCP driver subtask are displayed as well. These messages do not contain any time information, for example:

```
1RTNI TCP/IP: CONNECTION CLOSED FOR NODE POWER254 DUE TO STOP COMMAND
          (DURING INITIAL CONTACT)
```

---

## Additional TCP Component Console Trace Messages

Additional console or SYSLST tracing information can be obtained for the TCP/IP internal interfaces used by VSE/POWER PNET:

- EZASMI application interface trace messages, which can be initiated by the following Attention Routine command:

```
EZAAPI TRACE=ON[,PART=partition_id][,SYSLST]
```

For more information on the EZASMI trace, issue the attention routine command:

```
EZAAPI ?
```

- TCP/IP for VSE/ESA BSD/C program flow trace messages

When starting the EZASMI console trace, the TCP/IP for VSE/ESA program trace will also be activated. For additional information, refer to the *TCP/IP for VSE Programmers Reference* or the informational APAR II11836.

## Additional TCP Console Trace Message



---

## Appendix E. PNET Telecommunication Trace

VSE/POWER provides a combined I/O and buffer trace which helps the system programmer or IBM service personnel to find the cause of a failure.

With the operator's console log and a dump, the output from the telecommunication trace area provides enough information to locate internal PNET problems more easily and permits the reconstruction of I/O sequences.

A trace record is written for the following events:

- PNET BSC/CTCA I/O completed
- PNET SNA SEND or RECEIVE request completed
- PNET SNA RECEIVE request completed
- PNET TCP (any) socket call started, tested, or completed
- PNET SSL (any) socket call started, tested, or completed

The trace records generated by VSE/POWER are recorded in wraparound mode in main storage. The amount of storage allocated for the trace area is specified in the TRACESZ=xxx (default 24KB) parameter of the POWER macro at generation time. The value specified for TRACESZ should ensure that information is not destroyed because of wrapping in the trace area, unless trace area dumping is requested.

The operator activates the trace recording for a PNET line by specifying the TRACE operand in the PSTART command. The trace recording remains active until the connection to the other node is stopped.

Every trace record is 256 bytes long. These are the possible types of trace records:

- PNET BSC/CTCA trace record, described in "Format of the PNET BSC/CTCA Trace Record" on page 118.
- PNET SNA SEND/RECEIVE trace record, described in "Format of the PNET SNA Trace Record" on page 119.
- PNET TCP or PNET SSL socket call trace record, described in "Format of the PNET TCP and PNET SSL Trace Records" on page 120.

The trace records can be examined by displaying or taking a dump of the main storage location containing the trace area.

The telecommunication trace area is divided into two parts, referred as primary and alternate trace area. Both trace areas have the same size (integer number of page). When the primary trace area is full, VSE/POWER automatically swaps to the alternate trace area and starts filling that up. If now the alternate trace area is filled up, the primary trace area is addressed again and used for recording.

Optionally, whenever the trace area is full, it is written onto the VSE DUMP library. The operator is informed when a trace area has been successfully written into the user defined DUMP library. The dump members can then be printed by the appropriate VSE utility. Trace area dumping logging is requested by entering the PSTART DUMPTR command, and can be halted by PSTOP DUMPTR.

The trace area can be found by first locating the trace information block (TIB). The trace information block contains among others a pointer to a two-word trace area

## PNET Telecommunication Trace

descriptor of the trace area part currently in use at displacement X'14'. The first word of the trace area descriptor contains the starting address of the trace area part while the second word contains the ending address. The pointer to the next free trace area entry is stored at displacement X'50' of the TIB. The PDISPLAY TRINFO command can be used to get the start and end address of the entire telecommunication trace area as well as the next free trace entry address; for example:

```
IR46I TELETR: BEGIN=003D1000 END=003D5FFF CURR=003D1000
```

---

## Format of the PNET BSC/CTCA Trace Record

Bytes (Hex)	Data Length	NCB Field	Data Description
00-0F	16	NCBCCB	CCB
10-37	40	NCBCCW	Last 5 executed CCWs
38-3F	8	-	Not used
40-5F	32	-	Send buffer content (first and last 16 bytes)
60-7F	32	-	Receive buffer content (as above)
80-81	2	NCBSNS1/2	Sense bytes 1 and 2
82-83	2	-	Not used
84-87	4	-	Last executed CCW + 8
88-8B	4	-	Time of day (0HHMMSSF)
8C-8F	4	-	Not used

### Status of communication:

90-93	4	NCBIBUF	Address of input buffer in use
94-97	4	NCBOBUF	Address of output buffer being sent
98-9B	4	NCBCBFI	Address of buffer used by read CCW
9C-9F	4	NCBCBFO	Address of buffer used by write CCW
A0	1	NCBEBCB	Expected block sequence count (BCB)
A1	1	NCBTBCB	Transmitted block sequence count (BCB)
A2-A3	2	NCBRFCS	Received FCS bytes
A4-A5	2	NCBTFCS	Transmitted FCS bytes
A6	1	NCBREQF	Request code for I/O manager
A7	1	NCBLREQ	Last sent request code
A8	1	NCBNAK	Last non-NAK request sent
A9	1	NCBLDRQ	Request from line-driver
AA	1	NCBIOF1	Interface flags I/O manager
AB-AF	5	-	Not used

### Error counts:

B0-B3	4	NCBXCNT	Total transmission count
B4-B5	2	NCBTCNT	Total timeout count
B6-B7	2	NCBECNT	Error count
B8	1	NCBRCNT	Retry count
B9	1	NCBTIMC	Timeout retry count
BA	1	NCBTERR	Error count
BB-BF	5	-	Not used

### Status of node:

C0-C1	2	NCBACT1/2	Action bytes 1, 2
C2	1	NCBPROC	Process byte
C3-C4	2	NCBFLG1/2	Status bytes 1, 2
C5	1	-	Reserved
C6	1	NCBTTC	Termination code
C7	1	NCBTTCDE	Termination subcode
C8	1	NCBTTSTQ	Stop code qualifier
C9-EF	39	-	Not used
F0-F7	8	NCBNAME	Node-id
F8-FB	4	TCBTI	Task ID of task writing trace record
FC	1	-	Not used
FD-FF	3	NCBDVAD	Line address (cuu)

The following illustrates the layout of this trace record:

```

xxxxxx00 |-----CCB-----| |---1st-CCW---| |---2nd-CCW---|
          xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          |---3rd-CCW---| |---4th-CCW---| |---5th-CCW---| |---unused----|
xxxxxx20 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          |--data-begin-of-send-buffer-----| |data-end-of-send-buffer-(see-1)-|
xxxxxx40 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          |--data-begin-of-receive-buffer---| |data-end-of-rcv-buffer--(see-1)-|
xxxxxx60 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          SNS-|--| last-ccw |HHMMSS| |-----| |--buffer-addresses-----|
xxxxxx80 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          |-----counts-----| |--unused-| |-----counts-----| |--unused-|
xxxxxxA0 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          |flags-----| |--unused-----|
xxxxxxC0 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
          -----| |-TCBTI| |-cuu|
xxxxxxE0 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
    
```

1) If data length is 16 bytes or less, 'data-end' is dropped.

---

## Format of the PNET SNA Trace Record

Bytes (Hex)	Data Length	NCB/BUFFER Field	Data Description
00-07	8	NCBNAME	Remote node name
08-1F	24	BUFHDR	Buffer header
08-0B	4	BUFNEXT	Next buffer in chain
0C-0F	4	BUFREAL	Real address of buffer (empty for SNA)
10-13	4	BUFLOWN	Address of related TCB
14-17	4	BUFNCBA	Address of related NCB
18-19	2	BUFDATL	Byte count to send or received
1A-1B	2	BUFSZ	Buffer size (excluding header)
1C	1	BUFSTAT	Buffer status byte
1D	1	BUFEST1	Data Stream Status
1E-1F	2	-	Reserved
20-83	100	BUFRPL	VTAM RPL
84-87	4	-	Time of day (0HHMMSSF)
88-97	16	-	First 16 bytes of data
98-A7	16	-	Last 16 bytes of data

### Status of node:

A8-A9	2	NCBACT1/2	Action bytes 1, 2
AA	1	NCBPROC	Process byte
AB-AC	2	NCBFLG1/2	Status bytes 1, 2
AD	1	-	Reserved
AE	1	NCBTTC	Node termination code
AF	1	NCBTTCDE	Termination subcode
B0	1	NCBTTSTQ	Termination code qualifier
B1-B3	3	-	Reserved
B4-B5	2	NCBCNRV	Number of receivers currently active
B6-B7	2	NCBCNTR	Number of transmitters currently active
B8-BB	4	NCBIFRE	Address of free input buffer queue
BC-BF	4	NCBOTBS	Address of to-be-sent queue
C0-C3	4	NCBOBTL	Tail pointer to-be-sent output queue

## PNET Telecommunication Trace

C4-C7	4	NCBOTBP	Address of receive buffer
C8-CB	4	NCBOBTP	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	NCBIBUF	Address of input buffer in use
D8-DB	4	NCBOBUF	Address of output buffer sent
DC-DF	4	NCBCBFI	Address of buffer for actual receive
E0-E3	4	NCBCBFO	Address of buffer for actual send
E4-E5	2	NCBRFCS	Remote Held/Released Stream Status
E6-E7	2	NCBTFCS	New Held/Released Stream Status
E8-EB	4	NCBPBFRQ	Address of suspended buffer queue
EC-EF	4	NCBIFREX	Address free input buffer ahead queue
F0-F3	4	NCBOTBSX	Address send ahead queue
F4	1	NCBSGTE	Send gate
F5	1	NCBRGTE	Receive gate
F6-F9	4	NCBSFL1/4	Session status flag bytes 1 to 4
FA	1	NCBSEST	Session status byte
FB	1	-	Reserved
FC-FF	4	TCBTI	Task ID of task writing trace record

The following illustrates the layout of this trace record:

```

|--NCBname-----| BUFNEXT  BUFREAL  BUFOWN  BUFNCBA  DATL|SZ| STAT
xxxxxx00 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

|---VTAM-RPL-FROM-BUFFER-----
xxxxxx20 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

----VTAM-RPL-FROM-BUFFER----(continued)-----
xxxxxx40 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

----VTAM-RPL-FROM-BUFFER----(continued)-----
xxxxxx60 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

-----| |HHMSS| |--data-begin-of-send/rcv-buffer--| |-data-end-of-s/r
xxxxxx80 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

-buffer-(see-1)-| |-flags-----|unused #RCV#XMT NCBIFRE  NCBOTBS
xxxxxxA0 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

NCBOBTL  NCBOTBP  NCBOTBP  size          NCBIBUF  NCBobuf  NCBCBFI
xxxxxxC0 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

NCBCBFO  FCS-stat  NCBPBFRQ  NCBIFREX  NCBOTBSX  gate|status-flgs| |-TCBTI|
xxxxxxE0 xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

```

1) If data length is 32 bytes or less, then complete data is shown.

---

## Format of the PNET TCP and PNET SSL Trace Records

Bytes (Hex)	Data Length	NCB/BUFFER Field	Data Description
00-0C	13		Type of socketcall, for example BIND, SEND, etc.
0D-0F	3		unused
10-17	8		time-stamp as delivered by STCK
18-1E	7	TRCEYE1	eyecatcher 'TCP/IP' or 'TCPSSL'
1F-1F	1	TRCEYES1	identifying stepnumber of socketcall: 1 = before issuing socketcall

## PNET Telecommunication Trace

			2 = socketcall has been issued, but ECB has not yet been posted
			3 = after issuing socketcall and ECB has been posted
			4 = immediately after issuing socketcall
(20-33)	(20)	ITPTPTC2	control info out of TCP/IP workarea:
20-21	2	ITPSCSOD	.. socket descriptor
22-23	2		.. unused
24-25	2	ITPSCBFM	.. addressing family
26-27	2	ITPSCBPT	.. port number
28-2B	4	ITPSCDIP	.. IP-address in binary format
2C-33	8		.. unused
34-37	4		unused
(38-4F)	(24)	ITPTPTC3	control info out of TCP/IP workarea:
38-38	1	SOCSCST1	.. status byte
39-39	1		.. unused
3A-3B	2	SOCSCCNT	retry counter
3C-3F	4	SOCSCBUF	input for socketcall:
			.. buffer address if SEND/RECV
			.. ECB address if CANCEL
40-43	4	SOCSCNBY	input for socketcall:
			.. number of bytes of SEND/RECV
			.. unused if CANCEL
			(if step =1: number to be sent/received)
			(if step =3: number actually sent/received)
44-47	4	SOCSCRC	.. TCP/IP return code of socket call
48-4B	4	SOCSCERN	.. TCP/IP error number
4C-4F	4	SOCSCSAB	.. ECB of socketcall
50-53	4		unused
(54-9F)	(76)		first 76 bytes of send or received data (or less)
54-5F	12		.. TTB and TTR, TCP/IP buffer control info
60-9F	64		.. data according NJE protocol (usually beginning with BSC starting frame =X'1002')
(A0-C3)	(36)	ITPTPTC1	control info out of TCP/IP workarea
A0-A5	6	ITPTPTST	.. status bytes
A6-A7	2		.. unused
A8-B7	16		.. return addresses according return code of IPW\$STS
B8-BB	4		.. return code of IPW\$STS
BC-BC	1		.. socketcall as binary number
BD-BF	3		.. unused
C0-C3	4		.. value of timer interval
C4-CF	12		unused
D0-D7	8	TDCBTRC1	control info out of TDCB
D8-DF	8		unused
E0-EE	15	ITPTPIPC	IP-address in readable form
EF-EF	1	ITPTPTYP	type of connection:
			A = active connection
			C = client
			S = server
			P = passive connection
F0-F7	8	NCBNAME	node-id or 'PASSIVE '
F8-FF	7	TRCEYE2	eyecatcher ' TCP/IP' or ' TCPSSL', same as TRCEYE1
FF-FF	1	TRCEYES2	identifying stepnumber of socketcall, same as TRCEYES1

The following illustrates the layout of this trace record:

```

xxxxxx00 |-----SocketCall-----|Unused |-----Time-----| T C P / I P n
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

xxxxxx20 |-----Socket Info-----|-----Unused-----| Misc SOCSBUF
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

xxxxxx40 No-Bytes SOC-RC SOC-ERR ECB |Unused-|--NJE DATA (TTB+TTR)--...|
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

xxxxxx60 |...-----NJ E Data (cont.)-----...
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

xxxxxx80 ...-----NJ E Data (cont.)-----|
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

xxxxxxA0 |-----TCP/IP Control Info-----...|
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

xxxxxxC0 ...etc--|-----Unused-----|-TDCB Cntl Info--|---Unused-----|
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

xxxxxxE0 |-----IP Addr (Readable) -----|----NCBNAME-----| T C P / I P n
xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx xxxxxxx

```

---

## Appendix F. Examples of Defining TCP/IP Connections

The examples in this appendix show how to connect VSE/POWER to any other VSE/POWER in the World Wide Web. Note, however, that VSE/POWER can also be connected to other systems participating in an NJE-network, as for example, RSCS with TCP/IP support (version RSCS 3.2 or later). The example shows how to:

1. define the TCP/IP configuration file IPINITnn.L
2. define the VSE/POWER network definition table

---

### Defining the TCP/IP for VSE/ESA Configuration File IPINITnn.L

IBM distributes a priced feature called TCP/IP for VSE/ESA, which VSE/POWER uses to communicate with the World Wide Web (WWW). One part of this feature runs as a separate program in its own partition and gets information from a configuration file read in during its initialization (usually cataloged as IPINIT00.L). Following information in this configuration file is necessary for VSE/POWER.

1. 1. The ip-address of the local system which is defined by a SET statement, for example:

```
SET IPADDR=019.009.019.044
```

2. 2. A communication path to the world wide web (www), which is defined by a DEFINE LINK and DEFINE ROUTE statement. Following is a sample for such statements, when the OSA-Express hardware feature is installed.

```
DEFINE LINK, ID=OSA700, TYPE=OSAX, DEV=(700,701), DATAPATH=702, MTU=1500, -  
PORTNAME=F14ABC1  
DEFINE ROUTE, ID=POWERHUS, LINKID=OSA700, IPADDR=0.0.0.0, -  
GATEWAY=019.009.019.1  
START LINKID=OSA700
```

where:

#### **TYPE=OSAX**

identifies the link as an OSA-Express hardware feature

#### **DEV=(700,701)**

identifies the cuu addresses for the OSA-Express feature. The cuu addresses must have been defined during IPL of VSE/ESA using an ADD statement, for example:

```
ADD 700:702, OSAX
```

Always three cuu addresses are used for one OSAX reference. The device numbers in the DEV parameter must be an even/odd pair. The cuu address in the DATAPATH parameter may be even or odd. You may specify for example for another OSAX definition:

```
DEFINE LINK, ID=OSA700, TYPE=OSAX, DEV=(704,705), DATAPATH=703,...
```

If running under VM, ensure that also the real device numbers as generated in the IOCP are an even/odd pair. Otherwise the initialization of the OSAX feature by TCP/IP for VSE/ESA fails indicated by messages like for example:

```
0S39I ERROR DURING OSA EXPRESS PROCESSING, REASON=0011  
0000: IPL605E Unable to Initialize IJBOSA, return code: 121  
0000: IPL609E Unable to initialize OSA Express, Link: OSA700
```

## Sample TCP/IP Connection Definitions

### MTU=1500

identifies the maximum transfer unit used by the feature TCP/IP for VSE/ESA. An unfavorable value may result in bad performance. A favorable value may be retrieved by issuing the DISCOVER command of the TCP/IP for VSE/ESA feature together with the ip-address of the remote system, for example:

```
repl-id DISCOVER 19.09.19.45
```

where repl-id is the id of the message interface of TCP/IP for VSE/ESA, which is established by issuing the attention routine command MSG part-id where part-id is the id of the partition N which TCP/IP for VSE/ESA is running.

---

## Defining the VSE/POWER Network Definition Table

To define a TCP/IP node a Network Definition Table (NDT) must be created by assembling a series of VSE/POWER PNODE macro calls. Following is a sample of the source code for an NDT used at NODE1 for two nodes NODE1 and NODE2:

```
NODE1NDT PNODE NODE=NODE1,LOCAL=YES
          PNODE NODE=NODE2,IPHOSTAD=019.009.019.045
          END
```

The operand LOCAL=YES in the first line is used to define the local node with the name NODE1. The output of the assembly must be cataloged as a phase by using the Linkage Editor. The name of this phase is the label of the first PLINE macro, NODE1NDT in the above sample.

Following is a sample of the jobstream to create an NDT:

```
* $$ JOB JNM=NODE1NDT,CLASS=A,DISP=D
// JOB NODE1NDT
// OPTION CATAL
// LIBDEF PHASE,CATALOG=IJSYSRS.SYSLIB
// EXEC ASSEMBLY,SIZE=100K
NODE1NDT PNODE NODE=NODE1,LOCAL=YES
          PNODE NODE=NODE2,IPHOSTAD=019.009.019.045
          END
/*
// EXEC LNKEDT
/&
* $$ E0J
```

Following is a sample of the source code for an NDT for the two nodes NODE1 and NODE2, which must be created at NODE2:

```
NODE2NDT PNODE NODE=NODE2,LOCAL=YES
          PNODE NODE=NODE1,IPHOSTAD=019.009.019.044
          END
```

For each additional node, another PNODE macro call must be added to the source code, for example:

```
NODE2NDT PNODE NODE=NODE2,LOCAL=YES
          PNODE NODE=NODE1,IPHOSTAD=019.009.019.044
          PNODE NODE=NODE3,IPHOSTAD=019.009.019.046
          END
```

**Note:** No ip-address has to be specified for the local node in the PNODE macro. The ip-address is retrieved from the feature TCP/IP for VSE/ESA (see 'previous chapter') at the time the PNET support is initialized. The VSE/ICCF library 59 provides a skeleton SKPWRNDT for your convenience.



## Sample TCP/IP Connection Definitions

The network definition table is made known to VSE/POWER by specifying its name in the PNET operand of the POWER macro. The POWER macro is assembled and linked (similar to the NDT, see above) to generate a POWER phase. Following is a sample of the jobstream to create a POWER phase:

```
* $$ JOB JNM=POWNODE1,CLASS=A,DISP=D
// JOB POWNODE1
// OPTION CATAL
// LIBDEF PHASE,CATALOG=IJSYSRS.SYSLIB
// EXEC ASSEMBLY,SIZE=100K
POWNODE1 POWER ACCOUNT=YES,.....
                PNET=NODE1NDT,....
                END
/*
// EXEC LNKEDT
/&
* $$ E0J
```

## Starting the TCP/IP Support of VSE/POWER

When VSE/POWER is started and a TCP/IP node has been defined in the network definition table, VSE/POWER initializes its interface to the feature TCP/IP for VSE/ESA which runs in its own partition (usually F7). Since the partition is started after VSE/POWER initializes its interface, usually messages are displayed on the console reporting the delay of the initialization, for example:

```
F1 0001 1RB4I PLOAD NETWORK DEFINITION TABLE NODE1NDT LOADED
F1 0097 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL INITAPI ISSUED
F1 0097 1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL SOCKET ISSUED
F1 0097 1RT2I TCP/IP: EZASMI MACRO-REQUEST SOCKET FAILED, RC=-0000001,
                ERRNO=+0001117, TS=08
F1 0097 1RTJA TCP/IP: INITIALIZATION OF INTERFACE UNSUCCESSFUL, RETRY GOING
                ON
```

As soon as TCP/IP for VSE/ESA has completed its initialization and establishes its interface to VSE/POWER, the highlighted message 1RTJA is no longer highlighted and rolls off the console and following messages are issued by VSE/POWER:

```
1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL GETHOSTID ISSUED
1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL BIND ISSUED
1RT7I TCP/IP: INTERFACE STARTING, SOCKET CALL LISTEN ISSUED
```

The messages reporting the delayed initialization may be avoided by following definitions:

1. Omit the definition of a TCP/IP node in the network definition table. One PNODE macro call for the local node is sufficient to initialize the PNET during the startup of VSE/POWER, for example

```
NODE1ND0 PNODE NODE=NODE1,LOCAL=YES
                END
```

And specify the network definition table NODE1ND0 in the POWER macro (see above).

2. Activate the network definition table NODE1NDT with the TCP/IP node(s) by using the PLOAD command of VSE/POWER after TCP/IP for VSE/ESA has completed its initialization. To automate the activation of the NDT, you may add the following two lines in the ASI-procedure of a partition:

```
// EXEC IESWAIT,PARM='20'
// EXEC DTRIATTN,PARM='PLOAD PNET,NODE1NDT'
```

## Sample TCP/IP Connection Definitions

The PARM-value of 20 in the first statement causes the system to wait for 20 seconds before the next statement is executed. Usually TCP/IP for VSE/ESA completes its initialization in less than 20 seconds, if not, use an appropriate value.

VSE/ESA distributes a sample procedure (USERBG.PROC in VSE/AF library IJSYSRS.SYSLIB, respectively the skeleton SKUSERBG in VSE/ICCF library 59), where this technique is used to tune the initialization of VTAM and CICS. Therefore it might be the best to add the two lines above to this procedure.

## Stopping the TCP/IP Support of VSE/POWER

When it becomes necessary to stop the part of TCP/IP for VSE/ESA running in its own partition, the TCP/IP support of VSE/POWER should be stopped first by issuing the VSE/POWER command PSTOP TCPIP. However, VSE/POWER's TCP/IP support is terminated not before all TCP/IP nodes have been stopped (by issuing the command PSTOP PNET,nodename).

## Multihoming Support (More than One Ip-Address)

TCP/IP for VSE/ESA supports the multihoming feature, which means one system may be addressed by more than one ip-address. In addition to the local ip-address (which is defined by the SET IPADDR statement), each link may have defined its own ip-address. This may be useful, when data is passed through a firewall or some hardware boxes must have their unique ip-address. Usually, it is not necessary to make use of this multihoming support (just do not specify an ip-address in the DEFINE LINK statement).

The following samples illustrate the necessary definitions. The system NODE1 has defined multiple ip-addresses in its configuration file, the local ip-address 19.9.19.44 and the ip-address 19.9.19.100 for its link.

```
SET IPADDR=19.9.19.44
DEFINE LINK, ID=OSA700, TYPE=OSAX, DEV=(700,701), DATAPATH=702, MTU=1500, -
PORTNAME=F14ABC1, :hp1. IPADDR=19.9.19.100:ehp1.
DEFINE ROUTE, ID=POWERHUS, LINKID=OSA700, IPADDR=0.0.0.0, -
GATEWAY=19.9.19.1
START LINKID=OSA700
```

When NODE2 connects to NODE1 through the link with the ip-address 19.9.19.100, VSE/POWER must identify itself to NODE2 with the ip-address of the link (19.9.19.100) and not with the local ip-address (19.9.19.44). This is achieved by using the IPEXTRAD operand in the PNODE macro for NODE2. The ip-address of NODE2 (19.9.19.45) remains unchanged.

```
NODE1MHG PNODE NODE=NODE1, LOCAL=YES
PNODE NODE=NODE2, IPHOSTAD=19.9.19.45, IPEXTRAD=19.9.99.100
END
```

Since NODE2 connects to NODE1 through the ip-address of the link (19.9.19.100), the NDT at NODE2 must read:

```
NODE2MHG PNODE NODE=NODE2, LOCAL=YES
PNODE NODE=NODE1, IPHOSTAD=19.9.19.100
END
```

When the IPEXTRAD has been specified for a PNODE macro, it is displayed by the command PDISPLAY PNET,nodename, for example:

## Sample TCP/IP Connection Definitions

```
PDISPLAY PNET,NODE2
1RB7I ***** NDT NAME = NODE1MHG *****
1RB7I NODE      ROUTE1  ROUTE2  AUTH BSIZE APPLID/IPADDR  IPPORT SPORT
1RB7I NODE2    TCP LINK                JOB 4096 19.9.99.45    E 175
1RB7I          IPEXTRAD=19.9.19.100
```

The character 'E' in the first line below the heading IPPORT indicates that the IPEXTRAD operand has been specified. When the command PDISPLAY PNET,ALL is issued, the second line isn't displayed and the character 'E' is the only indication for the usage of the IPEXTRAD operand, for example:

```
PDISPLAY PNET,ALL
1RB7I ***** NDT NAME = NODE1MHG *****
1RB7I NODE      ROUTE1  ROUTE2  AUTH BSIZE APPLID/IPADDR  IPPORT SPORT
1RB7I NODE1    ----- LOCAL -----                175
1RB7I NODE2    TCP LINK                JOB 4096 19.9.99.45    E 175
:exmp.
```

## Sample TCP/IP Connection Definitions

---

## Glossary

This glossary includes terms and definitions related primarily to IBM VSE/POWER. If you do not find the term you are looking for, refer to the index of this book or view the *IBM Glossary of Computing Terms* at

<http://www.ibm.com/ibm/terminology>

The glossary includes definitions with:

- Symbol \*, which denote entries from the *IBM Glossary of Computing Terms*.
- Symbol (A) from the *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036.
- Symbols (I) or (T) from the *ISO Vocabulary - Information Processing* and the *ISO Vocabulary - Office Machines*, developed by the International Organization for Standardization, Technical Committee 97, Subcommittee 1. Definitions of published sections of the vocabularies are identified by symbol (I) after definition; definitions from draft international standards, draft proposals, and working papers in development by the ISO/TC97/SC1 vocabulary subcommittee are identified by symbol (T) after definition, indicating final agreement has not yet been reached among participating members.

The following cross-references are used:

- Contrast with. This refers to a term that has an opposed or substantively different meaning.
- Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the dictionary.
- Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.
- See. This refers the reader to multiple-word terms that have the same last word.
- See also. This refers the reader to related terms that have a related, but not synonymous, meaning.

**access register (AR).** A hardware register that a program uses to identify an address space or a data space. Each processor has 16 ARs, numbered 0 through

15, which are paired one-to-one with the 16 general-purpose registers (GPRs).

**account file.** A disk file maintained by VSE/POWER containing accounting information generated by VSE/POWER and the programs running under VSE/POWER.

**address.** 1. The location in the storage of a computer where data are stored. 2. In data communication, the unique code assigned to every device or work station connected to a network.

**addressing mode (AMODE).** A program attribute that refers to the address length that a program is prepared to handle on entry. Addresses may be either 24 bits or 31 bits in length. In 24-bit addressing mode, the processor treats all virtual addresses as 24-bit values; in 31-bit addressing mode, the processor treats all virtual addresses as 31-bit values. Programs with an addressing mode of ANY can receive control in either 24-bit or 31-bit addressing mode.

**address space.** A range of up to two gigabytes of contiguous virtual storage addresses that the system creates for a user. Unlike a data space, an address space contains user data and programs, as well as system data and programs, some of which are common to all address spaces. Instructions execute in an address space (not a data space). Contrast with data space.

**address space control (ASC) mode.** The mode (determined by the PSW) that tells the system where to find referenced data. It determines how the processor resolves address references for the executing programs. z/VSE supports two types of ASC modes:

1. In **primary ASC mode**, the data that a program can access resides in the program's own (primary) address space. In this mode, the system uses the contents of general-purpose registers to resolve an address in the address space; it does not use the contents of the access registers (ARs).
2. In **access register (AR) ASC mode**, the data that a program can access may reside in an address space other than the primary or in a data space. In this mode, the system uses both a general-purpose register (GPR) and the corresponding access register together to resolve an address in another address space or in a data space. Specifically, the AR contains a value, called an ALET, that identifies the address space or data space that contains the data, and the GPR contains a base address that points to the data within the address space or data space.

\* **adjacent nodes.** Two nodes connected without any intermediate nodes. (T)

\* **alternate route.** A secondary or backup route that is used if normal routing is not possible. Contrast with primary route.

**AR.** Access register.

**AR (access register) mode.** If a program runs in AR mode, the system uses the access register/general-purpose register pair to resolve an address in an address space or data space. Contrast with *primary mode*. See also *address space control (ASC) mode*.

**ASC mode.** Address space control mode.

**assemble.** To translate a program from assembler language into object code.

\* **autostart.** A facility that starts up VSE/POWER with little or no operator involvement.

**binary synchronous communication (BSC).** Method of telecommunication using binary synchronous line discipline. Contrast with SDLC.

**block.** Usually, a block consists of several records of a file that are transmitted as a unit. But if records are very large, a block can also be part of a record only.

**BSC.** Binary synchronous communication.

**BSD.** Used with tracing of the BSD/C (Berkeley C language) socket interface of TCP/IP for VSE/ESA.

**buffer.** An area of storage temporarily reserved for input or output operations; an area into which data is read or from which data is written.

**byte.** Eight adjacent binary digits that are operated upon as a unit and that constitute the smallest addressable unit of information within a computer system. Normally, it represents a stored character.

**catalog.** 1. A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (I) (A) 2. To store a library member such as a phase, module, or book in a sublibrary.

\* **cataloged procedure.** A set of control statements placed in a library and retrievable by name.

**channel-to-channel-attachment (CTCA).** A function that allows data to be exchanged (1) under the control of VSE/POWER between two virtual VSE machines running under VM or (2) under the control of VTAM between two processors.

**CICS.** Customer Information Control System.

\* **CICS/VSE.** An IBM program that lets transactions entered at remote terminals be processed concurrently by application programs. It includes facilities for building, using, and maintaining data bases.

**class.** In VSE/POWER, a means of grouping output or jobs that require the same set of resources.

**console.** A display and keyboard used for communication between the operator and the system.

**CTCA.** channel-to-channel attachment.

**data set header (DSHR).** The NJE control record that provides general information for SYSOUT data (list, punch) or record characteristics for SYSIN data (job).

**data space.** A range of up to two gigabytes of contiguous virtual storage addresses that a program can directly manipulate through ESA/370 instructions. Unlike an address space, a data space can hold only user data; it does not contain shared areas, system data or programs. Instructions do not execute in a data space, although a program can reside in a data space as non-executable code. Contrast with address space.

**default value.** A value assumed by the program when no value has been specified by the user.

**device.** A hardware component of a computer system with a specific purpose.

**device address.** 1. The identification of an input/output device by its channel and unit number. 2. In data communication, the identification of any device to which data can be sent or from which data can be received.

**disk.** Loosely, a magnetic disk unit.

**disposition.** A means of indicating to VSE/POWER how job input and output is to be handled. A job may, for example, be deleted or kept after processing.

**drain.** To deactivate a transmitter or a receiver task.

**dump.** 1. Data that has been dumped. (I) (A) 2. To write at a particular moment some contents of storage to another data medium for the purpose of safeguarding or debugging the data. (T)

**exit.** A routine, normally user-supplied, that receives control from the system when a certain event occurs (abnormal-end exit, for example).

**extended addressability.** 1. See *31-bit addressing*. 2. The ability of a program to use virtual storage that is outside the address space in which the program is running. Generally, instructions and data reside in a single address space - the primary address space. However, a program can have data in address spaces other than the primary or in data spaces. (The instructions remain in the primary address space, whilst the data can reside in another address space or in a data space.) To access data in other address spaces, a program must use access registers (ARs) and execute in access register mode (AR mode).

**EZASMI macro.** An assembler application programming interface for TCP/IP services that addresses the BSD/C interface of TCP/IP for VSE/ESA through the LE (Language Environment) C support.

**file.** A named set of records stored or processed as a unit. (T)

**general-purpose register.** A register, usually explicitly addressable, within a set of registers that can be used for different purposes.

**generate.** To produce a computer program by selecting subsets of standardized code under the control of parameters. (A)

\* **GETVIS space.** Storage space within a partition or the shared virtual area, available for dynamic allocation to programs.

**GPR.** General-purpose register.

**hard copy.** A copy of machine output printed on paper in a visually readable form.

**hard-copy file.** A system file on disk, used to log all lines of communication between the system and the operator at the system console, to be printed on request.

**hardware.** Physical equipment used in data processing, as opposed to programs, procedures, rules, and associated documentation. (I) (A) Contrast with software.

**initial program load (IPL).** The process of loading system programs and preparing the system to run jobs.

**input.** 1. Information or data to be processed by a program. 2. Pertaining to a resource that serves to process input data, for example, input buffer. Contrast with output.

**interactive partition.** An area of virtual storage dynamically allocated by CICS/VSE for the purpose of processing a job that was submitted interactively from a terminal.

**IPL.** Initial Program Load.

**JCL.** Job Control Language.

**JECL.** Job Entry Control Language.

**job.** In VSE/POWER, a job is identified either by a JOB statement followed by one EXEC statement for each of the programs or job steps (VSE job) or by one or several such jobs enclosed within an \* \$\$ JOB and an \* \$\$ EOJ statement (JECL).

**job control language (JCL).** A language that serves to prepare a job or each job step of a job to be run. Some of its functions are: To determine the I/O devices to be

used, set switches for program use, log (or print) its own statements, fetch the first phase of each job step.

**job entry control language (JECL).** A control language that allows the programmer to specify how VSE/POWER should handle a job.

**job header record (JHR).** The NJE control record that provides general information relating to the job as a whole.

**job stream.** The sequence of jobs as submitted to an operating system.

**job trailer record (JTR).** The NJE control record that terminates the job and generally provides accounting information.

**library.** Disk storage space in the system where programs in various forms and storage dumps are stored. The form of a program is indicated by its type ID, such as object module, phase, or procedure. Source code is identified by a one-character type ID of which some are reserved. A library consists of at least one sublibrary which can contain any type of member. Storage dumps from a library can be viewed interactively.

**link-edit.** To create a loadable computer program by having the linkage editor process compiled (assembled) source programs.

\* **load.** To bring a program phase from a library into virtual storage to run it.

\* **logging.** The recording of data about specific events.

**macro (instruction).** 1. In assembler programming, a user-invented assembler statement that causes the assembler to process a set of statements defined previously in the macro definition. 2. A sequence of VSE/ICCF commands defined to cause a sequence of certain actions to be performed in response to one request.

**macro definition.** A set of statements and instructions that defines the name of, format of, and conditions for generating a sequence of assembler statements and machine instructions from a single source statement.

**macro expansion.** See macro generation.

**macro generation.** An assembler operation by which a macro instruction gets replaced in the program by the statements of its definition. It takes place before assembly. Synonymous with macro expansion.

**message.** 1. In VSE, a communication sent from a program to the operator or user. It can appear on a console, a display terminal, or on a printout. 2. In telecommunication, a logical set of data being transmitted from one node to another.

**ML.** multi-leaving.

**multi-leaving (ML).** Fully synchronized two directional transmission of a variable number of data streams between terminals and/or computers using BSC facilities.

**NCP.** Network Control Program.

**network.** 1. An arrangement of nodes (data stations) and connecting branches. 2. The assembly of equipment through which connections are made between data stations.

**network job entry (NJE).** 1. A facility for transmitting jobs (JCL and in-stream data sets), SYSOUT data sets, (job-oriented) operator commands and operator messages, and job accounting information from one computing system to another. 2. A facility that provides access to batch computing facilities from other host systems. It enables users to transfer work and data throughout a distributed network of batch computing facilities. ("NJE" is not a part of "Systems Network Architecture (SNA)", but is an application layer which uses SNA, BSC, and CTC transmission facilities.) 3. The JES2 program product implementation of the NJE Protocol.

**network job interface (NJI).** A set of protocols for communication between various programs via telecommunication lines. The protocols describe what actions a system must take in defined situations while communicating with another node in the network.

**Network Control Program (NCP).** An IBM program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

**NJE.** network job entry.

**NJI.** network job interface.

**node.** 1. In SNA, an end point of a link or junction common to several links in a network. Nodes can be distributed to host processors, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities. 2. In VTAM, a point in a network defined by a symbolic name. Synonymous with network node.

\* **operator command.** A statement to a control program, issued via a console or terminal. It causes the control program to provide requested information, alter normal operations, initiate new operations, or end existing operations.

\* **operator console.** A display console used for communication between the operator and the system.

**output.** Data that has been processed and is transported from storage to an output device.

\* **partition.** A division of the virtual address area available for running programs.

\* **password.** In computer security, a string of characters known to the computer system and a user. He must specify it to gain full or limited access to the system and to the data stored in it.

**PNET.** Programming support available with VSE/POWER; it provides for the transmission of selected jobs, operator commands, messages, and program output between the nodes of a network.

**POWER.** See VSE/POWER.

**primary mode.** If a program runs in primary mode, the system resolves all addresses within the current (primary) address space. Contrast with *AR (access register) mode*. See also *address space control (ASC) mode*.

**primary route.** The preferred routing for the transmission of jobs or output. Contrast with alternate route.

**priority.** A rank assigned to a partition, a task, or a VSE/POWER queue entry, that determines its precedence in receiving system resources.

**procedure.** A set of control statements placed in a library and retrievable by name. Synonymous with cataloged procedure.

\* **processing.** The performance of logical operations and calculations on data, including the temporary retention of data in processor storage while this data is being operated upon.

**processor storage.** The storage contained in one or more processors and available for running machine instructions. Synonymous with real storage.

**punch.** 1. To make holes in some data medium according to a signal code and thus save data on that medium. 2. A machine (output device) to punch 80-column punch cards.

\* **queue.** 1. A line or list formed by items in a system waiting for service; for example, tasks to be performed or messages to be transmitted in a network. 2. To arrange in, or form, a queue.

**queue file.** A disk file maintained by VSE/POWER that holds control information for the spooling of job input and job output.

\* **RCB.** Record control byte

\* **read.** To acquire or interpret data from a storage device, from a data medium, or from another source. (I) (A)

**reader.** An input device that reads data that is punched into cards or is written or printed on paper.

**real storage.** The storage contained in one or more processors and available for running machine instructions. Synonym for processor storage.



**receiver task.** In VSE/POWER, a program task which receives spooled input or output.

\* **record.** A collection of related data or words, treated as a unit.

\* **Record control byte (RCB).** In the NJE facility, a control character used to identify each record type within a transmission block.

\* **reentrant.** The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by several tasks.

**remote.** Pertaining to a system, program, or device that is accessed through a telecommunication line. Contrast with local. Synonym for link-attached.

**Remote job entry (RJE).** Submission of jobs through an input unit that has access to a computer through a data link.

**RJE.** Remote job entry.

\* **routine.** Part of a program, or a sequence of instructions called by a program, that may have some general or frequent use. (I) (A)

\* **routing.** The assignment of the path by which a message will reach its destination.

**remote spooling communications subsystem (RSCS).** The program that transmits spool files between VM/SP users, remote stations, and remote and local batch systems via its telecommunication facilities.

**RSCS.** remote spooling communications subsystem.

\* **run.** 1. A performance of one or more jobs. 2. A performance of one or more programs. 3. To cause a program or job to be performed.

**SDLC.** Synchronous Data Link Control.

\* **Secure Sockets Layer (SSL).** A security protocol that provides communication privacy. SSL enables client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery. SSL was developed by Netscape Communications Corp. and RSA Data Security, Inc.

**segmentation.** In VSE/POWER, a facility that breaks list or punch output of a program into segments so that printing or punching can start before this program has finished generating such output.

**session.** The period of time during which a user of a terminal can communicate with an interactive system. A session can exist between any two nodes of a network and is established by VTAM statements.

**shared area.** An area of storage that is common to all address spaces in the system. z/VSE has two shared areas:

1. The shared area (24 bit) is allocated at the start of the address space and contains the supervisor, the SVA (for system programs and the system GETVIS area), and the shared partitions.
2. The shared area (31 bit) is allocated at the end of the address space and contains the SVA (31 bit) for system programs and the system GETVIS area.

\* **shared spooling.** A function that permits the VSE/POWER account file, data file, and queue file to be shared among several computer systems with VSE/POWER.

\* **shutdown.** The process of ending operation of a system or a subsystem, following a defined procedure.

**SNA.** Systems Network Architecture.

\* **software.** Programs, procedures, rules, and any associated documentation pertaining to the operation of a computer system.

\* **spool-access support.** A function of VSE/POWER that allows user programs or subsystems running on VSE to access the spool files of VSE/POWER.

\* **spool file.** 1. A file that contains output data saved for later processing. 2. One of three VSE/POWER files on disk: queue file, data file, and account file.

\* **spooling.** Reading and writing input and output streams on an intermediate device in a format convenient for later processing or output.

\* **SRCB.** Subrecord control byte

\* **SSL.** See Secure Sockets Layer.

**SSP.** System Support Program.

**startup.** The process of performing IPL of the operating system and of getting all subsystems and application programs ready for operation.

**station.** 1. One of the input or output points of a network that uses communication facilities; for example, the telephone set in the telephone system or the point where the business machine interfaces with the channel on a leased private line. 2. One or more computers, terminals, or devices at a particular location.

**storage.** A device, or part of a device, that can retain data. See also auxiliary storage, processor storage, virtual storage.

**submit.** A function that passes a job to the system for processing.

\* **subrecord control byte (SRCB).** In NJE, a control character used to provide additional information about a system.

\* **subsystem.** A secondary or subordinate system or program, usually capable of operating independently of, or asynchronously with, the operating system.

**synchronous data link control (SDLC).** A discipline, conforming to international standards, for managing synchronous information transfer over a communication link.

\* **system console.** A console, usually equipped with a keyboard and display screen for control and communication with the system.

**Systems Network Architecture (SNA).** The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

**System Support Program (SSP).** An IBM program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

\* **task.** The basic unit of synchronous program execution. A task competes with other tasks for system resources such as processing time and I/O channels.

**TCP/IP for VSE/ESA.** A product distributed by IBM that provides the TCP/IP interface within a z/VSE system.

**terminal.** A point in a system or network at which data can either enter or leave. (A) Usually a display screen with a keyboard. Also used for RJE workstation.

**trace.** 1. To record a series of events as they occur. 2. A record of specified events during the run of a program. 3. A program to produce such a record.

**transmit.** To send data from one place for reception elsewhere via an NJE network.

**transmitter task.** In VSE/POWER, a program task which transmits spooled input or output.

**user exit.** A programming service provided by an IBM software product that may be requested during the execution of an application program for the purpose of transferring control back to the application program upon the later occurrence of a user-specified event.

**virtual disk.** A range of up to two gigabytes of contiguous virtual storage addresses that a program can use as workspace. Although the virtual disk exists in storage, it appears as a real FBA disk device to the user program. All I/O operations directed to a virtual disk are intercepted and the data to be written to, or read from, the disk is moved to or from a data space.

Like a data space, a virtual disk can hold only user data; it does not contain shared areas, system data or programs. Unlike an address space or a data space,

data is not directly addressable on a virtual disk. To manipulate data on a virtual disk, the program has to perform I/O operations.

\* **virtual I/O area (VIO).** An extension of the page data set; used by the system as intermediate storage, primarily for control data.

**VSE (Virtual Storage Extended).** A system that consists of a basic operating system and any IBM-supplied and user-written programs required to meet the data processing needs of a user. VSE and the hardware it controls form a complete computing system. Its current version is called z/VSE.

**VSE/POWER.** A program primarily used to spool input and output. The program's networking functions enable a z/VSE system to exchange files with or run jobs on another remote processor. It is part of VSE Central Functions, a base program of z/VSE. The acronym POWER stands for 'Priority Output Writers, Execution Processors, and Input Readers'.

**z/VSE.** The current designation of the VSE operating system, formerly referred to as VSE/ESA.

**31-bit addressing.** Provides addressability for address spaces of up to 2 gigabytes. (The maximum amount of addressable storage in previous systems was 16 megabytes.)

---

# Index

## A

A-BOOK 85, 95  
abbreviations xiii  
accounting 28  
activating transmitters and receivers 24  
ADD command 10  
alternate route 9, 25  
authorization, commands 38  
automatic node restart 43

## B

B-BOOKs 11  
bibliography xii  
blank truncation 31  
BSC connection 15  
BSC line trace 107  
BSC line, password 19  
BSC lines 6  
BSC/CTCA, PNET trace record 118  
buffer and I/O trace 117  
buffer size 52  
buffer, transmission 9  
buffers for transmission 52

## C

capacity 1  
changes, summary of xv  
command authorization table 39, 41  
commands priority 2  
commands, authorization 38  
concurrent transmitters 24  
connecting CTCAs 10  
connecting the network 5  
connection, BSC 15  
connections to RSCS 59  
conventions for input exit routine 30  
COUPLE command 10  
CTCA connection 16  
CTCA line trace 107  
CTCA line, password 19  
CTCA PSTART command 16  
CTCA requirements 10  
CTCA support 7  
CTCA support overview 6  
CTCA/BSC, PNET trace record 118

## D

data flow through network 2  
data security 42  
data set header record (DSHR) 29, 33, 36, 37  
debugging, exit 32, 37  
DEFINE command 10  
defining a CTCA 10  
destination reached 3  
diagnostic aids 63

diagnostic message, priority 2  
disabling exit routine 29  
disposition 21, 59, 61, 66  
distribution code (DIST), passing to VM/CP 59  
draining transmitter/receiver 25  
dump 32, 37

## E

enabling exit routine 29  
encryption type  
    NORMAL 18  
    STRONG 18  
    WEAK 18  
entries for future nodes 12  
event for notify 27  
excluded transmitter combinations 24  
exit routine  
    transmitter 33  
exit routine failure 32, 37  
exit routine, disable/enable 29  
exit routines, intercommunication 32  
exit work area size 29  
exit, PNET 29

## F

failure of exit routine 32, 37  
failure, exit 32  
FIFO order within priority 1  
flow of data 2

## I

I/O and buffer trace 117  
IBP\$IDM 37  
Idump 37  
information xii  
initialization of network with JES2 59  
initialization of PNET 11  
initialization of work area 29  
initialization with autostart 15  
intercommunication, exit routines 32  
IPW\$IDM 32

## J

JECL considerations 54  
JES2 node, receiving output from 60  
JES3 NJE 60  
job characteristics, change of 54  
job header record (JHR) 28, 29, 31, 32, 33, 36, 37, 55, 57, 58  
JOB JECL statement 54  
job number assignment 24  
job trailer record (JTR) 29, 31, 36, 37, 57

## L

LDEST operand 21  
line password 16, 19  
line trace 107  
load levelling 25  
loading PNET 11  
loading the NDT 12  
lost path 2

## M

modetab example 83  
multi-leaving 15  
multitrunki 60

## N

NCP 50  
NDT 5  
NDT generated 9  
NDT in GETVIS space 11  
NDT loading 12  
NDT wrong or missing 64  
NDT, specifying the 8  
NETEXIT 29  
NETEXIT type routine 85  
network considerations 44  
network security 42  
network, connecting 5  
networking overview 2  
networking with shared spooling 54  
node name, changing 45  
node password 16, 18  
node password omitted 65  
node restart 43  
notify support examples 27  
notify support for VSE/ICCF 27  
number of transmitters and receivers 53

## O

originators, types of 1  
output files 1  
output files queued 3  
output flagged 21  
output lost 25  
output secured 42  
output, receiving from JES2 node 60  
own node name, changing 45

## P

pacing 11, 83  
PACT command, example 24  
parameter list, transmitter exit 34  
password 16, 43  
password not specified in JES3 connection 61  
password omitted 65

- password, line 19
- password, node 18
- passwords 9
- path lost 2
- path selection 25
- PDISPLAY TRINFO 118
- PDRAIN command 25
- PEND command 12
- performance considerations 51
- performance influenced 9
- PLINE macro 7
- PNET BSC/CTCA trace record 118
- PNET exit routine sample 85
- PNET I/O and buffer trace 117
- PNET initialization 11
- PNET instead of shared spooling 1
- PNET receiver exit, blank truncation 31
- PNET SNA trace record 119
- PNET SSL system messages 63
- PNET SSL trace record 120
- PNET support considerations 6
- PNET TCP system messages 63
- PNET TCP trace record 120
- PNET trace 107
- PNET, console trace 63
- PNET, line trace 63
- PNET, status 23
- PNODE generations 9
- POWER macro requirement 6
- primary route 9
- prime route 25
- priority 23
- problem isolation 64
- processors, number of 1
- protection, of data 42
- PSTART command 16

## R

- receiver exit example, PNET 85
- receiver exit, failure 32
- receiver, draining 25
- receivers, activating 24
- receivers, number of 53
- record of PNET BSC/CTCA trace 118
- required PLINE macro 7
- requirements, for VTAM 49
- resources freed 1
- restart of node, automatic 43
- return codes
  - PNET receiver exit 30
  - PNET transmitter exit 36
- RJE with PNET 44
- routing 8
- RSCS networking, CTCA support 6
- RSCS, connections to 59
- RSCS, passing distribution code (DIST) to VM/CP 59
- RSCS, restricted transmitters 66

## S

- sample for exit routine 85
- sample of a transmitter exit 95
- security, of data 42
- security, of output 42

- security, to lines and nodes 18
- selecting a path 25
- session starting 15
- SET ODNDST operand 16
- SET RSCSROOM operand 59
- shared spooling and networking 54
- shutdown of network 26
- SIGNON records 15
- size of buffers 52
- SNA link, password 19
- SNA links 7
- SNA session, connection interval 16
- SNA, PNET trace record 119
- specifying network layout 8
- spool-access support 1, 5
- spooling to transit nodes 1
- SSL line trace 107
- SSL requirements 11
- SSL, PNET trace record 120
- summary of changes xv

## T

- TCP line trace 107
- TCP, PNET trace record 120
- TCP/IP connections 7
- TCP/IP connections, starting subtasks for 13
- TCP/IP considerations 50
- TCP/IP interface xvi
- TCP/IP requirements 11
- telecommunication trace 117
- TELETR 118
- time interval, SNA session 16
- timeout 8
- timeout on PSTART 15
- trace record, PNET BSC/CTCA 118
- trace record, PNET SNA 119
- trace record, PNET SSL 120
- trace record, PNET TCP 120
- trace, exit failure 32
- trace, PNET 63
- trace, PNET BSC/CTCA/TCP 107
- trace, telecommunication 117
- traces 63
- transit nodes, spooling to 1
- transmission buffers 52
- transmission types 1
- transmit queue 21, 23
- transmitter exit 33
- transmitter exit example, PNET 95
- transmitter exit, failure 37
- transmitter throughput 52
- transmitter, draining 25
- transmitters, activating 24
- transmitters, number of 53
- TRINFO 118
- types of originators 1
- types of transmission 1

## U

- Use of PNET 1

## V

- VM/CP, sending distribution code (DIST) to 59
- VSE/ICCF notify support 27
- VTAM commands, results 50
- VTAM connection information 5
- VTAM examples 81
- VTAM pacing 52, 83
- VTAM requirements 11
- VTAM, considerations 49
- VTAM, node name to 8
- VTAM, opening interface to 16

## W

- work area
  - PNET receiver exit 29
  - PNET transmitter exit 33
- work area initialized 29

## X

- XDEST 21
- XMTEXT 33

---

## Readers' Comments — We'd Like to Hear from You

IBM z/VSE  
VSE Central Functions  
VSE/POWER Networking  
Version 7 Release 1

Publication No. SC33-8249-01

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Complete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to find	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Well organized	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Applicable to your tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

\_\_\_\_\_  
Name

\_\_\_\_\_  
Address

\_\_\_\_\_  
Company or Organization

\_\_\_\_\_  
Phone No.



Fold and Tape

Please do not staple

Fold and Tape



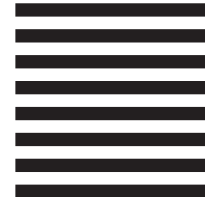
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

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



Fold and Tape

Please do not staple

Fold and Tape





File Number: S370/S390-34  
Program Number: 5686-CF7

Printed in USA

SC33-8249-01





Spine information:



IBM z/VSE  
VSE Central Functions

VSE/POWER Networking

Version 7 Release 1

SC33-8249-01