z/OS

Network Job Entry (NJE) Formats and Protocols

Version 2 Release 2

Contents

Figures

Tables

About This Document

Network Job Entry Formats and Protocols explains the Network Job Entry (NJE) formats and protocols used by program products that support networking.

For product-related material, consult the product libraries.

Organization of this Manual

Network Job Entry Formats and Protocols is divided into the following chapters:

Introduction to Network Job Entry (NJE) introduces the concepts needed to understand the material in the remaining chapters. This chapter defines a network and describes the 5 layers of networking.

Formats of NJE Transfer Units describes the format of the control blocks used to transmit data across the network.

Job Header discusses the content and format of the job header field.

Data Set Header discusses the content and format of the data set header field.

Job Trailer discusses the content and format of the job trailer field.

NJE Nodal Message Records (NMR) discusses the content and format of the NJE nodal message records.

Networking Connection Control Records (NCCR) discusses the content and format of the networking connection control records.

Presentation Services discusses how the networking protocols span, compress, and compact data to be transmitted through transmission buffers.

Stream Control discusses the multileaving functions of SNA, BSC, and CTC.

NJE Connection Control discusses establishing or terminating a connection between two nodes.

Link Protocols discusses SNA, BSC, and CTC protocol sequences for initialization, normal processing, and termination between nodes. This chapter also discusses error-recovery processing for each of the protocols.

Appendix A, "System-Dependent Considerations," on page 219 describes system-dependent considerations and deviations from the protocols described in this book.

Program Products

The following lists the program products that use the protocols as they are documented in this manual.

- expand.
- MVS/SP JES2

- – Version 2.2.0
- – Version 3.1.1
- – Version 3.1.3
- – Version 4.1.0
- – Version 4.2.0
- – Version 4.3.0
- – Version 5
- MVS/SP JES3
  - – Version 2.2.3
  - – Version 3.1.2
  - – Version 3.1.3
  - – Version 4.2.1
  - – Version 4.2.2
  - – Version 4.3.0
  - – Version 5
- RSCS
  - – Version 2.2
  - – Version 2.3
  - – Version 3.1
  - – Version 3.1.1
  - – Version 3.2.0
- VSE/POWER
  - – Version 6.6 with VSE/ESA 2.6
  - – Version 6.7 with VSE/ESA 2.7
  - – Version 7 with z/VSE 3
- MVS/BDT Version 2
- AS/400 Communication Utilities
  - – Version 2.1.0
  - – Version 3.1
  - – Version 3.2
  - – Version 3.6
  - – Version 3.7

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS V2R2 Information Roadmap*.

To find the complete z/OS® library, go to IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or provide any other feedback that you have.

Use one of the following methods to send your comments:
1. Send an email to mhvrcfs@us.ibm.com.
2. Send an email from the "Contact us" web page for z/OS (http://www.ibm.com/systems/z/os/zos/webqs.html).

Include the following information:
- Your name and address.
- Your email address.
- Your telephone or fax number.
- The publication title and order number:
      V2R2 Network Job Entry (NJE) Formats and Protocols
      SA32-0988-02
- The topic and page number that is related to your comment.
- The text of your comment.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute the comments in any way appropriate without incurring any obligation to you.

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

Do not use the feedback methods that are listed for sending comments. Instead, take one of the following actions:
- Contact your IBM service representative.
- Call IBM technical support.
- Visit the IBM Support Portal at z/OS Support Portal (http://www-947.ibm.com/systems/support/z/zos/).

Summary of changes for z/OS Version 2 Release 2 (V2R2) as updated December, 2015

The following changes are made for z/OS V2R2 as updated December, 2015.

Changed

- "Data set header general section" on page 48 is updated to add flag byte NDHGFLG2.

z/OS Version 2 Release 2 summary of changes for Network Job Entry (NJE) Formats and Protocols

The following changes are made to z/OS Version 2 Release 2 (V2R2).

New

- Table 9 on page 42 in "JES2 Section of the Job Header" on page 42 of Chapter 3, "Job Header," on page 19 has been updated with new fields NJH2DFJG and NJH2FLG2.
- With APAR OA41740, fields NDHGF2HX, NDHGF2TR, and NDHGF2NO have been added to Table 14 on page 48 in "Data set header general section" on page 48 of Chapter 4, "Data set header," on page 47.
- Field NCCINOS has been added to Table 39 on page 137 in "Initial Signon Record" on page 137.

Changed

- Table 9 on page 42 in "JES2 Section of the Job Header" on page 42 of Chapter 3, "Job Header," on page 19 Bit NDHGF2HB, Bit NDHGF2HA, and NJH2FLG1 have been updated.
- "Initialization" on page 211 is updated to add a negative acknowledgement (NAK) of 4.

z/OS Version 2 Release 1 summary of changes

See the following publications for all enhancements to z/OS Version 2 Release 1 (V2R1):

- *z/OS V2R2 Migration*
- *z/OS Planning for Installation*
- *z/OS Summary of Message and Interface Changes*
- *z/OS V2R2 Introduction and Release Guide*

Chapter 1. Introduction to Network Job Entry (NJE)

An NJE network is a group of two or more complexes or systems that communicate with each other. An NJE network is comprised of nodes that can transmit or receive a unit of work. The nodes in an NJE network use protocols to communicate with each other. Protocols are rules a node uses to:

- Become part of an NJE network
- Receive a unit of work
- Send a unit of work
- Indicate it was removed from the network.

For an illustration of an NJE network, see Figure 1.
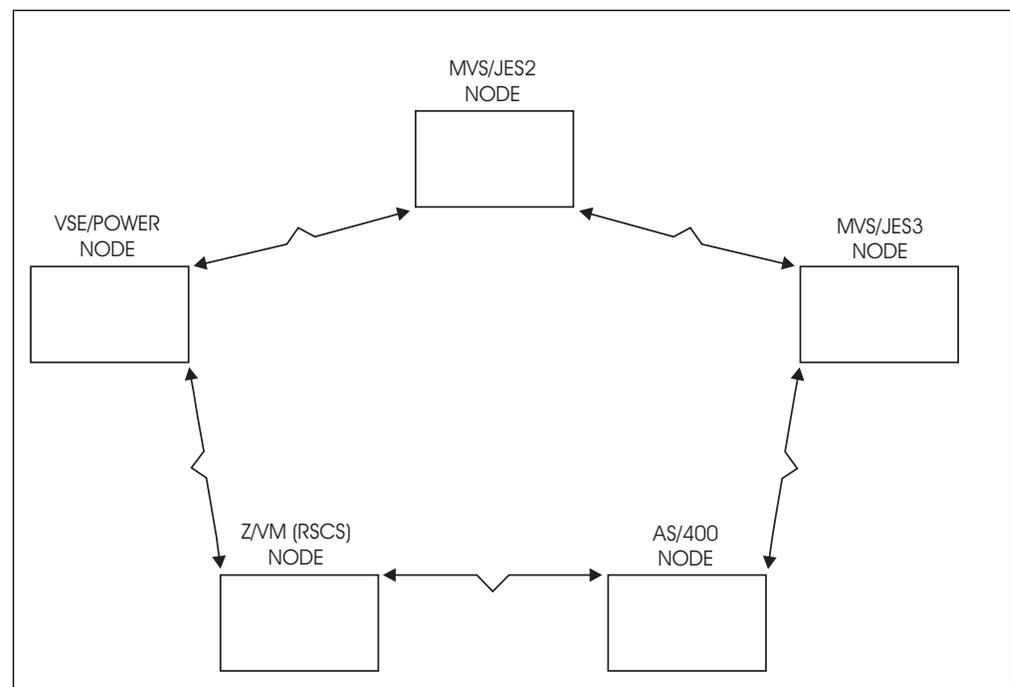
Figure 1. An NJE Network

To become part of an NJE network, an installation must install an NJE network facility that uses the protocols documented in this manual. Any of the following network facilities can be installed:

- JES2
- JES3
- RSCS
- VSE/POWER
- AS/400 Communication Utilities (AS/400 [1] does not support VSE/POWER communications in an NJE network.)

---

1. AS/400 is a trademark of the IBM Corporation.

Units of Work

An NJE transfer unit is a unit of work that is transmitted across the network. An NJE transfer unit can be either an NJE job or a nodal message record (NMR).

An **NJE job** is a transfer unit that contains data to be processed at another node in the NJE network. It begins with a job header, is followed by data, and ends with a job trailer. The type of data contained in the NJE job further defines the type of NJE job. The data between the job header and job trailer can be either SYSIN or SYSOUT data. An NJE SYSIN job is an NJE job that contains JCL for a job and may have one or more SYSIN data sets. An NJE SYSOUT job is an NJE job that contains one or more SYSOUT data sets. Each SYSOUT data set is preceded by a data set header.

A **nodal message record (NMR)** is a unit of work that begins with an NMR header and is followed by message text. The message text can be either a message or a system command.

Functions of a Node

A node is a system or complex that is defined to an installation. A node in the network can be another complex or system within a single location or it can be a complex that resides in a remote location. Each node that a complex can access must be identified to other complexes by a unique NJE node name.

Note: If a node uses SNA protocols, the node has two names:
- An LU name (as defined to VTAM® [2]), and
- An NJE node name created during initialization processing

The NJE node name appears in job headers, data set headers, and NMRs. Do not confuse the LU name with the node name, they are two separate entities.

Each node in the network can do the following with an NJE transfer unit:

`Transmit`
> The node packages the NJE transfer unit and transmits it to another node.

`Receive`
> The node recognizes the NJE transfer unit, receives, and stores it.

`Store-and-forward`
> The node accepts the NJE transfer unit, stores it, and schedules it to be forwarded to another node.

Types of Nodes

NJE uses the following terminology for the nodes that comprise an NJE network.
- Originating Node is the node where the user submitted the request to *transmit* the data to another complex.
- Intermediate Node is a node that lies in the path of either the:
  - Originating node and execution node
  - Execution node and the destination node

---

2. VTAM is a trademark of the IBM Corporation

It *receives* and *transmits* the NJE transfer unit to the next node in the path of the target node.

- Target Node is the node where a NJE job or NMR is *received* and will either be *executed* or be *processed.* The target node can be either a:

Destination Node

  is a node that receives and processes:

  - An NJE SYSOUT job. A node processes an NJE SYSOUT job by printing or punching the SYSOUT data set.
  - A message contained in an NMR.

  When an NJE transfer unit reaches its destination, it may or may not be processed as the user intended, depending on the facilities available at that node. NJE protocols allow the destination node to reject files that it cannot process or perform other system-dependent actions.

Execution Node

  is the node where:

  - JCL contained in an NJE SYSIN job executes. The node packages the SYSOUT data sets created by the SYSIN in an NJE SYSOUT job and sends the NJE job to the destination node.
  - A command contained in an NMR is processed. The node packages the messages that are a result of the command in an NMR and sends the NMR to the destination node.

The execution node may not necessarily be the destination node. If, for example, a user submits a job specifying that the job execute at one complex and job's output be printed at a different complex then the complex where the job runs is the execution node and the complex where the SYSOUT prints is the destination node. If no SYSOUT destination is specified, then the origin node and destination node are the same by default.

Figure 2 illustrates the different types of nodes in an NJE network. The network is composed of 3 nodes. If a user submits a job at node 1 to be executed at node 3:

- Node 1 is the originating node because that is where the user submitted the request.
- Node 2 is the intermediate node, because it is in the path of the destination node. Node 2 receives the data, stores it, then forwards it to the next node in the path of the target node.
- Node 3 is both the execution and destination node because it is the node that the user specified as the target.
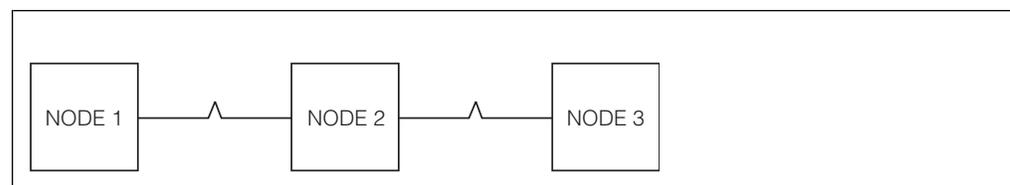
Figure 2. Sample NJE Network

To transmit an NJE transfer unit to a complex other than the user's installation (a remote node), the user issues a command or submits a job specifying a destination node name. The destination node can be either directly- or indirectly-connected to

the originating node. In the network depicted in Figure 2 on page 3, if NODE1 is the originating node, NODE2 is a directly-connected node to NODE1, and NODE3 is an indirectly-connected node to NODE1.

Types of NJE Users

Originating User is the user that submits an NJE transfer unit at the originating node. The originating user submits the NJE transfer unit at an operator console, terminal, or an RJE workstation. An NJE transfer unit may originate from another NJE transfer unit.

Destination User is a user or device (printer or punch) that is the target of an NJE SYSOUT job.

Notification User is the user who receives messages that notify the user of the status of the NJE transfer unit.

Accounting User is the user that receives the notification of the amount or cost of system resources used in processing an NJE transfer unit.

NJE Addressing

NJE protocols route an NJE job to the specified destination. By obtaining the destination from the NJE job's JCL. If the originating user identifies an RJE workstation or a userid that should receive the output or notification messages, the additional information is appended to the end of the destination node. Figure 3 identifies the different methods of addressing nodes.

```
Destination  =  Node

Destination  =  (Node.  Remote  id)

Destination  =  (Node.  User  id)
```

Figure 3. NJE Addressing

The destination node routes the job to the remote or user at the destination node.

Note: JES2 and JES3 have implemented a convention for routing files to TCP/IP connection workstation by specifying the destination in the form

Dest = node.IP:ip_addr

where ip_addr can be an IP address or domain name.

NJE Protocols

Nodes in an NJE network are connected by lines such as SNA sessions, channel-to-channel adapters, leased or dial-up telephone lines, microwave links, or satellite. An **NJE connnection** is the use of NJE protocols between two nodes. NJE protocols are rules that direct the logical structure, format and operational sequence for transmitting NJE transfer units between two nodes in an NJE network. The three NJE protocols are:

• Binary synchronous communications (BSC)

- LU type 0 Systems network architecture (SNA)
- Channel-to-channel (CTC).

Channel-to-channel protocols are similar to BSC protocols and are treated the same in this manual.

NJE protocols are divided into five layers, as illustrated in Figure 4. The remaining information in this chapter gives an overview of these layers (and Chapters 2 through 6 the following topics provide more detailed information). Appendix A, "System-Dependent Considerations" on page A-1 contains product-specific information about these layers.
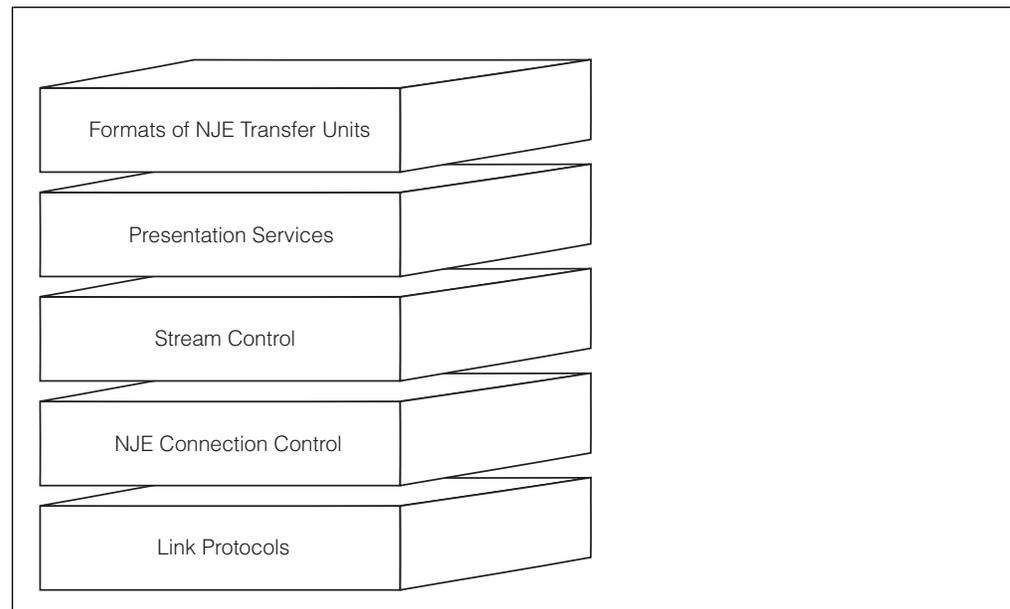
Figure 4. NJE Protocol Layers

Formats of NJE Transfer Units

This section discusses the data format for NJE jobs and nodal message records (NMRs). For detailed information about the control blocks contained in an NJE transfer unit, see Formats of NJE Transfer Units.

Format of an NJE Job

An NJE job contains either SYSIN or SYSOUT data. The data must be packaged in a format that is recognizable by the sending and receiving nodes. An NJE job must contain a job header, data, and a job trailer.

An NJE job containing SYSIN data begins with a job header record followed by the job and SYSIN data, and ends with the job trailer record. Figure 5 shows the format of an NJE SYSIN job.

Job Header	Job/SYSIN data	Job trailer

Figure 5. Format of an NJE SYSIN Job

An NJE job containing SYSOUT data begins with a job header record followed by one or more data set header records and SYSOUT data sets, and ends with a job trailer record. Figure 6 shows the format of an NJE SYSOUT job.

Job Header	Data set header	SYSOUT data	Data set header	Data set header	SYSOUT data	Job trailer

Figure 6. Format of an NJE SYSOUT Job

The following describes the header, data, and trailer fields that comprise an NJE SYSIN or SYSOUT job.

Job Header

The job header record is a variable length record that contains one or more variable length sections. The job header record contains the:

- Job identifier
- Network jobid
- Execution address
- Notification address
- Origination address
- Security information
- Accounting information
- Execution defaults
- Estimated resources required.

Data

The networking protocols assume the work specification record length (SYSIN record length) is 80 bytes fixed. Trailing blanks may be truncated before transmission. All data is transmitted in transparent mode (unprintable characters are not translated).

- **Job/SYSIN data** is a single job or work specification. The syntax for work specification is that of the node where the job will execute. NJE protocols do not standardize work specification syntax. The work specification can contain more than one SYSIN data set, each with a different record length. If the record length varies, a record change characteristics section (RCCS) containing the record length of the data (LRECL) and the record format (U, F, or V) will precede each SYSIN data set.

  Note: For JES3 nodes, the work specification will be MVS™ JCL or JES3 JECL. JES3 does not recognize the RCCS, but uses the record length information at the beginning of each record to determine the record's length.

- The **Data Set Header** and **SYSOUT Data** are created either when a:
  - Job is sent to a node to execute and the job creates SYSOUT data sets that must be sent to another node
  - SYSOUT data set is sent to another node

  An NJE SYSOUT job can contain one or more unique SYSOUT data sets preceded by one or more data set headers. The data set header contains the following information:
  - Data set name (unique only within the creating job)

- – Destination address
- – Source data record format and logical record length
- – Security information
- – Destination processing attributes such as SYSOUT class, FCB, and forms identifier.

Job Trailer

The job trailer record is a variable length record that contains one or more variable length sections. The job trailer is designed to contain execution time statistics and, although present at the end of the SYSIN data, contains no usable information until after job execution. For SYSOUT data, it may contain execution statistics for the job that created the SYSOUT data set.

Format of NMRs

Commands and messages are transmitted as individual nodal message records (NMRs). NMRs that contain a command or message cannot exceed 256 bytes. The NMR header fields specify the originating address, destination address, and the identifier of the console where the command was issued.

All NMRs contain a header followed by a message or command (formatted or unformatted) as shown in Figure 7.

Header	Message

or

Header	system-dependent unformatted command

or

Header	system-independent formatted command	command parameters
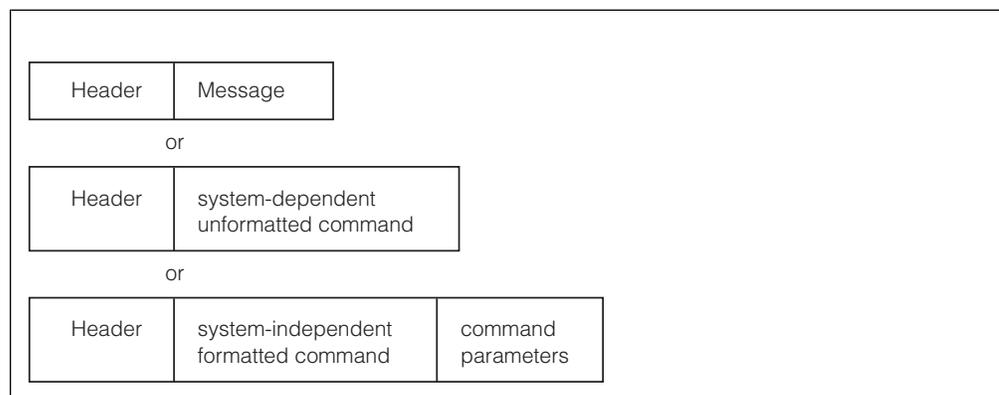
Figure 7. Format of an NMR

Command NMR

When a command reaches the execution node, the execution node retains the header portion of the NMR. If there is a response to the command, the execution node uses the header information to create a header for a response NMR.

If an NMR has text that is **unformatted**, then the NMR contains commands that have system-dependent syntax. If an NMR has **formatted** text, then the NMR contains one of the following commands, which have system-independent syntax:

- DISPLAY
- CANCEL
- HOLD
- RELEASE
- REROUTE

Message NMR

For messages, the NMR header may also contain an importance level and an output priority for unsolicited status messages that are not command responses.

The message text can begin with a time stamp and/or originating userid. An indicator in the NMR header defines whether any of these fields are present at the start of the text. The originating userid in the message text is used at the destination node to issue the messages to the user that submitted the job. If the message is in response to a command, then the message is transmitted to the node where the command was issued.

Presentation Services

Presentation services is the second layer of NJE protocols. It describes how an NJE transfer unit is packaged into one or more transmission buffers so that it can be transmitted across the network. Before transmitting an NJE transfer unit, the networking facility must:

- Obtain the logical record from spool, if necessary
- Segment the logical record into transmission buffers, if needed
- Compress the transmission buffers within transmission blocks
- Prepare NJE transmission buffers.

Preparing Transmission Buffers

Each transmission buffer contains routing information used by the protocol and part of the NJE transfer unit. Usually, more than one NJE transmission buffer is needed to transmit an NJE job, while an NMR can be contained within a single buffer.

The format of the transmission buffer depends on the type of protocols the node uses. A transmission buffer always contains control information used to transmit the buffer and transmission blocks.

Spanning Data

NJE facilities partition or span data into lengths it can easily manage for purposes of writing data to spool or transmitting data across the network. However, for the following reasons, the data in an NMR or a SYSIN data set is not spanned:

- The length of an NMR never exceeds the length of a transmission block.
- NMRs are not usually written to spool.
- The length of a logical record for SYSIN data never exceeds the length of a transmission block.

In order to transmit data across the network, the networking protocols documented in this manual require data to be divided into record segments that are no longer than 256 bytes. When an NJE network facility is spanning data, trailing blanks may be truncated. Each record segment contains a control segment and no more than 256 bytes of data. A control segment contains:

- The function control sequence (FCS) stream identifier (two function control sequence bytes indicating whether the data is SYSIN or SYSOUT).
- The type of record (job header, data set header, data, or job trailer).
- The length of the data.

Compressing Data

While preparing data for transmission, the networking facility compresses it removing blanks and duplicate characters. Compressing data reduces the amount of time required to transmit a transmission buffer.

When the networking facility encounters a string of duplicate letters or blanks, it places a string control byte (SCB) in the record to indicate the number of blanks or repeating non-blank characters that were deleted from the transmission buffer. The receiving node uses the information in the SCB to restore the record. An SCB can indicate up to 31 repeating characters and up to 63 non-repeating characters.

Format of a Transmission Buffer for BSC and CTC Protocols

Figure 8 illustrates the format of a buffer to be transmitted using BSC and CTC protocols.
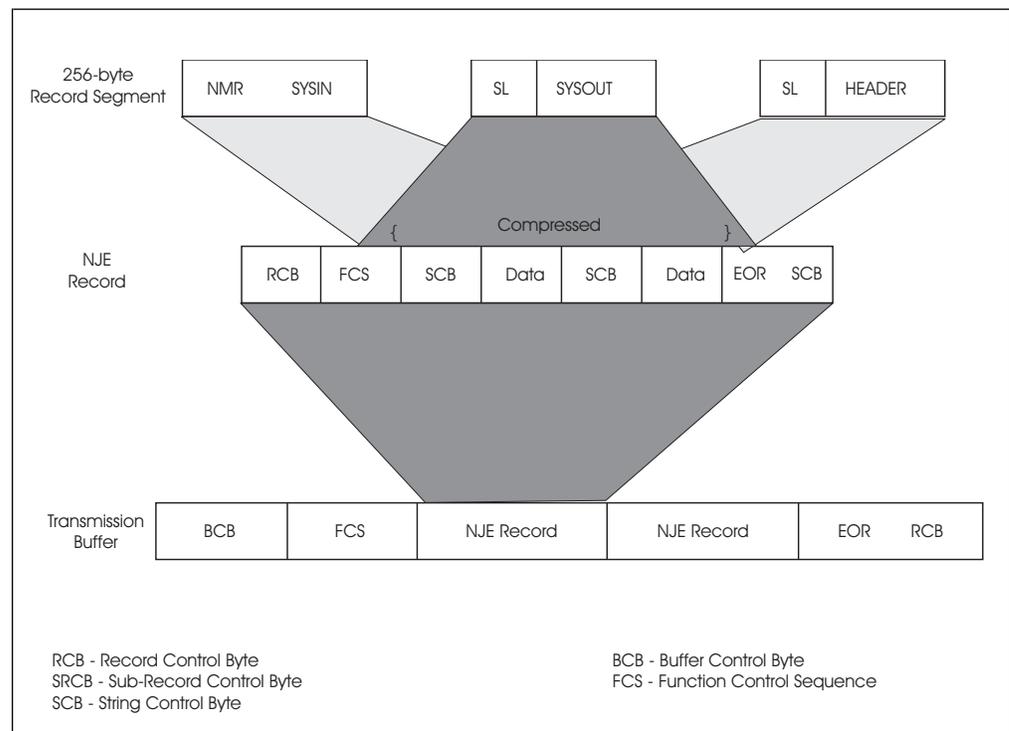
Figure 8. Format of a Transmission Buffer for BSC and CTC Protocols

Control and End-of-Buffer Information: Control information is placed at the beginning of each buffer that will be transmitted using BSC or CTC protocols. Every BSC transmission buffer begins with a block control byte (BCB) containing the outbound buffer sequence number (module 16). The sending node and the receiving node maintain an outbound and inbound buffer sequence counter.

Following the BCB are two bytes used by each receiver to control the inbound flow. These function control sequence (FCS) bytes permit nodes using BSC protocols to allow 8 SYSIN and SYSOUT streams to flow concurrently in each direction. Transmission blocks are placed after the FCS bytes. The format of a BSC transmission buffer does not prevent NJE from placing transmission blocks from different NJE transmission units in the same transmission buffer, but the common protocol precludes it.

Chapter 1. Introduction to Network Job Entry (NJE)   **9**

When a compressed transmission block cannot fit into the buffer, a special stand-alone record control byte (RCB) of X'00' is placed after the last transmission block and the buffer is truncated.

NJE Records: Each record segment is compressed into an NJE record. BSC compression results in string control bytes (SCBs) interspersed in the NJE record with duplicate character strings compressed. The SCB defines how many blanks or repeating non-blank characters are to be inserted by the receiver (up to 31 characters) or the length of strings without any duplicate characters (up to 63 characters). The resulting format will always begin with an initial SCB and may have additional SCBs interspersed.

The BSC protocol places a record control byte (RCB) and a subrecord control byte (SRCB) at the beginning of an NJE record. The RCB contains the FCS stream identifier and number; the SRCB defines the type of NJE record contained in the transmission buffer.

A stand-alone SCB with a value of X'00' (sometimes referred to as an "EOR SCB") is always placed at the end of each NJE record. In the BSC protocol, an end-of-file is represented by an RCB with the FCS stream identifier, and an SRCB of X'00'.

Format of a Transmission Buffer for the SNA Protocol

The SNA protocol compresses, and optionally compacts, the information in the NJE record before placing it in a transmission buffer. Therefore, NJE compresses both the information in the transmission buffer and the control information.

After the data in the NJE record is compressed, including the record identifier (RID), it is placed in the transmission buffer. A transmission buffer for SNA protocols is also called a request unit (RU). There is no control information placed before the first transmission NJE record. Figure 9 illustrates the format of a transmission buffer to be transmitted using the SNA protocol.
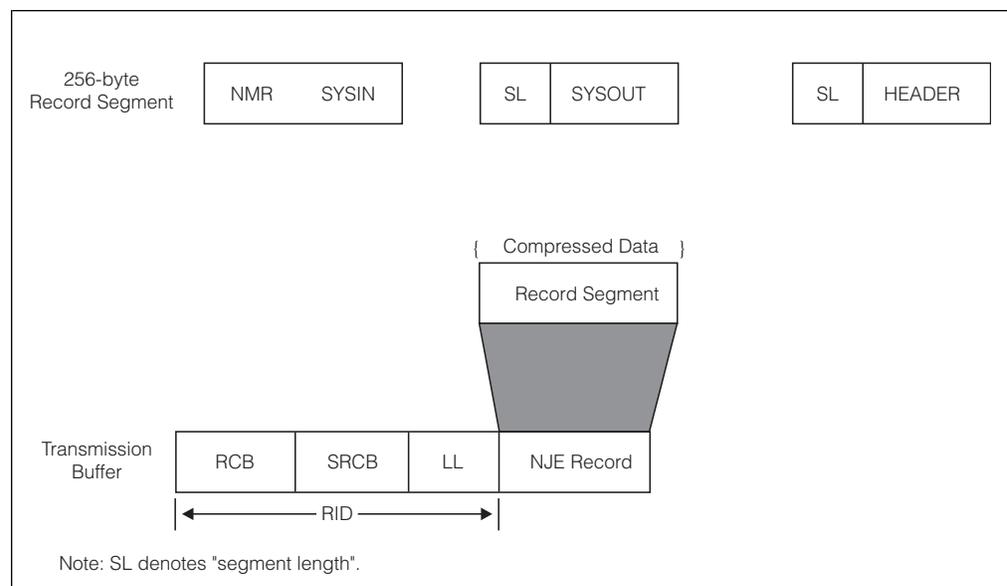
Figure 9. Format of a transmission buffer for SNA protocol

Transmission Buffer: A transmission buffer contains a record control byte (RCB), a subrecord control byte (SRCB), and a logical record length (LRL).

These are placed at the front of the NJE transmission buffer. For the SNA protocol, the RCB, SRCB, and the LL comprise the record identifier. The RCB contains the stream identifier and number; the SRCB defines the type of NJE record contained in the transmission buffer; the LL contains the length of the data prior to compaction and compression, minus 1. Presentation services at the receiving node uses the length byte to locate the next RCB in the inbound RU.

Stream Control

Stream control is the third layer of NJE protocols. After NJE's presentation services packages the NJE record into transmission buffers and creates a stream of buffers, the stream can be transmitted. The stream control layer describes a node's ability to concurrently transfer multiple streams within the same BSC connection or SNA session. This is known as "multileaving".

Multileaving for the SNA and BSC Protocols

NJE protocols permit the identification of:

- Seven SYSIN streams
- Seven SYSOUT streams
- An NMR stream, and
- A control stream in both directions.

The capability of a node to send a stream in one direction while receiving unrelated data in another direction is referred to as "full duplexing". Table 1 compares the SNA and BSC protocols with respect to stream control.

Table 1. Stream Control Comparison Between SNA and BSC Protocols

Stream control	SNA (LU type 0)	BSC
Duplexing	• True full duplex. Each end concurrently can be in send and receive states. The two nodes operate independently. • Is handled by VTAM at the session level.	• Not true full duplex. Each node must flip-flop between the send and receive states. Both ends are synchronized.
Individual Stream Control	• Does not allow for "individual" control of streams	• Allows for "individual" control of streams -- receiving node tells the sending node to hold one or more streams, thus letting the remaining streams continue.
Transmission failures	• VTAM retries most transmission errors.	• The receiving node must notify the sending node of temporary link failures so that the transmission can be retried.

NJE Connection Control

NJE Connection Control is the fourth layer of NJE protocols. JES2 uses the Network Path Manager (NPM) to establish a connection, while the others do not. If one of the connecting nodes is a JES2 complex, then NPM influences the protocols used to establish the connection. This session discusses NJE connection control for complexes establishing a connection.

- Without NPM (no JES2 nodes)
- With NPM
  - Between a JES2 Node and Another Networking Facility
  - Between Two JES2 Nodes.

After the primary and secondary node have established communications, an exchange of signon records occurs. The primary node sends an NJE initial signon record to the secondary node. The NJE initial signon record contains the:

- Name of the primary node. (A member number is also included for multi-CPU complexes.)
- Line and node passwords.
- Size of the BSC buffer.
- Signon concurrence flags, which allow the two nodes to determine the extended capabilities each node supports.

The secondary node responds to the initial signon record with a response signon record.

NJE Connection Control Without NPM

An exchange of control records establishes a connection between two nodes that do not use the NPM. The initial signon record, sent by the "primary" node, is called an "I record". The receiving node, or "secondary" node, sends a response signon record, called a "J record". Extended NJE features exist and are acknowledged by the two nodes by setting bits in their respective I and J records.

Normal disconnection of an NJE node (using BSC or CTC protocols) occurs when the primary node sends a final signoff record, called a "B record". In the SNA protocol, disconnection takes place when the primary node sends a CLSDST or when the secondary node sends a TERMSESS record.

NJE Connection Control With NPM

A JES2 node uses the NPM to establish and manage a connection. NPM notifies other nodes about connections and disconnections to the network and maintains line status information. In controlling network traffic, NPM relies upon a (programmer-specified) line resistance value and the use of connection event sequence (CES) values taken from the system TOD clock.

Between a JES2 Node and Another Networking Facility: In connecting a JES2 node to another networking facility, pre-defined connections are required before I and J records can be sent. Once defined, either node can initiate an NJE network connection. The non-JES2 facility must discard NPM records other than I, J, and B sent by the JES2 node.

Between Two JES2 Nodes: The EBCDIC translation of the installation-supplied node name determines which node is "high" and which is "low". The high end is the primary node; the low end is the secondary node. Either end may initiate a connection or reset a connection if the connection is determined to be invalid. Connection status information is traded between Network Path Managers (NPMs) through Add Connection and Subtract Connection records.

Link Protocols

The lowest layer of the NJE protocols is the link protocols layer, which describes establishing and terminating BSC, CTC, and SNA sessions. Installations using BSC lines or CTC adapters use a link protocol to establish and terminate a session.

SNA sessions use a link protocol to:

- Initiate a session
- Identify the capabilities of each node
- Identify the compaction tables

- Terminate a session.

The following introduces these two separate link protocols.

Initiating a BSC or CTC Session

To initiate a BSC or CTC session, the nodes exchange signon records as shown in Figure 10.
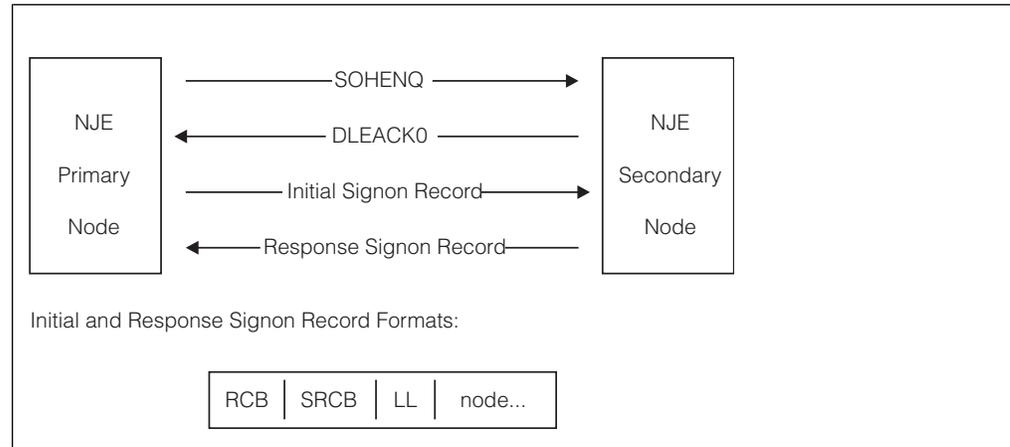
Figure 10. Establishing a BSC or CTC Session

A node using BSC lines or CTC adapters initiates a session by sending the SOH ENQ request. Because both nodes can simultaneously request a session, the node that receives the SOH ENQ request first becomes the **secondary node.** The secondary node acknowledges the request for the session by sending a DLE ACK0 response to node that initiated the session. The **primary node** is the node that receives the DLE ACKO response.

Terminating a BSC or CTC Session

The operator at either the primary or secondary node can issue an operator command to stop the transmission of an NJE job. The termination request is sent as part of a transmission block. After receiving the request, the node can quiesce the transmission buffers, then terminate the connection by sending a sign-off record.
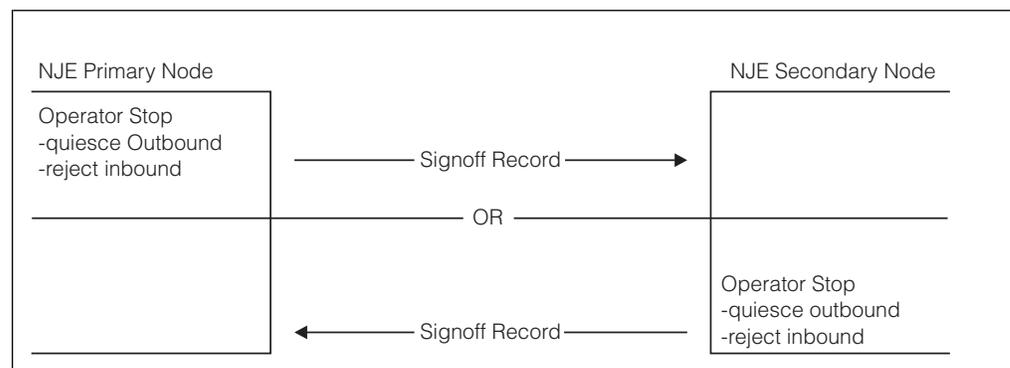
Figure 11. Terminating a BSC or CTC Session

Initiating an SNA Session

To initiate an SNA session, the nodes exchange signon records as shown in Figure 12.
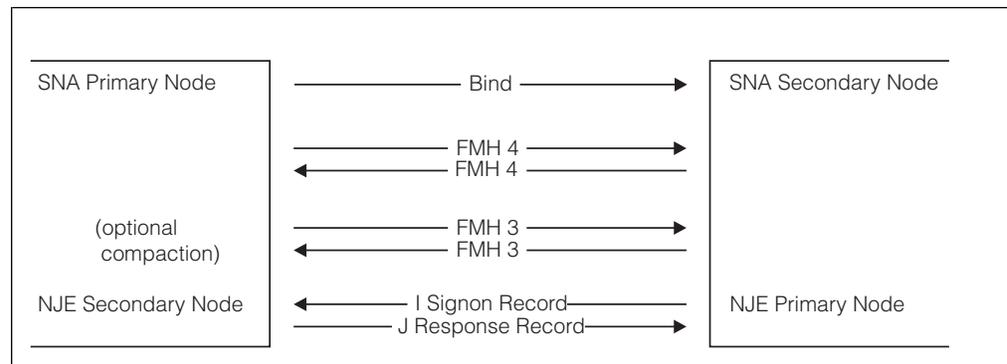
Figure 12. Establishing an SNA Session

To initiate an SNA session, a node sends a BIND request. The node that issues the BIND request is the SNA primary node and the node that receives the BIND request becomes the SNA secondary node. After initiating the session, the SNA primary and SNA secondary nodes must identify their capabilities. The nodes exchange private NJE Function Management Header type 4 (FMH4) records to describe:

- The size of the transmission buffer or request unit (RU) for the node.
- Whether SNA primary node will compact the data in the NJE transmission unit.
- Network topology records.

If the FMH4 records indicate the data in the transmission buffers will be compacted, compaction translation tables are exchanged in FMH3 records. FMH3 and FMH4 records are sent with "definite responses" requested. A negative response by either of the nodes, terminates the session. All other records are sent as "exception response only" records.

After the exchange of FMH records, the nodes are ready to signon. The SNA node with the higher node name (the NJE primary node) sends the signon record. The NJE secondary node responds as in BSC/CTC protocol.

Terminating an SNA Session

The operator can terminate an SNA session at either the SNA primary or the SNA secondary node but only the SNA secondary node can request the termination. Figure 13 on page 15 illustrates termination processing when an SNA secondary node terminates the session. Figure 14 on page 15 illustrates termination processing when the SNA primary node terminates the session.
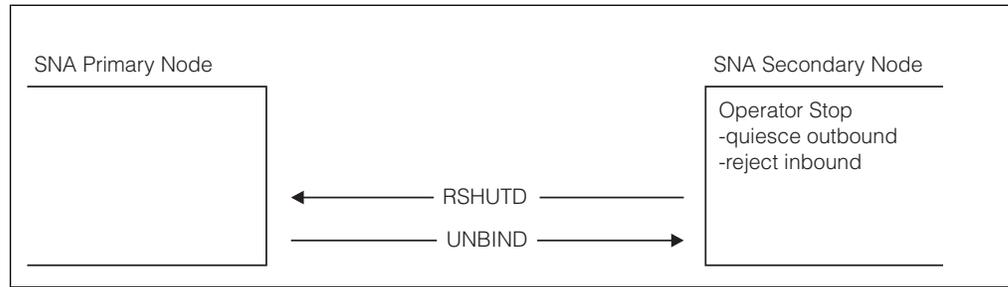
Figure 13. Terminating Processing by an SNA Secondary Node

The SNA secondary node requests the SNA primary node to shut down by sending a RSHUTD request. When the primary receives the request, it quiesces the outbound transmission buffers. When traffic has ended, the SNA primary node sends an UNBIND request. The SNA secondary node should not send any other data following the RSHUTD request.
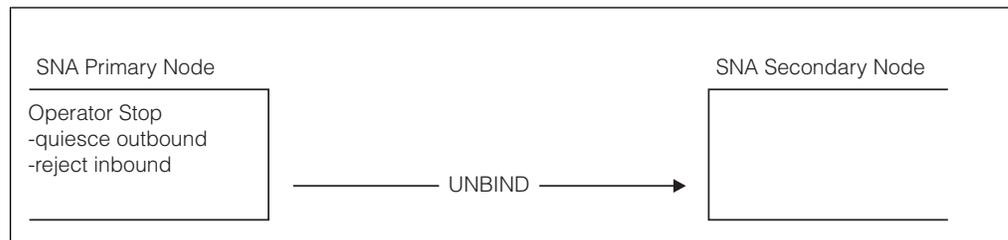
Figure 14. Terminating Processing by an SNA Primary Node

If the operator at the SNA primary node wants to terminate a session, the termination command generates an UNBIND request. If the SNA secondary node initiates a new transmission while the SNA primary node is quiescing the current transmission, an UNBIND request is sent to the secondary to deny the transmission.

Chapter 2. Formats of NJE Transfer Units

Chapter two to seven describes the format of the control data used to transmit a NJE transfer unit across a network. The types of transfer unit a node can transmit are:

- An NJE job, either SYSIN or SYSOUT.
- An NJE Nodal Message Record (NMR) which is a record that contains a command or message.
- An NJE Connection Control Record (NCCR).

Contents of an NJE Job

An NJE job contains either SYSIN or SYSOUT data and the control records used to identify the data being transmitted A node uses these control records to transmit an NJE job:

- **a job header**. See Chapter 3, "Job Header," on page 19 for additional information on the job header.
- **a data set header** if the data is SYSOUT or optionally SYSIN data. If the NJE job contains a job and a SYSIN data set, a record characteristics change section (RCCS) may be included in the NJE job to indicate a change in the length of the records. See Chapter 4, "Data set header," on page 47 for additional information on the data set header.
- **a job trailer**. See Chapter 5, "Job Trailer," on page 119 for additional information on the job trailer.

All NJE jobs contain both a job header and a job trailer. NJE SYSOUT jobs contain one or more data set headers before every SYSOUT data set.

The following sections describe the control records that are found in an NJE job. Each chart contains the:

- **Field name** of the fields found in the control record
- **Offset** or position of the field in the control record
- **Length** of the field or bit. A whole number represents a byte of data. If the length is a fraction, field describes a bit. For example, field NJHGFLG1 is a flag byte that contains 8 bits. The first two bits of field NJHGFLG1 is 2 bits and is reserved for IBM's use.
- **Type of Data** indicates if the data contains characters (char), binary data, or is a bit indicator.

  If the field contains character data, the valid characters are A-Z, 0-9, #, @, and $. If the field is updated by POWER® in addition to the above characters, the fields may contain lower case letters, a hyphen, a period, or a slash.

  If the field is a flag byte, information in the type of data column can also indicate the position of the bit in the field. Each flag byte is composed of 8 bits. The high-order half word of the byte is represented by a 'x0' and the low-order half word of the byte is represented by a '0x', where 'x' is a hexadecimal digit between 0 and F. See Table 2 on page 18 for an illustration of the bit notation used in the type of data column.

Table 2. Example of the Bit Notation Used in the Type of Data Column

Bits	Description
1...	Refers to bit 0. This bit is represented by a 80.
1.1.	Refers to bits 0 and 2. These bits are represented by a A0.
....1	Refers to bit 8. This bit is represented by a 01.
.... ..11	Refers to bits 7 and 8. These bits are represented by a 03.

- **Default Value** indicates the default value of the field for an NJE job.
- **Valid Range** identifies the range of data the field can be in.
- **Description** indicates the use of the field.

The layout of the control records, which is described in the following chapters, may be extended in the future if necessary.

Chapter 3. Job Header

An NJE job must contain a job header to indicate the start of the job. The NJE job can contain either SYSIN or SYSOUT data.

The job header may include several sub-sections depending on the NJE product where the NJE job originated. All NJE products require a prefix section and a general section in the job header.

The following lists the additional sections that can be included in the job header for individual NJE products:

Table 3 describes which sections in the header are built and used by each product.

Table 3. Job Header Sections used by Each Product

Section	JES2	JES3	RSCS	POWER	AS/400
General Section	built and used	built and used	built and used	built and used	built and used
Scheduling Section	built and used	built and used	not built or used	not built or used	not built or used
Security Section	built and used	not built or used	not built or used	not built or used	not built or used
Job Accounting Section	built and used	not built or used	not built or used	not built or used	not built or used
Product-specific Sections	"JES2 Section of the Job Header" on page 42	- none -	- none -	"POWER Section of the Job Header" on page 43	- none -

Additionally, all NJE products may include an installation-defined section. Products may ignore sections that do not apply, but these sections are not be discarded.

Job Header Prefix

Table 4 describes the fields in the prefix section of the job header.

Table 4. Fields in the Prefix Section of a Job Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHLEN	0	2	binary	none	8-32764	The entire length of the job header (including all sections). Note that although the valid length ranges for NJE headers (and individual sub-sections) are specified as being up to 32,764 bytes, the actual lengths for each (sub-)section supported by the NJE systems are defined as the offset of the last field defined in the section plus its length. The NJE systems are designed to receive sections with actual lengths greater than those defined here without error, but data beyond the defined lengths are ignored. Also, the NJE subsystems may have internal restrictions that limit the actual length of the header sections that can be received.
NJHFLAGS	2	1				Reserved for IBM's use.
NJHSEQ	3	1	binary	0	none	The sequence indicator. It contains valid information after the header is segmented for transmission specifying: • The position of the segment within the record • Whether more record segments follow.
		.1	80 bit	0		High order bit means that more segments follow.
		.7	7F bits	0		Low order bits contain the segment number starting with zero (0).

Job Header General Section

Table 5 describes the fields in the general section of the job header.

Table 5. Fields in the General Section of a Job Header

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGLEN	0	2	binary	none	4-32764	Length of the job header general section.
NJHGTYPE	2	1	binary	0	none	Major type identifier, general section.
NJHGMOD	3	1	binary	0	none	Modifier of the major type.
NJHGJID	4	2	binary	0	1-65535	The job number assigned to the NJE job at the originating node. **AS/400** and **RSCS** use numbers 1-9999. **JES2** and **JES3** use numbers 1-65535. **POWER** uses numbers 1-65535.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGJCLS	6	1	char	A	A-Z 0-9	The execution class associated with the job. **AS/400, JES3, and RSCS** set the value for A for jobs that originate at the specified sybsystem. The field is unused. **JES2** sets the class from the job card, JECL statement, or input device class. **POWER** sets the job class from the job's JECL statements.
NJHGMCLS	7	1	char	A	A-Z 0-9	Message class associated with the job. **JES2** sets the message class from the job card. **AS/400, JES3, RSCS**, and **POWER** set the value to A but do not use the field.
NJHGFLG1	8	1	bits	0	none	Flag byte.
bit NJHGF1PR		.1	80 bit	0		The recompute selection priority bit. **AS/400** sets this bit to the default, but does not use it. **JES2** uses the bit if the submitter specifies the job's priority on either the /*PRIORITY JECL or JOB statement. If JES2 is receiving an NJE job and the bit is: • **off**, JES2 gives the NJE job the installation's default priority. • **on**, JES2 uses the priority in NJHGPRIO as the NJE job's priority.
		.3	70 bits	0		Reserved for IBM's use.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
bit NJHGF1CF		.1	08 bit	0		Indicates whether notification of transmission forwarding is to be sent to the notify user (identified by NJHGNTYN/NJHGUSID). Forwarding is defined as the successful transmission of the job or SYSOUT to another node. Notification consists of a status message NMR indicating successful forwarding.
						NJHGF1CF set to 0 indicates that notification of forwarding may be sent from any intermediate node that processes this transmission. NJHGF1CF set to 1 indicates that notification of forwarding is not to be sent.
						AS/400 and JES2 do not use this flag.
						JES3 does not use this flag. JES3 sends notification of transmission forwarding on BSC links.
						RSCS sets the flag according to the SENTMSG parameter on the TAG command.
						POWER does not use this flag. POWER does not send notifications of transmission forwarding.
bit NJHGF1CA		.1	04 bit	0		Indicates whether notification of transmission acceptance is to be sent to notify the user (identified by NJHGNTYN/NJHGUSID). Acceptance is defined as successful receipt (and storage) of the transmission at the destination node. Notification consists of a status message NMR indicating acceptance.
						NJHGF1CA set to 0 indicates that notification of acceptance may be sent from the destination node. NJHGF1CA set to 1 indicates that notification of acceptance is *not* to be sent.
						JES2 does not check this flag and will issue messages $HASP122 or $HASP546 regardless of the setting.
						RSCS sets this flag according to the FINALMSG parameter on the TAG command.
						POWER does not use this flag. POWER always sends a notification of transmission acceptance.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
bit NJHGF1PE		.1	02 bit	0		Indicates that the value in NJHGPASS has been encrypted by the security product at the originating node. An encrypted password has an implicit length of 8 bytes, even if it contains embedded or trailing x'40' or x'00' bytes. **JES2 and JES3** in MVS/SP 3.1.3 with RACF® 1.9 use this (and the next) bit. **RSCS and POWER** do not use or set this (or the next) bit.
bit NJHGF1NE		.1	01 bit	0		Indicates that the value in NJHGNPAS has been encrypted by the security product at the originating node. An encrypted password has an implicit length of 8 bytes, even if it contains embedded or trailing x'40' or x'00' bytes. **Note:** 1. NJHGF1PE=1 and NJHGF1NE=0 indicates that NJHGPASS is encrypted and that NJHGPAS is zeros. 2. NJHGF1PE=0 and NJHGF1NE=1 is not permitted. 3. See the security product for documentation on the encryption algorithm used.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGPRIO	9	1	binary	0	0-F	The selection/transmission job priority. 0 is the lowest priority and 15 is the highest job priority. **JES2** uses this bytes as the selection priority. If the JES2 node is receiving the NJE job and NJHGF1PR is on, JES2 uses the indicator as the execution priority. If the bit is off, JES2 ignores the indicator. If the JES2 node is transmitting an NJE job, JES2 uses NJHGPRIO to indicate the execution priority. **JES3** does not use this byte because NJE jobs are transmitted FIFO. **AS/400** sets this field to "6". Jobs received are processed FIFO. **RSCS** translate their priorities 99-0 to 0-15 on transmission and translate them again when the stream is received. Store-and-forward jobs are never altered, even if their priority is changed while on a z/VM® (RSCS) system. • 0 to 99; 90-99 to 0 • 1 to 92; 84-89 to 1 • 2 to 85; 78-83 to 2 • 3 to 78; 72-77 to 3 • 4 to 71; 66-71 to 4 • 5 to 64; 60-65 to 5 • 6 to 57; 54-59 to 6 • 7 to 50; 48-53 to 7 • 8 to 44; 42-47 to 8 • 9 to 37; 36-41 to 9 • 10 to 31; 30-35 to 10 • 11 to 27; 24-29 to 11 • 12 to 19; 18-23 to 12 • 13 to 12; 12-17 to 13 • 14 to 6; 6-11 to 14 • 15 to 0;0-5 to 15

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGPRIO, cont.	9	1	binary	0	0-F	**POWER** translates NJHGPRIO to the POWER priority (0-9) on received jobs and translates NJHGPRIO from the POWER priority on transmitted jobs. The priority is never changed for store-and-forward jobs, even if the operator alters the value on the POWER node. • 0 to 0; 0 to 0 • 1 to 1; 1 to 1 • 2 to 2; 2 to 2 • 3 to 2; 2 to 3 • 4 to 3; 3 to 5 • 5 to 3; 3 to 5 • 6 to 4; 4 to 7 • 7 to 4; 4 to 7 • 8 to 5; 5 to 8 • 9 to 5; 5 to 8 • 10 to 6; 6 to 10 • 11 to 7; 7 to 12 • 12 to 7; 7 to 12 • 13 to 8; 8 to 13 • 14 to 8; 8 to 13 • 15 to 9; 9 to 15
NJHGORGQ	A	1	binary	0	none	The system qualifier for the originating node in a loosely coupled multi-processor complex. The system qualifier is used to return status messages to the node where the NJE job was submitted. **AS/400** sets the field to the default value but does not use it. **JES2** uses the indicator as the member number. **JES3** uses the indicator as the index to a local main for TSO submitted jobs. If the job originated at a JES3 node and was submitted from a TSO user, the proper index value is set. A qualifier of 0 indicates the job is not from TSO. On output, the field is saved for generating TSO NOTIFY messages. **RSCS** does not set the field, but uses it for status messages. **POWER** sets the field from the POWER SYSID field. POWER uses the field to define the shared spooling system when running in a shared spooling complex. The field may contain X'40', X'F1'- X'F9', or X'00'- X'09'.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGJCPY	B	1	binary	1	none	The number of output copies for the entire NJE job as indicated before the job is executed. **JES2** sets the field from the /*JOBPARM statement. At the destination node, this count is multiple by the data set header copy count for each SYSOUT data set in the job. **AS/400 and RSCS** set the field to the default value but do not use it. **JES3** sets the field to zero at the origin node or defaults it if the execution node equals the origin node. Not used. **POWER** sets the field to zero. The field is unused.
NJHGLNCT	C	1	binary	0	none	The number of lines per page for NJE SYSOUT jobs. A value of X'00' or X'FF' causes the system not to count lines or to use the default value at the node that prints the SYSOUT data set. Any other value causes a page eject after the specified number of lines. **JES2** sets the lines per page for SYSOUT files from the /*JOBPARM or JOB statement. The special values X'00' and X'FF' are used as follows: • X'00'--use the default value of the destination node. • X'FF'--Fo not force any page ejects. Let the skipping be solelydetermined by the carriage control (if present) in the SYSOUT data. **AS/400 and JES3** set the field to the default value but do not use it. **RSCS** sets the value to X'FF' for printing and sets the value to the default for punching. The field is not used. **POWER** uses the default value.
	D	1				Reserved for IBM's use.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGHOPS	E	2	binary	0	1-65535	This field contains a count of the number of nodes (which support this field) that have processed (i.e., stored and forwarded) this transmission. It may be used to detect looping of a transmission in a network.
						Set to zero at the origin node (and not reset for NJE SYSOUT jobs generated from this job) and incremented at intermediate nodes.
						AS/400 does not use or set this field.
						JES3 sets the field to zero at origin nodes for all jobs and increments the field at intermediate nodes, but does not use the field.
						JES2 increments the value and holds the job if it exceeds the installation-defined threshold.
						POWER sets the default value, but does not increment or use the field.
NJHGACCT	10	8	char	blanks	none	The job's network accounting information.
						AS/400 sets the field to the default value but does not use it.
						JES2 sets this field from the /*NETACCT statement or converts the local account number using the NETACCT translation tables when transmitting data. JES2 uses this field as input to the NETACCT translation tables. Blanks can be accepted as the default when data is being received.
						JES3 obtains accounting information from the //*NETACCT control statement.
						RSCS does not set or use the field.
						POWER sets, but does not use this field.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGJNAM	18	8	char	none	none	The name of the job. **AS/400** initializes the field with the job name, using AS4nnnnn. 'nnnnn' conatins the job number specified in field NJHGJID. **JES2** initializes the field with the job name, which can contain blanks. **JES3** initializes the field with the job name. If a job name is not supplied, NJEJOB is used as the default. **RSCS** initializes the field from the VM userid. In Version 3.1, RSCS can set this field from either the VM jobname or the JOBNAME=tag parameter. **POWER** initializesthe field to the POWER job name. If the field is blank when POWER is receiving a job, POWER sets the field to JOB appended with the value specified in NJHGJID.
NJHGUSID	20	8	char	blanks	none	This field contains the destination userid at the NJHGNTYN node for status notification message NMRs associated with the job or SYSOUT transmission. **AS/400** sets this field to the originating userid and uses this field as the originating userid if NJHGORDR is blank or zero. If the field is not set, AS/400 defaults to the userid SYSTEM. **JES2** obtains the information from either the NOTIFY parameter on the JOB statement or the userid specified on the /*NOTIFY JECL statement. Set to the origin userid if none of the above is specified. **JES3** obtains the information either from the userid of the person who submitted the request, if the request was submitted using TSO, or from the //*NETACCT control statement. Otherwise, sets the field to blanks. **RSCS** uses this field as the origin userid if NJHGORGR is blanks or zeros. Set from the origin CMS userid. **POWER** sets the field with the notify userid specified on the NTFY operand.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGPASS	28	8	char or binary	0	none	The password for the job used to validate the user (specified in NJHGORGR) at the execution node. **AS/400** sets the field to the default value, but does not use it. **JES2** obtains the value from the JOB statement. JES2 blanks the field when transmitting an NJE SYSOUT job and does not use the field if receiving an NJE SYSOUT job. **JES3** obtains the value from the job's JOB JCL statement **RSCS** sets the default to binary zeros but does not use the field. **POWER** sets the field to the value specified for the PWD operand of the *$$ JOB statement. The password is used to control access to POWER's spool file.
NJHGNPAS	30	8	char or binary	0	none	The new password. **AS/400** sets the field to the default value, but does not use it. **JES2 and JES3** obtain the new password from the JOB statement. Not used or set for SYSOUT jobs. **RSCS** uses the binary zeros default value for the new password. **POWER** does not use this field.
NJHGETS	38	8	binary	0	none	The time and date the job entered the system in S/370 STCK (store clock) format. The time stamp reflects the Greenwich Mean Time (GMT). **AS/400** sets the time to the default value, but does not use it. **JES2 and JES3** set and use the time stamp for the System Management Facility (SMF). **POWER** sets the time stamp but never uses it.
NJHGORGN	40	8	char	none	none	The name of the node where the job originated (where the user submitted the job). NJHGORGN must never be modified during the life of the NJE job. It may be used as the true origin of the job for security authorization and auditing purposes.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGORGR	48	8	char	blanks	none	The identifier of the user who submitted the job or the name of the remote workstation from which the job was submitted (at node NJHGORGN). **AS/400** sets and uses the field as the originating userid. **JES2 and JES3** set the field as follows for NJE SYSIN jobs: • Set to submitter's userid if submitted via internal reader (e.g., TSO userid if TSO internal reader), • Set to remote name if submitted via RJE workstation, • Set to input divide group name if submitted via input device (JES3 only), • Set to default otherwise. **JES2** propagates the job header from the job of job creating SYSOUT. **POWER** updates this field with the value of the FROM operand, if specified.
NJHGXEQN	50	8	char	blanks	none	The name of the node where an NJE SYSIN job is sent for execution. If an operand issues a command to reroute the NJE SYSIN job, the value in this field will be changed to indicate the new node name. **AS/400** sets this field to the destination address for NJE SYSIN jobs. **JES2** sets the execution node name for an NJE SYSIN job from the: • /*ROUTE XEQ statement • /*XEQ statement • /*XMIT statement **JES3** sets the execution node name for an NJE SYSIN job from the: • //*ROUTE XEQ statement • //*XMIT JECL statement

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGXEQU	58	8	char	blanks	none	The execution userid. **AS/400** sets this field to the destination userid for NJE SYSIN jobs. **JES2** sets the execution userid from the /*XEQ or /*XMIT JECL statement. JES2 does not use the execution userid. **JES3** sets the execution userid from either: • //*ROUTE XEQ • //*XMIT JECL statement. JES3 uses the execution userid for the System Management Facility (SMF). **RSCS** uses the execution userid as the destination virtual machine for SYSIN jobs, and sets it to zeros. For NJE SYSOUT jobs, RSCS sets this field as the origin userid.
NJHGPRTN	60	8	char	blanks		The default destination where the system sends print files that are not specifically routed by the job to be printed. **AS/400** sets this field to the originating address for both NJE SYSIN and NJE SYSOUT jobs, but does not use it. **JES2** obtains the default destination from the job's JCL or from the statement used to define the reader where the job was submitted. **JES3** sets the field from the job's JECL and uses it. It uses the originating nodeas the default destination for NJE SYSOUT jobs. **RSCS** sets the field to the originating node and uses this value as the print destination node for NJE SYSIN jobs. For SYSOUT jobs, RSCS sets the field to the destination node and uses this value as the print destination node. **POWER** sets the field to the originating node or to the value specified on the LDEST parameter.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGPRTR	68	8	char	blanks		The default print output destination (either a userid or a remote workstation) created by the job. **AS/400:** • For SYSIN, this field is set to the originating userid byte not used. **Note:** SYSIN jobs submitted from TSO are set to ANYLOCAL, which implies the SYSTEM at NJHGPRTN). • For SYSOUT, this field is set to the destination userid but not used. **JES2** obtains the default destination from the READER initialization statement or from the PRDEST parameter on the statement that defines the reader where the job was submitted. JES2 uses the destination at the execution node to route the output to if a node is not specified. JES2 uses this field to initialize the destination (field NDHGRMT) in the data set header. **JES3** uses the originating node as the default destination for NJE SYSIN jobs. **POWER** sets the field from the originating userid (or remote workstation) or sets it to the value specified on the LDEST parameter on the JOB statement or the DEST parameter on the LST statement.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGPUNN	70	8	char	blanks		The default punch destination node for punch files that are not specifically routed to another punch destination node. **AS/400:** • For SYSIN jobs, this field is set to the originating address but not used. • For SYSOUT jobs, this field is set to the destination address but not used. **JES2** sets the default punch destination node from either the job's JECL or the PUNODE parameter on the reader where the job was submitted. If neither is specified, JES2 sets the punch destination node to the originating node. **JES3** sets default punch destination node to the originating node for NJE SYSIN jobs. **RSCS** sets the default punch destination node to the oriinating node for NJE SYSIN jobs and to the destination node for NJE SYSOUT jobs. **POWER** sets the default punch destination node from the originating node or from the node specified on the PDEST parameter.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGPUNR	78	8	char	blanks		The default punch output destination (either a userid or a remote workstation). **AS/400:** - For SYSIN, this field is set to the originating userid but not used. **Note:** SYSIN jobs submitted from TSO are set to ANYLOCAL, which implies the SYSTEM at NJHGPUNN. - For SYSOUT, this field is set to the destination userid but not used. **JES2** sets the default punch output destination from either the: /*ROUTE PUNCH statement - PUDEST parameter on the Rnnnn.RDm statement. JES2 uses the default punch output destination at the execution node to: - Route SYSOUT if a destination is not specified - Set field NDHGRMT in the data set header. **JES3** sets the default punch output destination to the remote originating identifier when transmitting an NJE SYSIN job. **RSCS** sets the field to the destination userid or the destination remote workstation. If specified destination is SYSTEM, X'00' is set in the field. For NJE SYSIN jobs, RSCS sets the field to the originating userid or X'00' if the NJE SYSIN job originated from a remote workstation. **POWER** sets the default punch output destination to: - The originating userid - The originating remote workstation - The values specified for the PSEDT parameter on the JOB statement - The values specified for the DEST parameter on the PUN statement.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGFORM	80	8	char	blanks		The default form characteristics for any output created by the job during execution. It is copied into field NDHGFORM in the data set header during execution. **AS/400** sets this field to the form type of the spooled file if the NJE SYSOUT job is sent specifying DTAFMT(*RCDDATA) on the SNDNETSPLF command. **AS/400** does not use this field for any other NJE jobs. **JES2** sets the default form characteristics at the destination node from the job's JECL or from initialization statements. **JES3 and RSCS** do not use the field but set it to zeros **POWER** does not use or set this field.
NJHGICRD	88	4	binary	0		The number of statements in the NJE SYSIN job. **AS/400** sets this field to the number of records in the file for SYSIN jobs and to the default value for SYSOUT jobs. In both cases, **AS/400** does not use the field. **JES3** sets the field to the default value and does not use it. **RSCS** sets the field to the number of records in the file for NJE SYSIN jobs but does not use the field. **POWER** sets the field to the transmitted record count for the NJE job but does not use this field.
NJHGETIM	8C	4	binary	0		The estimated CPU time it will take to execute the job. **AS/400** sets this field to the default value, but does not use it. **JES2** sets the time from the job's JECL and uses the field for SMF records. **JES3** sets the time from the job's JECL and uses it in SMF type 26 records. **RSCS and POWER** set the field to the default but do not use it.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGELIN	90	4	binary	0		The estimated number of output print lines. **AS/400 and POWER** set this field to the default value, but do not use it. **JES2** sets the field from the job's JECL and uses it for SMF accounting. **JES3** sets the field from the job's JECL and uses it for SMF accounting **RSCS** does not use or set the field.
NJHGECRD	94	4	binary	0		The estimated punch card output. **AS/400, RSCS, and POWER** set this field to the default value, but do not use it. **JES2** sets the field from the job's JECL and uses it for SMF accounting. **JES3** sets the field from the job's JECL and uses it for SMF accounting.
NJHGPRGN	98	20	char	blanks		The name of the programmer specified on the JOB statement. **AS/400** sets this field to zeros for NJE SYSIN jobs and to the originating userid for NJE SYSOUT jobs, but does not use the field in either case. **JES3** sets the programmer name from the //*NETACCT statement. **RSCS** sets the name of the submitter for NJE SYSOUT jobs. It is set to X'00' for NJE SYSIN jobs. RSCS does not use the field. In Version 3.1, **RSCS** can set this field from the PGRNAME=tag parameter. **POWER** uses the field for information that is printed on the separator pages for the NJE job's SYSOUT data.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGROOM	AC	8	char	blanks		The room number of the programmer as specified on the JOB statement. **AS/400** sets this field to the default value, but does not use it. **JES2** sets the room number from the JOB statement or the /*JOBPARM statement on its separator page. **Note:** JES2 sets only four characters in this field, left-adjusting them and padding the remainder with blanks. **JES3** sets the room number from the //*NETACCT statement. **RSCS** sets the field from the z/VM (RSCS) distribution code. In Version 3.1, **RSCS** can set this field from the ROOM=tag parameter. **POWER** uses the field for information printed on the separator pages for the NJE job's SYSOUT data.
NJHGDEPT	B4	8	char	blanks		The programmer's department number as specified on the JOB statement. **RSCS** sets the field with the default but does not use it. In Version 3.1, **RSCS** can set this field from the DEPT=tag parameter. **AS/400** sets this field to the default value, but does not use it. **JES2** uses the department number on the separator pages of the NJE job's SYSOUT. **JES3** sets the field with information on the //*NETACCT statement. **POWER** uses the department number on the separator pages of the NJE job',s SYSOUT.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGBLDG	BC	8	char	blanks		The programmer's building number as specified on the JOB statement. **AS/400 and RSCS** set this field to the default value, but do not use it. In Version 3.1, **RSCS** can set this field from the BLDG=tag parameter. **JES2** uses the building number on the separator pages of the NJE job's SYSOUT. **JES3** sets the field with information on the //*NETACCT statement. **POWER** uses the building number on the separator pages of the NJE job's SYSOUT.
NJHGNREC	C4	4	binary	0		The record count for an NJE SYSOUT job. It is not set for an NJE SYSIN job. **AS/400** sets this field to the number of records in the file for NJE SYSIN jobs and to the default value for NJE SYSOUT jobs, but does not use the field in either case. **JES2** sets the field to the sum of record counts for each data set after the job header. The sum does not include the job headers or data set headers in the NJE SYSOUT job. It is used by the node receiving the NJE SYSOUT job to issue a message acknowledging receipt of the job. **JES3** sets the field to the default and does not use it. **RSCS** sets the field from the number of records in the file for NJE SYSOUT jobs. RSCS does not use the field for NJE SYSOUT jobs. For NJE SYSIN jobs, RSCS does not set or use the field. **POWER** uses the field as the line count for the output file. Records containing spanned data count as 1 record.
NJHGJNO	C8	4	binary	0	1-4B	Expanded job number ('4B'=4,294,967,295) JES2 and JES3 set and use this field.
	C8	4	binary	0		Reserved for IBM's use.

Table 5. Fields in the General Section of a Job Header (continued)

Field Name	Offset in Hex	Length in Dec.	Type of Data	Default Value	Valid Range	Description
NJHGNTYN	CC	8	char	blanks		This is the node name of the user to notify about job/SYSOUT transmission or processing status. **AS/400** does not set or use this field. Instead, it sends status messages to NJHGORGN using the originating userid specified by NJHGORGR. **JES2** sets this field to the node name specified via the /*NOTIFY JECL statement. **JES3** sets this field to the node name at which the user submitted the job. **RSCS** does not set this field. **POWER** sets this field to the node name at which the user submitted the job or to the node name specified via the NTFY operand.

Scheduling Section of the Job Header

Table 6 describes the format of the job scheduling section of the job header. The job scheduling section is used by JES2 and JES3 only.

Table 6. Fields in the Job Scheduling Section of the Job Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHELEN	0	2	binary	none	4-32764	Length of the job scheduling section of the job header.
NJHETYPE	2	1	binary	X'8A'	none	The major type identifier of the job scheduling section.
NJHEMOD	3	1	binary	X'00'	none	Modifier of the major type of this section.
NJHEPAGE	4	4	binary	0	none	The estimated number of "begin page" structured fields for page-mode SYSOUT data sets. Each "begin page" record starts with the sequence X'D3A8AF'. **JES3** sets the field from the job's JECL and uses it for SMF accounting.
NJHEBYTE	8	4	binary	0	none	The estimated number of output bytes. This includes all the bytes in the job's SYSOUT data set. **JES3** sets the field fro the job's JECL and uses it for SMF accounting.

Security Section of the Job Header

Table 7 on page 40 describes the format of the Security Section of the Job Header.

The section is identified by an identifier field of X'8C' and a modifier field of X'00'. It may be built as part of a created NJE work element (job or SYSOUT), or may be added to a forwarded work element.

The security section contains information related to the identification of the owner or submitter user associated with the NJE job (SYSIN or SYSOUT). The information in this section is to be treated as a token that is to be interpreted by the security facility component of the trusted computer base.

Table 7. Fields in the Security Section of the Job Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHTLEN	0	2	binary	84	4-32764	Length of the job security section.
NJHTTYPE	2	1	binary	X'8C'	none	The major type identifier of the security section.
NJHTMOD	3	1	binary	0	none	Modifies the major type identifier.
NJHTLENP	4	2	binary	4	4-32764	Length of the prefix sub-section.
NJHTFLG0	6	1	bits	0	none	This flag byte defines the following flags:
Bit NJHTF0JB		.1	80 bit	0	none	Indicates that the security section represents the owner of the NJE job. The bit not set indicates that the security section represents the submitter of the NJE job.
RESERVED	7	1		0	none	Reserved for IBM's use.
	8					End of prefix sub-section
NJHTLENT	0	1	binary	80	80-255	Length of the security data subsection.
NJHTVERS	1	1	binary	X'01'	1-255	Security section version.
NJHTFLG1	2	1	bits	0	none	This flag byte defines the following flags:
Bit NJHT1EN		.1	80 bit	0	none	Indicates that the security section (from NJHTSTYP on) is encrypted. Encryption algorithm is private to the security product.
Bit NJHT1EXT		.1	40 bit	1	none	Indicates the format of the security section. B'1' indicates external format which is used exclusively in NJE.
Bit NJHTSNRF		.1	20 bit	1	none	Indicates that the data set is sent from a system without RACF Version 1.9.
NJHTSTYP	3	1	binary	X'07'	1-255	Security session type. '7' indicates batch job type.
NJHTFLG2	4	1	bits			This flag byte defines the following flags:
Bit NJHT2DFT		.1	80 bit	1	none	Indicates that the security section data has not been verified by the security product at the node which created the security section. If not set, the security data may be assumed to have been verified.
Bit NJHTUNRF		.1	40 bit	1	none	Indicates that the data set is sent from an undefined user that does not have RACF Version 1.9.

Table 7. Fields in the Security Section of the Job Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NJHT2MLO		.1	20 bit	0	none	Indicates that all required ML options were active at the node which created the security section.
Bit NJHT2SHI		.1	10 bit	0	none	Indicates that the security data could not be verified at the node creating the security section.
Bit NJHT2TRS		.1	08 bit	0	none	If set, the user to which this section applies is to be considered as part of the trusted computer base.
Bit NJHT2SUS		.1	04 bit	0	none	If set, the security data was obtained via a surrogate user at the node of creation.
Bit NJHT2RMT		.1	02 bit	1	none	Indicates that the job or data set was originated remotely.
		.1		01 bit		Reserved
NJHTPOEX	5	1	binary	X'03'	1-255	Indicates the port of entry class index. The following port of entry class indices are defined: • 1 = Terminal Class • 2 = Console Class • 3 = JESINPUT Class
RESERVED	6	2				Reserved for IBM's use.
NJHTSECL	8	8	char	none	A-Z,0-9	The security label associated with the user represented by this section.
NJHTCNOD	10	8	char	none	A-Z,0-9	Node at which security section was generated.
NJHTSUSR	18	8	char	none	A-Z,0-9	Userid of submitting user.
NJHTSNOD	20	8	char	none	A-Z,0-9	Node of job submission.
NJHTSGRP	28	8	char	none	A-Z,0-9	Groupid of submitting user.
NJHTPOEN	30	8	char	none	A-Z,0-9	Port of entry name at the creation node for this job.
RESERVED	38	8				Reserved for IBM's use.
NJHTOUSR	40	8	char	none	A-Z,0-9	Security session owner userid.
NJHTOGRP	48	8	char	blanks	A-Z,0-9	Security session owner groupid.

Job Accounting Section

This figure describes the format of the Job Accounting Section of the Job Header. Field lengths can be as large as different components tolerate (MVS allows a 142-byte area for accounting information).

This section is identified by an identifier field of B'8D' and a modifier of B'00'. Field lengths are determined by users.

Note that while filed NJHAJLEN supports a range of only 0-142.

Table 8. Fields in the Job Header Job Accounting Section

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHALEN	0	2	binary	none	10-32k	Length of the job accounting section.
NJHATYPE	2	1	binary	B'8D'	none	Major type identifier of the accounting section.
NJHAMOD	3	1	binary	0	none	Modifier of the major type.
NJHAOFFS	4	2	binary	0	none	Offset of NJHAJLEN.
NJHAFLG1	6	1	binary	0	8-?	Flags.
NJHAF10V		.1	80 bit			Specifies whether or not the job accounting string can be overridden by a non-null string at any node other than where the string originated.
	7	1				Reserved for IBM's use.
NJHAJLEN	8	0-32579	binary	none	0-?	Number of sub-strings.
NJHAJAC1	11	variable	mixed	none	none	Specifies the first substring, followed by other substrings as specified in the NJHAJNR field above.

JES2 Section of the Job Header

This section describes the format of the JES2 section of the job header.

Table 9. Fields in the JES2 Section of the Job Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJH2LEN	0	2	binary	none	4-32764	Length of the JES2 section of the job header.
NJH2TYPE	2	1	binary	X'84'	none	The major type of JES2 section.
NJH2MOD	3	1	binary	0	none	Modifier for the major type of this section.
NJH2FLG1	5	1	bits	0	none	Job level flags.
Bit NJH2DFJG		.1	20 bit	0	none	If on, this is a logging job representing a JOBGROUP.
Bit NJH2FJOB		.02	03 bits			The NJE job was submitted as a batch job if the field is X'00'.
Bit NJH2FSTC		.1	01 bit			The NJE job was submitted under a started task if this bit is on.
Bit NJH2FTSU		.1	02 bit			A time-sharing user submitted the NJE job if this bit is on.
Bit NJH2USE		.1	04 bit			Userid present in NJH2USID.
Bit NJH2TPO		.1	08 bit			The job represents output fro an APPC/MVS transaction program.
NJH2FLG2	5	1	bits	0	none	Job level flags.
Bit NJH2ERST		.1	20 bit			If on, this data set is suppressing job restart records.
Bit NJH2ESMF		.1	80 bit			If on, this data set is suppressing SMF records.

Table 9. Fields in the JES2 Section of the Job Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NJH2ETRC		.1	10 bit			If on, this data set is suppressing trace records.
Bit NJH2EUSR		.1	08 bit			If on, this data set is suppressing user-defined records.
NJH2ACCT	8	4	char	0	none	The submitter's account number.
NJH2USID	C	8	char	blanks	none	A USER SMF field. NJE does not use this field; it is used by the JES2 spool offload facility.
NJH2USR	14	8	char	binary zeros	none	The userid specified on the JCL used to submit the NJE job. NJE does not use this field; it is used by the JES2 spool offload facility.
NJH2GRP	1C	8	char	binary zeros	none	The group identifier specified on the JCL used to submit the NJE job. NJE does not use this field; it is used by the JES2 spool offload facility.
NJH2SUSR	24	8	char	binary zeros	none	The userid of the user who submitted the NJE job. NJE does not use this field; it is used by the JES2 spool offload facility.
NJH2SGRP	2C	8	char	binary zeros	none	The group identifier of the user who submitted the NJE job. NJE does not use this field; it is used by the JES2 spool offload facility.

POWER Section of the Job Header

This section describes the format of the POWER section of the job header.

Table 10. Fields in the POWER Section of the Job Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHPLEN	0	2	binary	none	4-32764	Length of POWER section of the job header.
NJHPTYPE	2	1	binary	X'86'	none	The major type of the POWER section.
NJHPMOD	3	1	binary	0	none	Modifier of the major type of this section.
NJHPFLG1	4	1	bits	0	none	Reserved for IBM's use.
NJHPDISP	5	1	char	D	none	The job's position. POWER obtains the job's disposition form the * $$ JOB statement.
	6	1	binary	0		Reserved for IBM's use.
NJHPSYID	7	1	char	blank	none	The receiving node's system qualifier. POWER uses the system qualifier only if the node uses a shared spool environment.
NJHPUSER	8	16	char	blanks	none	The user information specified on the * $$ JOB statement.
NJHPDSKT	18	2	binary	0	none	The diskette address.
	1A	2	binary	0		Reserved for IBM's use.

Table 10. Fields in the POWER Section of the Job Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHPDD	1C	15	binary	0	none	Due date information indicating the date and time when a SYSIN job is to be scheduled. For a SYSOUT job, this field contains zeros. This field includes the remaining fields in the section.
NJHPDGP1	1C	1	binary	0	none	Due date flags.
Bit NJHPDGP1R		.1	80 bit	0	none	The job must be rerun if the due date expired during the time the system was down.
		.3	70 bits	0	none	Reserved for IBM's use.
Bit NJHPDG1T		.1	08 bit	0	none	The job should run daily.
Bit NJHPDG1W		.1	04 bit	0	none	Weekdays are specified as due dates.
NJHPDG1D		.1	02 bit	0	none	Numbers are specified as due dates.
		.1	01 bit	0	none	Reserved for IBM's use.
NJHPDGP2	1D	1	binary	0	none	Due date flags.
		.7	FE bits	0	none	Reserved for IBM's use.
NJHPDG2X		.1	01 bit	0	none	Information about the due date exists.
NJHPDCY	1E	6	binary	0	none	This field indicates if the job has to be scheduled more than once (for example, every Monday or every first of the every month). This field includes the following two fields.
NJHPDMD	1E	2	binary	0	none	The months in which a job is to be scheduled. X'80'= January, X'40'=February,....
NJHPDDM	20	4	binary	0	none	Days of a month on which a job is to be scheduled: • X'80'=first, X'40'=second, ... • or, if weekdays have been specified: • X'80'=Monday, X'40'=Tuesday, ...
NJHPDN	24	6	binary	0	none	The date and time at which the job is to be scheduled (may be zero, if not yet calculated). If a job is to be scheduled more than once, a new scheduling date is calculated according to the information in NJHPDCY. This field includes the next five fields.
NJHPDNY	24	2	binary	0	1988-2087	The year in which the job is to be scheduled (in packed decimal format without sign).
NJHPDNM	26	1	binary	0	1-12	The month in which the job is to be scheduled (in packed decimal format without sign).
NJHPDND	27	1	binary	0	1-31	The day in which the job is to be scheduled (in packed decimal format without sign).

Table 10. Fields in the POWER Section of the Job Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHPDNTH	28	1	binary	0	0-23	The hour at which the job is to be scheduled (in packed decimal format without sign).
NJHPDNTM	29	1	binary	0	0-59	The minute at which the job is to be scheduled (in packed decimal format without sign).
	2A	1	binary	0		Reserved for IBM's use.
NJHPDGP3	2B	1	binary	0	none	General purpose byte 3
Bit NJHPDG3G		.1	80 bit	0	none	LOG=NO specified
Bit NJHPDG3M		.1	40 bit	0	none	EOJMSG=YES specified
Bit NJHPDG3Q		.1	20 bit	0	none	Queue job completion message
NJHPDIST	2C	8	char	0	none	Distribution code as specified in the * $$ LST or * $$ PUN statements.
NJHPONUM	34	4	binary	0	none	Job number of parent job.
NJHPDAPL	38	8	char		none	Application program-id who submitted the job.
NJHPDUID	40	8	char		none	Userid that submitted the job.
NJHPONOD	48	8	char		none	Node name where job was submitted.
NJHPOUL	50	1	binary	0	none	Node's system qualifier where job was submitted.
	51	3	binary	0	none	Reserved for IBM's use.
NJHPSID	54	8	char		none	Security userid as specified for the SEC operand of the * $$ JOB statement. Used for VSE/ESA security.
NJHPSPW	5C	8	binary	0	none	Security password as specified for the SEC operand of the * $$ JOB statement. Used for VSE/ESA security.
NJHPSECN	64	8	char		none	Security node name as specified for the SECNODE operand of the POWER generation macro. Used for VSE/ESA security.
NJHPPRIV	6C	8	binary	0	none	Programmer's private information
NJHPECHO	74	8	char	blanks	none	Userid for echo-message
NJHPMRIN	7C	25	mixed	blanks/0	none	Information for job-status-messages
NJHPDGP5	95	2	binary	0	none	Reserved for IBM's use
NJHPRQUL	97	1	binary	0	none	Sysid of system which receives job
NJHPMRON	98	4	binary	0	none	Job number of job-status-message
	9C	32	binary	0	none	Reserved for IBM's use

Installation-Defined Section

This section is identified by an identifier field of B'11xxxxxx' and a modifier field of the installation's choice. This section is not used by any IBM [3] products, but will be passed through the network.

Beyond the four-byte definition, this section is the responsibility of the installation. The section length is limited to 32764 bytes. The combined length of all sections in the header is limited to 32764 bytes, although product implementations may be more restrictive. The length must be reflected in the NJHULEN field in the front of the control record section header. The following depicts a sample eight-byte field:

Table 11. Fields in the Job Header Installation Extension Section

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJHULEN	0	2	binary	none	4-32764	Length of the section.
NJHUTYPE	2	1	binary	none	C0-FF	The major type of the installation-defined section.
NJHUMOD	3	1	binary	none	none	Modifier for the major type identifier.
NJHUCODE	4	4	char	none	none	This could be used for a GUIDE/SHARE installation code, or some other unique identifier.

---

3. IBM is a trademark of the IBM Corporation.

Chapter 4. Data set header

An NJE job begins with a job header, is followed by data, and ends with a job trailer. The data between the job header and the job trailer can be either SYSIN or SYSOUT data.

An NJE SYSIN job contains JCL for a job and one or more SYSIN data sets. It may or may not contain data set headers. If the SYSIN data set has record characteristics other than 80-byte fixed records, then it is preceded by a data set header containing only a record characteristics change section. See "Record Characteristics Change Section" on page 68.

An NJE SYSOUT job contains one or more SYSOUT data sets. Each SYSOUT data set is preceded by one or more data set headers. A data set header contains a prefix section and a basic general section, which may be followed by any of the following sections:

- **General Section** -- for SYSOUT from all products.
- **RSCS Section** -- for SYSIN from JES2, POWER, and AS/400.
- **3800 Section** -- for output to be printed on a 3800 printer.
- **Output Processing Section** -- for advanced function printing [4] output.
- **Security Section** -- for resource authorization control.
- **Product Section** -- for output created at a VM/RSCS or a VSE/POWER node.

Table 12 describes which sections in the header are built and used by each subsystem.

Table 12. Data set header sections used by each subsystem

Section	JES2	JES3	RSCS	POWER	AS/400
"Data Set Header General Section"	built and used for SYSOUT only	built and used for SYSOUT only	built and used for SYSOUT only	built and used for SYSOUT only	built and used for SYSOUT only
"Record Change Characteristics Section"	built and used for SYSIN only if the record length (LRECL) is not greater than 80 bytes, or if the record format (RECFM) is not fixed format.	not built, used, nor forwardedby JES3 built and used by BDT	not built, but used for forwarding SYSIN	built and used	built and used for SYSIN only
"3800 Section"	built and used for 3800 attributes	built and used for 3800 attributes	built and used for 3800 attributes	built and used for 3800 attributes	built and used for 3800 attributes
"Output Processing Section"	built and used for Advanced Functioning Printer (AFP) attributes	built and used for AFP attributes	built and used for AFP attributes	built and used for AFP attributes	built and used for AFP attributes

---

4. Advanced Function Printing is a trademark of the IBM Corporation.

Table 12. Data set header sections used by each subsystem (continued)

Section	JES2	JES3	RSCS	POWER	AS/400
"Security Section'	built and used beginning with JES2 3.1.3	built and used beginning with JES3 3.1.3	neither built nor used	neither built nor used	neither built nor used
Product Sections	none	none	RSCS	POWER	none

Data set header prefix

The data set header begins with the standard four-byte control record prefix containing the combined length of all sections. Table 13 describes the format of the prefix section of the data set header.

Table 13. Fields in the Prefix Section of a Data Set Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHLEN	0	2	binary	none	4-32764	The Combined length of all sections in the data set header.
NDHFLAGS	2	1	binary	none	none	Reserved for IBM's use.
NDHSEQ	3	1	binary	none	none	The sequence indicator. It contains valid information after the header is segmented for transmission. The information specifies: • The position of the segment within the record. • More record segments to follow.
	.1	80 bit	0			High order bit means that more segments follow.
	.7	7F bit	0			Low order bits contain the segment number starting with zero (0).

Data set header general section

The basic general section is identified by an identifier field of X'00' and a modifier field of X'00'. Table 14 describes the format of the basic general section of the data set header.

Table 14. Fields in the general section of a data set header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGLEN	0	2	binary	none	4-32764	Length of the data set header general section.
NDHGTYPE	2	1	binary	none	none	The major type of this section. General sections are identified by a X'00'.
NDHGMOD	3	1	binary	0	none	Modifier for the major type identifier. A modifier code of X'00' is the modifier for the general section.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGNODE	4	8	char	none	N/A	The destination node for the data set. This field may be set at execution time from either NJHGPRTN or NJHGPUNN. **AS/400** sets and uses this field as the destination address for the data set.
NDHGRMT	C	8	char	blanks	N/A	Either the destination userid, remote workstation, or printer identifier. This field may be set at execution time from either NJHGPRTR or NJHGPUNR. **AS/400** sets and uses this field as a destination userid. If the received userid is blanks or zeros, this product substitutes 'SYSTEM' as the destination userid. **JES2 and JES3** also use this field for the TSO/E TRANSMIT and RECEIVE commands. If this field and the NDHGXWTR field are the same, this field is used as the TSO destination userid. **RSCS** sets the field to the destination userid or printer (or X'00' when SYSTEM is the destination). **POWER** uses the field to route the NJE job to a remote workstation, userid, or program.
NDHGPROC	14	8	char	blanks	none	The JCL procedure that was executing when this data set was created. **AS/400**: • For print files, this field is set to the originating userid and used for the spooled file user data information. • For punch files, this field is set from the database file member name and used as the member name for the database file. **RSCS** sets the field from the information in FILENAME. RSCS does not use the field. **POWER** defaults this field and does not use it.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGSTEP	1C	8	char	blanks	none	The step name that was executing when the data set was created. For print files, **AS/400** sets this field from the spooled file name and uses this field for the spooled file name. For punch files, **AS/400** sets this field to the database file name and uses this field as the database file name. **RSCS** sets the field from the information in FILETYPE. RSCS does not use the field. **POWER** defaults this field and does not use it.
NDHGDD	24	8	char	blanks	none	The DDNAME that referenced the data set when it was created. **AS/400** sets this field to the default value, but does not use it. **JES3** If blank or zero on input, JES3 uses the NJHGROOM value as a DDNAME. If NJHGROOM is blank this system uses a default DDNAME of '*NJEWKnn', where 'nn' is an internal index value. **RSCS** and **POWER** default the field and do not use it.
NDHGDSNO	2C	2	binary	0	0-32767	A counter incremented each time a SYSOUT data set is allocated within the same job. This field maintains the uniqueness of data sets, while allowing them to be spun off. **AS/400** sets this field to the default value, but does not use it. **JES2** uses and sets the field as follows: 1. Ignores it on input if less than or equal to 100. 2. Sets it from the PDBDSKEY field if less than 32767, then it will be set to zero. There is no internal JES2 limit to this number. There is a storage limitation in the converter to 4096 datasets in one job. **JES3, RSCS, and POWER** default the field and do not use it.
	2E	1				Reserved for IBM's use.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGCLAS	2F	1	char	A	A-Z 0-9	The job's SYSOUT class. **AS/400** sets this field from the CLASS parameter on the SNDNETF and SNDNETSPLF commands, but does not use it. **JES2** sets the SYSOUT class to A if the class specified in the job's JCL is not alphanumeric. **JES3** sets the field to the SYSOUT class. If receiving an NJE SYSOUT job, this field is used if the class and type (indicated by NDHGF2PR and NDHGF2PU) do not conflict with the local system's use of class/type. If a conflict exists, a default class is used for the appropriate type. **RSCS** uses the field for the VM output class and sets it from this class unless overridden by the SYSOUT keyword on the TAG. **POWER** sets the SYSOUT class to the output class. POWER Version 4.1.2 supports numeric classes for output. For all versions prior to 4.1.2, only A-Z are valid (set to A if not in the range A-Z when received).
NDHGNREC	30	4	binary	0	none	The count of the number of records in the data set. **AS/400** sets this field, but does not use it. **JES2** uses the field as the line count of the data set. A spanned record counts as 1, regardless of the number of segments. **JES3** sets the number of records in the data set but does not use the information. **RSCS** uses the information in a message to the user. RSCS sets the field from the record count in the spool file block. **POWER** sets and uses the field as the line count.
NDHGFLG1	34	1	bits	0	none	A flag byte containing the following flags:

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NDHGF1SP		.1	80 bit	0	none	The spin data set flag. **AS/400** sets this bit off and does not use it. **JES2** sets the bit from information in the job's JCL. JES2 sometimes sets this bit for the TSO/E TRANSMIT and RECEIVE commands according to the data set disposition. **JES3** does not use this bit but sets it for spin data sets and data sets generated by the TSO/E TRANSMIT and RECEIVE commands. **RSCS** uses the bit to separate data sets into individual spool files, but does not set this bit. **POWER** sets the bit only if the data set is segmented and uses the bit to determine the job number. If the flag is on, the job number is set from the job header (NJHGJID), otherwise a new job number is assigned.
Bit NDHGF1HD		.1	40 bit	0	none	The destination node is to hold the data set. **AS/400** sets and reads this bit to determine whether the print file should be held before printing. **JES2** sets the bit from information in the JCL and uses it to place the data set on the system output hold queue. **JES3** sets the bit and uses it to place the data set on the system output hold queue. **RSCS** uses the bit to indicate to the destination node that it is to hold the data set. The bit is set when specified by the originator on the TAG command using the HOLD keyword. **POWER** uses the bit and sets it if the data set disposition is 'H' or 'L'.
Bit NDHGF1LG		.1	20 bit	0	none	The job log indicator bit. **AS/400, JES3, RSCS, and POWER** do not use the bit but set it to the default value. **JES2** uses the bit to indicate that the data set is the JES2 job log.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NDHGF1OV		.1	10 bit	0	none	The page overflow indicator. If the bit is on, the data set is allowed to print over the page fold. If off, the printer should start at the top of each page. **AS/400** sets this bit on for print files, but does not use it. **JES2 and RSCS** do not use this bit but set it with the default value. **JES3** sets the bit if OVFL=OFF is specified on the //*FORMAT statement in the job's JCL. JES3 uses the bit as described above. **POWER** uses the field but does not set it.
Bit NDHGF1IN		.1	08 bit	0	none	The interpreter feature is to be used for punched output if punched on a device that has the interpreter feature installed. **Note:** For further interpretation of this field, see the NDHGF1DF field (01 bit). **AS/400, RSCS and POWER** do not use this bit but set it to the default value. **JES2 and JES3** use the information from the job's JCL to set the bit.
Bit NDHGF1LC		.1	04 bit	0	none	The NDHGLNCT field (offset X'3A') contains information if this bit is on. **AS/400** sets this bit on for print files, but does not use it. **JES2** sets and uses the bit if field NDHGF1LC is set on. **JES3, RSCS, and POWER** set this bit to the default value, but do not use it.
Bit NDHGF1ST		.1	02 bit	0	none	If on, the job log data set contains statistics for the job. **AS/400** sets this bit off and does not use it. **JES2** uses the bit to avoid multiple instances of the job statistics in the job log. **JES3, RSCS, and POWER** do not use this bit, but set it to the default value.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NDHGF1DF		.1	01 bit	0	none	Specifies whether the interpret bit (NDHGF1IN) was explicitly set off (1) or not (0). The NDHGF1DF is used with the NDHGF1IN bit as follows: ```\nNDHGF1IN 0 1 0 1\nNDHGF1DF 0 0 1 1\n\n			|\n V	V |\nMeaning: Use	Do not |\n default	interpret |\n V V\n Interpret Interpret\n``` **AS/400** sets this bit to the default value, but does not use it. **JES2, RSCS, and POWER** default this bit and do not use it.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description

NDHGRCFM	35	.5	binary	0	none	Specifies the record format of the data set.
						
						`11.. ....   undefined format`
						`10.. ....   fixed format`
						`01.. ....   variable format`
						
						AS/400:
						
						For print files, this system sets either variable or fixed format (bits 1 and 1).
						
						For punch files, **AS/400** sets this field to X'80'.
						
						JES2 uses the RECFM in the DCB specified in the job's JCL to set the field. The DCB defines other settings that are not defined by this field. However, JES2 allows all flag settings and transmits them across the network.
						
						JES3 uses the RECFM in the DCB specified in the job's JCL to set the field beginning with z/OS release 1.8.
						
						RSCS sets this field to X'42' for print files, and X'80' for punch files.
						
						For print files, it is used to decide what record length to use in determining whether to make an incoming file a virtual 1403 file or a 3211 file. (For further details, see the following field, NDHGLREC.)
						
						For punch files, this field is used with NDHGLREC to determine the proper record length to use in detecting files containing punch records greater than 80 characters. It is also used in determining whether to forward punch records of a store-and-forward file with or without carriage control.
						
						This field is modified for certain types of store-and-forward data when RSCS acts as an intermediate node.
						
						POWER does not use this field, but sets it to C'Cx' (undefined) and supports all values of the field.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description

NDHGRCFM	35	.5	binary	0	none	Specifies the control characteristics used in the data set.
						
						`... .00. no carriage control`
						`... .10. ASA control characteristics`
						`... .01. machine control`
						`        characteristics`
						`... .11. MODCA or AFPDS print files`
						
						AS/400:
						
						For print files this system sets:
						• carriage control to ASA, machine, or none (bits 5 and 6 off).
						• MODCA or AFPDS print files (bits 5 and 6 on).
						
						For punch files, **AS/400** sets this field to X'80'.
						
						JES2 uses the RECFM in the DCB specified in the job's JCL to set the field.
						
						JES3 sets the carriage control to ASA, machine, or none (bits 5 and 6 off). Only carriage control bits are used on input.
						
						RSCS sets this field to X'42' for print files, and X'80' for punch files.
						
						For punch files, this field is used to determine whether to forward punch records of a store-and forward file with or without carriage control.
						
						This field is modified for certain types of store-and-forward data when RSCS acts as an intermediate node.
						
						For Version 3.1, this last restriction is removed.
						
						POWER does not use this field, but sets it to X'Cx' (undefined). At store-and-forward nodes, files having ASA carriage control will not have the CC converted to machine carriage control.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description

NDHGLREC	36	2	binary	0	1-32760	The maximum logical record length (LRECL) of any record that appears within the data set. It includes the carriage control character if specified for the data set.
						
						AS/400 sets this field to:
						• X'80' for punch files. When **AS/400** receives punch files with records greater than 80 bytes, **AS/400** truncates them at the destination node.
						• X'32,760' for AFPDS print files prior to Version 2.1.
						• the maximum record length of all records in the file for AFPDS, LINE, and FPDSLINE files in Version 2.1 and all subsequent releases. **AS/400** records the longest record in the file when it is first spooled.
						• X'4079' or '512' for all SCS, IPDS and ASCII files.
						Note: S/370 does **not** support these file types; you should route these through a S/370 only to get the final **AS/400** destination.
						
						JES2 obtains the LRECL from the DCB. It includes the carriage control character.
						
						JES3 uses the firld if it is not 0. The following shows how JES3 defaults the field according to the type of data set and the presence of carriage control:
						
						`Print with CC   133  (132 without CC)`
						`Punch with CC    81  (80 without CC)`

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description

NDHGLREC, cont.	36	2	binary	0	1-32760	**RSCS** uses this field as follows:
						
						For incoming print files, this determines whether they are defined as 1403, 3211, or 3800 files.
						
						`    Record Length        Device`
						`       1 - 132            1403`
						`     133 - 150            3211`
						`     151 - 204            3800`
						`205 - 32765VAFP (if supported`
						`on a VM system)`
						
						punch records greater than 80 (greater than 81 with carriage control) are truncated at the destination node. In setting the field, RSCS obtains the LRECL from TAGRECLN (with one added to each record to account for the CCTL byte) as follows:
						
						1. This field is set to 80 for punch files.
						
						2. Virtual 3800 files that contain any load CCWs have this field set to 8192 (regardless of the actual maximum record length in the file).
						
						RSCS does not modify this field as an intermediate node.
						
						POWER uses the field and sets it to the maximum length of any record in the data set, up to a maximum record length of 32K-1.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGDSCT	38	1	binary	0	0-255	The data set copy count. This field defines the number of copies to print or punch at the destination node and will be used if the data set is printed on an impact printer. If it is printed on a non-impact printer, the copy group count is used. **AS/400** sets this field: • to 0 for punch files. • to the number of copies left to print, then reads this field to determine the total number of copies for print files. **JES2** sets the data set copy count to the number of copies for the data set. **JES3** sets and uses the data set copy count. If the value in the field is 0, one copy of the data set is produced. **RSCS** sets the field from the copy count specified on the SPOOL command. RSCS uses the field to set the spool copy count. **POWER** sets the field with the information supplied from the COPY parameter of the '* $$ LST' or '* $$ PUN' statement. It is used as the copy count (defaulted to 1).
NDHGFCBI	39	1	binary	0	-31 to 31	Used as an index byte when loading the FCB on a 3211. It allows the user to print each line with the data shifted right or left by (up to) 31 characters. A negative value indicates left indexing; a positive value indicates right indexing. Right indexing adds leading blanks; left indexing removes data characters. **AS/400** sets this field to the default value, but does not use it. **JES2** supports both right and left indexing. **JES3 and POWER** do not use this byte but set the byte to the default value. **RSCS** does not use the field. Set from the INDEX keyword on the TAG statement. (Only positive values may be specified.)

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGLNCT	3A	1	binary	0	none	The default lines per page for SYSOUT files. See this field only when the NDHGF1LC field is (offset X′01′ bit) is on. A value of X′00′ or X′FF′ causes the system not to count lines or use a local default value, depending on the system. All other values are treated as an explicit number of lines on a page before a page eject is generated by the printing subsystem. **AS/400** sets this field to the default value, but does not use it. **JES2** obtains the value from the job's JCL. The special values X′00′ and X′FF′ are used as follows: **X′00′** (default): Use the default value of the printing subsystem. **X′FF′** : Do not force any page ejects. Let the skipping be solely determined by the carriage control (if present) in the SYSOUT data. **JES3, RSCS, and POWER** do not use this field, but set it to the default value.
	3B	1				Reserved for IBM's use.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGFORM	3C	8	char	blanks	N/A	The name of the form to use when printing or punching the data set. **AS/400** sets this field to blanks if AS/400 formtype is *STD or greater than 8 characters. In all other cases, the field is set equal to AS/400 formtype. **JES2 and JES3** use only the first four bytes, followed by blanks. Forms are set to zero if not specified. If received, either zero or blanks indicate the standard (STD) forms are desired. **RSCS** sets the field with the value supplied by CP FORM unless that value is overridden by the value supplied by TAG. The default is only set when the CP SPOOL command specifies NULL for the FORM name. This field is used for the CP FORM when not defaulted. When the header field contains the default (blanks), the CP FORM name is set to the installation's default. **POWER** uses only the first four bytes because POWER has only four-character forms names. For a 3800 SYSOUT data set, a value of X'00' indicates that hardware defaults are to be used. A value of X'40' indicates that software defaults are to be used. When the job originated on a non-POWER system (POWER section was not present), X'00' is used as the software default.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description

NDHGFCB	44	8	char	blanks	N/A	The name of the FCB to be used if this data set is printed.
						
						AS/400 sets the field to blanks for print files, then reads the field to determine the printer filename to open when spooling the files.
						
						JES2 uses only the first four bytes, followed by blanks. FCB is set to zero if not specified. If received, either zero or blanks indicate the default FCB (****) is desired. The first four bytes are prefixed with:
						
						`'FCB1'    1403`
						`'FCB2'    3211 and 3203-5`
						`'FCB3'    3800`
						`'FCB4'    4245 and 4248`
						
						JES3 sets and uses the field, but the default is zero.
						
						RSCS uses this field as the FCB on spool file. It is set from FCB on the SPOOL command. Only the first four characters are used. The field may be overridden by the CP TAG command (up to eight characters).
						
						POWER uses the field as the FCB name. The field is set from the FCB name if one was specified. The last four characters may be set to '$$$$' for device-independence. For a 3800 data set, a value of X'00' indicates that hardware defaults are to be used; a value of X'40' indicates that software defaults are to be used. When the job originates from a non-POWER system (POWER section was not present), X'00' is used as the software default.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description

NDHGUCS	4C	8	char	blanks	N/A	The name of the universal character set (UCS) to use if this data set is printed. It is more commonly known as the "print train".
						
						AS/400 sets the field to blanks for print files and does not use it.
						
						JES2 and JES3 use only the first four bytes, followed by blanks. UCS is set to zero if not specified. If received, either zero or blanks indicate that the default UCS (****) is desired. NDHATAB1 is used if UCS is specified as "****", blanks, or zeros. JES2 prefixes the first four bytes with:
						
						`'UCS1'    1403`
						`'UCS2'    3211`
						`'UCS3'    3203-5`
						`'UCS5'    4245`
						`'UCS6'    3262-5 and 4248`
						
						RSCS does not use the field. RSCS sets the field with the value specified on the UCS keyword in the TAG.
						
						POWER sets the field with the UCS name specified on the * $$ LST statement.

NDHGXWTR	54	8	char	blanks	N/A	The name of the external writer to process this data set. If equal to the NDHGRMT field, this field is also set and used by the TSO/E TRANSMIT and RECEIVE commands and the CMS SENDFILE command to allow files to be sent to individual MVS users. **AS/400** sets the field equal to NDHGRMT, but does not use it. **JES2 and JES3** set the field to the name of the external writer. A default of zero is set; either zero or blanks are accepted as defaults for input. **RSCS** does not use this field but sets it to the value specified of the EXTWTR keyword if specified on TAG statement. **POWER** does not use the field but sets it to the external writer name as specified on the DEST parameter.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGNAME	5C	8	char	blanks	A-Z,0-9	This is the data set name qualifier associated with the data set at the creation node. It may be user specified or generated by the system.
						AS/400 sets this field to the default value, but does not use it.
						JES2 and JES3 set from the DSNAME= keyword on the DD JCL statement, or default it to a single question mark '?'. It is used to construct a resource name associated with this data set for purposes of resource access authorization.
						RSCS does not use the field and sets it to the default.
						POWER does not use the field and sets it to the default.
NDHGFLG2	64	1	bits	0	none	Flag byte containing the following flags:
Bit NDHGF2PR		.1	80 bit	0	none	If on, the data set is to be printed.
						AS/400 sets and uses this bit for print files.
						JES2 does not use the bit, but sets it if the output class is a print class at the output execution node.
						JES3 uses this bit to denote print output. If there is print output, JES3 sets this bit.
						RSCS uses the bit to cause the data set to be spooled to a printer. RSCS sets the bit if the input file is a print file.
						POWER uses the bit to denote print output. It is set if the input file is a print file.
Bit NDHGF2PU		.1	40 bit	0	none	If on, the data set is to be punched.
						AS/400 sets and uses this bit for punch files.
						JES2 does not use the bit, but sets it if the output class is a punch class at the output execution node.
						JES3 uses this bit to denote punch output. If there is punch output, JES3 sets this bit.
						RSCS uses the bit to cause output to be spooled to a punch device. RSCS sets the bit if the input file is a punch file.
						POWER uses the bit to denote punch output. If there is punch output, POWER sets this bit.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NDHGF2RM		.1	20 bit	0	none	If on, the FILENAME and FILETYPE are taken from the NDHGPROC and the NDFGSTEP fields respectively. **AS/400** sets this bit, but does not use it. **JES2 and JES3** do not use the bit and set it to the default. **RSCS** uses the bit to take the FILENAME and FILETYPE from the NDHGPROC and the NDHGSTEP fields respectively. **POWER** does not use the bit and sets it to the default.
Bit NDHGF2HB		.1	10 bit	0	none	If on, the data set is to be held prior to any processing. **AS/400** sets this field to the default value, but does not use it. **JES2** sets this bit whenever the output disposition is either HOLD or LEAVE. **JES3** sets this bit whenever the output disposition of a data set is either HOLD or LEAVE. When this bit is set on, **JES3** also sets bit NDHGF1HD to hold the data set at nodes not supporting output dispositions. **JES3** uses this bit and bit NDHGF2HA to determine the output disposition of an incoming data set. **RSCS, and POWER** do not use or set this field.
Bit NDHGF2HA		.1	08 bit	0	none	If on, the data set is to be held after any processing. **AS/400** sets this field to the default value, but does not use it. **JES2** sets this bit whenever the output disposition is either KEEP or LEAVE. When this bit is set on, **JES2** also sets flag NDHGF1HD to hold the data set at nodes not supporting output dispositions. **JES3** sets this bit whenever the output disposition of a data set is either KEEP or LEAVE. When this bit is set on, **JES3** also sets bit NDHGF1HD to hold the data set at nodes not supporting output dispositions. **JES3** uses this bit and bit NDHGF2HB to determine the output disposition of an incoming data set. **RSCS, and POWER** do not use or set this field.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NDHGF2HX		.1	04 bit	0	none	JES3 uses this bit to indicate that the data set should be held for external writer at JES3 nodes. **AS/400, JES2, RSCS, and POWER** do not use or set this field.
Bit NDHGF2TR		.1	02 bit	0	none	If on, trailing blanks should be truncated. JES3 sets this bit to indicate the data had its blanks truncated when it was created and should have them truncated when the data set is received at the targeted JES3 node. **AS/400, JES2, RSCS, and POWER** do not test or set this bit.
Bit NDHGF2NO		.1	01 bit	0	none	If on, this data set is a 'non-output' SYSOUT data set. The data set is not to be printed or punched; instead it contains internal control information. JES2 uses this bit for its JCL images and step completion codes data sets (EVENTLOG). **AS400, JES3, RSCS, and POWER** do not test or set this bit.
NDHGUCSO	65	1	bits	0	none	The UCS option byte.
Bit NDHGUCSD		.1	80 bit	0	none	If on, indicates the UCS will be loaded with the block data check option. **AS/400, JES2, and RSCS** do not use this bit but set it to the default value. (JES2 always blocks data checks.)
Bit NDHGUCSF		.1	40 bit	0	none	If the bit is on, the UCS will be loaded with the FOLD option. **AS/400** sets this bit to the default value, but does not use it. **JES2, JES3, and RSCS** do not use this bit but set it to the default value. JES2 loads the FOLD option on a character set if bit X'40' of the first byte (byte 0) is on in the UCS image.
		.6	3F bits			Reserved for IBM's use.
	66	2				Reserved for IBM's use.

Table 14. Fields in the general section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHGPMDE	68	8	char	blanks	none	The process mode for the data set specified by the user. The process mode does not necessarily imply the presence of a particular type of record; it only indicates a request by the user for the preferred type of output processing. This field is used for SYSOUT selection and scheduling. The IBM-defined process modes are: • LINE • PAGE • SOSI1 • SOSI2 **AS/400** sets this field to the appropriate process mode depending on AS/400 devtype, igcdta, and igcosi. If none of these settings are appropriate, the field is set to blanks. See OPTB for PRMODE, see key 0018 for other values when these are not appropriate. If neither NDHGPMDE nor PRMODE are set, or the PRMODE is a value not recognized by the AS/400, then the following values will be used: *LINE or *AFPDSLINE if NDHGRCFM bits are B'.01' or B'.10' *AFPDS if NDHGRCFM bits are B'... .11.' **JES2 and JES3** set the field from the PRMODE keyword on the OUTPUT JCL statement. If PRMODE is not specified, then the data itself is examined to determine the process mode. **RSCS** moves this field into an XAB and sets it from an XAB (if existing) for VM/PSF. For Version 3.1, **RSCS** sets this field from the PRMODE= tag parameter. **POWER** sets this field from the PRMODE operand on a JECL statement, but does not use this field.
NDHGSEGN	70	4	binary	0		The segment number of a data set which has been segmented. **AS/400** does not read or set this field. **JES2** sets this field if the segment number is less than or equal to 99999, and uses this number on the default separator page. **JES3 and POWER** do not read or set this field.
	74	4				Reserved for IBM's use.

Record Characteristics Change Section

This section is identified by an identifier field of X'00' and a modifier field of X'40'. This is only used on SYSIN data and is not required if all SYSIN data is RECFM F and LRECL 80. When present, it is sent as the only section in the data set header, without the "basic" general section, having a modifier of X'00'.

JES3 does not use, send, or store-and forward these sections.

Table 15. Fields in the Record Characteristics Change Section of a Data Set Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHCLEN	0	2	binary	none	4-32764	Length of the record characteristics change section of the data set header.
NDHCTYPE	2	1	binary	none	none	The major type of this section. The record characteristics change section of the data set header is identified by an identifier field of X'00'.
NDHCMOD	3	1	binary	0	none	Modifier for the major type identifier. A modifier code of X'40' indicates this is the record characteristics change section of the data set header.
NDHCFLG1	4	1	byte	0	none	Reserved for IBM's use.
NDHCRCFM	5	1	binary	0	none	The record format of the SYSIN data set. The bit settings and associated record formats are: ```\n11.. .... undefined format\n10.. .... fixed format\n01.. .... variable format\n``` **AS/400** sets this field to indicate fixed format but does not use it. **RSCS** does not create this section of the header. It uses this field at a store-and-forward node to determine if a SYSIN file should be stored as coded NOPs (done whenever NDHCRCFM indicates that the file is not in fixed format). **POWER** always has an undefined record format.
NDHCLREC	6	2	binary	0	0-252	The maximum record length of records in the SYSIN data set that follows. **AS/400** sets this field but does not use it. **RSCS** uses this field at a store-and-forward node to determine if a SYSIN file should be stored as coded NOPs (done whenever NDHCLREC indicates that the record length is not 80). **POWER** allows a maximum value of 128.

3800 Section

This section is identified by an identifier field of X'00' and a modifier field of X'80'.

Table 16. Fields in the 3800 Section of a Data Set Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHALEN	0	2	binary	none	4-32764	The length of the 3800 section of the data set header.
NDHATYPE	2	1	binary	none	none	Defines the major type of this section. General sections are identified by a X'00'.
NDHAMOD	3	1	binary	0	none	Modifier for the section identifier. For the 3800 section of the data set header, there must be a modifier code of X'80'.
NDHAFLG1	4	1	byte	0	none	Flag byte containing the following flags:
Bit NDHAF1J		.1	80 bit	0	none	If on, the data set contains table reference characters. The bit is set if OPTCD=J is specified. **AS/400** sets and reads this bit for TRC. **RSCS** *uses* this bit as follows: This flag is set as follows: 1. On for all virtual 3800 files. Any virtual 3800 file will have TRC bytes inserted (in all records except those representing CCWs for intermediate operations and all spanned records). Any select CAT CCWs in these files will be removed. 2. Off for non 3800 files unless OPTCD=J is specified on the TAG command. **Note:** If this flag is on, the incoming file is made a virtual 3800 print file. **POWER** uses this flag and sets it when a Spool-Access Support (SAS) user indicates that the first character is a TRC.
Bit NDHAF1BR		.1	40 bit	0	none	If on, the 3800 burster is to be used. **AS/400** sets this bit off and does not use it. **JES2 and JES3** use this bit and set it from the job's JCL. **RSCS** does not use the bit but sets it if BURST=Y is specified in the TAG. **POWER** uses the bit and sets it from information on the * $$ LST statement or from the default printer setup.

Table 16. Fields in the 3800 Section of a Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
Bit NDHAF1BN		.1	20 bit	0	none	If on, the 3800 burster should not be used. **AS/400** sets this bit off and does not use it. **JES2** does not use the bit, but sets it to the default value. **JES3** sets the bit if STACKER=C is specified on the JES3 //*FORMAT statement to indicate continuous forms stacking. **RSCS** does not use the bit but sets it if BURST=N is specified on the TAG command. **POWER** uses the bit and sets it from the * $$ LST statement or from the default printer setup.
Bit NDHAF1BN		.5	1F bits	0	none	Reserved for IBM's use.
NDHAFLCT	5	1	binary	0	0-255	The 3800 flash count defining the number of copies of this data set to be flashed. If this field is not specified, but NDHGFLSH was specified, then all copies are flashed. **AS/400** sets this bit off and does not use it. **JES2 and JES3** use and set this field from the job's JCL. **RSCS** sets the field using the CP SPOOL command or it may be overridden by the CP TAG command. RSCS uses the flash count with the spool file. **POWER** uses the field and sets it from information on the * $$ LST statement or from the default printer setup.
NDHATREF	6	1	binary	0	0-3	The table reference character that specifies which of the four translate table entries to use when printing the copy modification. **AS/400** sets this bit off and does not use it. **JES2 and JES3** obtain the information from the job's JCL. **RSCS** uses this field and sets it using the CP SPOOL command or it may be overridden by the MODTRC keyword on the TAG command. **POWER** uses the field and sets it from information on the * $$ LST statement or from the default printer setup.
	7	1				Reserved for IBM's use

Table 16. Fields in the 3800 Section of a Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHATAB1	8	8	char	blanks	N/A	The name of the first translate table. A translate table defines an index to a font that is defined for a 3800 printer. This means that, by using different translate tables, different character sets may be used within one print output. The translate table to be used is defined in the record when OPCTD=J is specified. Only the first four characters are used and set. **AS/400** sets and reads this field. **JES2** obtains the name of the translate table from the job's JCL or JECL. If received, zeros or blanks indicate that JES2 should default the table name to zero. For NDHATAB1 only, if the field is set to ****, blanks, or zeros, JES2 uses the name specified in NDHGUCS. **JES3** defaults the field to zeros. **RSCS** uses this field and sets it using the CHARS operand on the CP SPOOL command. It may be overridden using the CHARS keyword on the TAG command. **POWER** uses the field and sets it from the * $$ LST statement. For a 3800 SYSOUT data set, a value of X'00' indicates the hardware defaults should be used. A value of X'40' indicates the software defaults should be used. When the job originates on a non-POWER system (POWER section is not present), X'00' is used as a software default.
NDHATAB2	10	8	char	blanks	none	The name of the second translate table. Only the first four characters are used and set. **AS/400** sets and reads this field. **JES2, JES3, RSCS, and POWER:** See the individual product descriptions in the NDHATAB1 field.
NDHATAB3	18	8	char	blanks	none	The name of the third translate table. Only the first four characters are used and set. **AS/400** sets and reads this field. **JES2, JES3, RSCS, and POWER:** See the individual product descriptions in the NDHATAB1 field.

Table 16. Fields in the 3800 Section of a Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHATAB4	20	8	char	blanks	none	The name of the fourth translate table. Only the first four characters are used and set. **AS/400** sets and reads this field. **JES2, JES3, RSCS, and POWER:** See the individual product descriptions in the NDHATAB1 field.
NDHAFLSH	28	8	char	blanks	none	The flash cartridge identifier used to produce pre-printed forms. This overlay will be printed on every page before the data is placed into position. Only the first four bytes are used. **AS/400** sets the field to blanks and does not use it. **JES2 and JES3** use the field and set it from the job's JCL or JECL. If the user does not specify a value when submitting the job, it is set to zero. A value of NONE in the first four bytes indicates that no flashing is to be performed, regardless of device or installation default. **RSCS** uses the field and sets it from the CP flash name. It may be overridden by the FLASH keyword on the TAG command. **POWER** uses the field and sets it from the * $$ LST statement. For a 3800 SYSOUT data set, a value of X'00' denotes the use of hardware defaults; a value of X'40' denotes the use of software defaults. When a job originates on a non-POWER system (POWER section is not present), X'00' is used as the software default.

Table 16. Fields in the 3800 Section of a Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHAMODF	30	8	char	blanks	none	The copy modification id. It contains the name of a module placed on every page of the output data set when it is being printed. Only the first four bytes are used. **AS/400** sets the field to blanks and does not use it. **JES2** uses the field and sets it from information in the job's JCL or JECL. If the user does not specify a value when the job is submitted, the copy modification id is defaulted to 0. **JES3** uses and sets the copy modification id as specified in the job's JCL or JECL. The last four bytes are zeros. (If copy mode is not specified, the entire field consists of zeros.) **RSCS** uses and sets the field from the CP MOD name. It may be overridden by the MODIFY keyword on the TAG command. **POWER** uses and sets this field from the * $$ LST statement. In the VSE/SP 3800, the user can specify an additional character arrangement table to be used for the copy modification, which does not need to be specified in the CHARS parameter. If the user does this, POWER defaults to the first CHAR. For a 3800 SYSOUT data set, a value of X'00' denotes the use of hardware defaults; a value of X'40' denotes the use of software defaults.

Table 16. Fields in the 3800 Section of a Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHACPYG	38	8*1	binary	0	0-255	Identifies the copy groups. It is actually eight one-byte fields. The fields define the number of times each page of the data set will be copied when the data set is sent to the 3800 printer. The first field refers to the first transmission, the second to the second transmission, etc. In this case, NDHGDSCT is not used. The number of transmissions is determined by the number of copy groups defined. The sum of all copy groups, or a single copy group, cannot exceed 255. **AS/400** sets the field to blanks and does not use it. **RSCS** uses the sum of all bytes as the CP copy count and turns on the CP copy group flag. The CP copy count goes into the first byte if the CP copy group flag is on. This may be overridden by the COPYG keyword on the TAG command. **POWER** uses the field and sets it from information specified on the * $$ LST statement or by a program.

Output processing section

The Output Processing Section is used for transmitting output processing parameters and is identified by an identifier field of X'89' and a modifier field of X'00'. This section's length is variable and depends on the type and number of output processing parameters being transmitted.

The individual output processing parameters are represented by self defining data structures contained within the Output Processing Text Block (OPTB) which follows a fixed length portion of the section. The parameter representations are termed Output Processing Text Units (OPTUs or TUs) and their structure is defined in "Output Processing Text Block" on page 81. The set of NJE defined OPTUs are defined in "OPTU Key Definitions" on page 82.

The 3800 section of the data set header is still built if the user specifies any 3800 attributes. Some fields are duplicated in the OPTB. However, if any of the 3800 fields are changed by operator command, those changes need only be reflected in the 3800 section, not in the OPTB. The 3800 section fields override the OPTB at the SYSOUT destination.

Table 17 on page 75 describes the format of the output processing section of the data set header. The output processing section includes several fields that describe the *length* of parts of the output processing section. All length fields include the length of their own field. For example, field NDHSLEN is the length of the entire output processing section including the length of field NDHSLEN.

Note: JES2 and JES3 call this section the "data stream" section in their control block references.

Table 17. Fields in the output processing section of a data set header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHS	0	0	label	none	none	The beginning of the output processing section.
NDHSLEN	0	2	binary	none	none	The length of the entire output processing section. It includes the two bytes used to hold the length.
NDHSTYPE	2	1	binary	X'89'	none	The identifier for an output processing section.
NDHSMOD	3	1	binary	X'00'	none	The modifier for the output processing section, which must be X'00' to indicate OUTPUT OPTBs. The values from X'01' to X'FF' are reserved.
NDHSFLEN	4	2	binary	none	none	The length of the fixed area of the section, down through the field NDHSGPID, but not including NDHSOPTB or any of the OPTBs. It includes these two bytes used to hold the length.
NDHSFLG1	6	1	bits	X'00'	none	Flag byte.
Bit NDHSCPDS		.1	80 bit	0	N/A	If set, this flag indicates the presence of at least one CPDS data stream record in the data set. **Note:** The presence of "stream mode" data (SRCB = X'B0') and "CPDS" data are semantically identical in the NJE protocols. **AS/400** sets the field to 0, but does not use it.
	7	1				Reserved for IBM's use.
	8	8				Reserved for IBM's use.
NDHSNSTR	10	4	binary	none	none	The number of "begin page" structured fields in the data set (for example, the number of data records that begin with the sequence X'D3A8AF'). This field is not adjusted for multiple copies. **AS/400** sets this filed to the default value, but does not use it. **RSCS** does not set or use this field.

Table 17. Fields in the output processing section of a data set header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHSGPID	14	8	char	blanks	none	The output group name. This name may be provided by the user; otherwise, it is generated at job execution time. This name is used by the receiving node to determine how the job's data sets are grouped together for printing. **AS/400** sets the field to the default value, but does not use it. **JES2** sets the field from the user's output JCL or generates it at job execution time. **JES3, RSCS, and POWER** do not use this field and set it to the default value.
NDHSOPTB	1C	0	mixed	none	none	The start of the variable area containing Output Processing Text Units (OPTUs). The variable area consists of a prefix area followed by the OPTUs and mapped by IEFSJPFX in MVS.
NDHSPRID	1C	4	char	'SJPF'	none	A required constant that identifies the OPTB prefix. **AS/400** sets the field to the default value, but does not use it. **RSCS** and **POWER** do not use this field.
NDHSVERS	20	1	binary	X'02'	none	A required constant that identifies the version of the prefix. This constant should be set, but may be ignored by non-MVS systems. The settings indicate the following MVS systems: • 01 = MVS/SP 1.3.1 (FMID JBB1327) • 02 = MVS/SP 1.3.3 (FMID JBB1329) and later **AS/400** sets the field to the default value, but does not use it.
NDHSPLEN	21	1	binary	X'1C'	none	The prefix length, which is used to point to the beginning of variable-length OPTUs (start of OPTBs = NDHSOPTB + NDHSPLEN). This field is fixed at X'1C' for the NDHSVERS=02 version.
NDHSDLEN	22	2	binary	X'0000'	none	The length of the data (all OPTUs) following the prefix.
NDHSVERB	24	8	char	'Output'	none	A constant required for compatibility with MVS systems. This constant identifies the JCL statement normally used to specify output processing parameters in the MVS environment. **AS/400** sets the field to the default value and does not use it.
	2C	8	char	blanks	none	Reserved for IBM's use.

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHSFLG2	34	1	bits	X'80'	none	Flag byte reserved for IBM's use.
	35	1				Reserved for IBM's use.
	36					The start of the OPTUs. (The previous file is the end of the OPTB prefix.)

Security Section of the Data Set Header

Table 18 describes the format of the Security Section of the Data Set Header.

The section is identified by an identifier field of X'8C' and a modifier field of X'00'. It may be built as part of a created NJE work element (job or SYSOUT), or may be added to a forwarded work element.

The security section contains information related to the identification of the owner or submitter user associated with the data set described by the corresponding data set header. The information in this section is to be treated as a token that is to be interpreted by the security facility component of the trusted computer base.

Table 18. Fields in the Security Section of the Data Set Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHTLEN	0	2	binary	84	4-32764	Length of the data set security section.
NDHTTYPE	2	1	binary	X'8C'	none	The major type identifier.
NDHTMOD	3	1	binary	0	none	Modifies the major type identifier.
NDHTLENP	4	2	binary	4	4-32764	Length of the prefix sub-section.
NDHTFLG0	6	1	bits	0	none	This flag byte defines the following flags:
bit NDHTF0JB		.1	80 bit	0	none	Indicates that the security section represents the owner of the NJE job. The bit not set indicates that the security section represents the submitter of the NJE job.
RESERVED	7	1		0	none	Reserved for IBM's use.
	8					End of prefix sub-section
NDHTLENT	0	1	binary	80	80-255	Length of the security data sub-section.
NDHTVERS	1	1	binary	X'01'	1-255	Security section version.
NDHTFLG1	2	1	bits	0	none	This flag byte defines the following flags:
bit NDHT1EN		.1	80 bit	0	none	Indicates that the security section (from NDHTSTYP on) is encrypted. Encryption algorithm is private to the security product.
bit NDHT1EXT		.1	40 bit	1	none	Indicates the format of the security section. B'1' indicates external format which is used exclusively in NJE.

Table 18. Fields in the Security Section of the Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHTSTYP	3	1	binary	X'07'	1-255	Security session type. '7' indicates batch job type.
NDHTFLG2	4	1	bits			This flag byte defines the following flags:
bit NDHT2DFT		.1	80 bit	1	none	Indicates that the security section data has not been verified by the security product at the node which created the security section. If not set, the security data may be assumed to have been verified.
bit NDHT2MLO		.1	20 bit	0	none	Indicates that all required ML options were active at the node which created the security section.
bit NDHT2SHI		.1	10 bit	0	none	Indicates that the security data could not be verified at the node creating the security section.
bit NDHT2TRS		.1	08 bit	0	none	If set, the user to which this section applies is to be considered as part of the trusted computer base.
bit NDHT2SUS		.1	04 bit	0	none	If set, the security data was obtained via a surrogate user at the node of creation.
bit NDHT2RMT		.1	02 bit	1	none	Indicates that the job or data set was originated remotely.
NDHTPOEX	5	1	binary	X'03'	1-255	Indicates the port of entry class index. The following port of entry class indices are defined: • 1 = Terminal Class • 2 = Console Class • 3 = JESINPUT Class
RESERVED	6	2				Reserved for IBM's use.
NDHTSECL	8	8	char	none	A-Z,0-9	The security label associated with the user represented by this section.
NDHTCNOD	10	8	char	none	A-Z,0-9	Node at which security section was generated.
NDHTSUSR	18	8	char	none	A-Z,0-9	Userid of submitting user.
NDHTSNOD	20	8	char	none	A-Z,0-9	Node of job submission.
NDHTSGRP	28	8	char	none	A-Z,0-9	Groupid of submitting user.
NDHTPOEN	30	8	char	none	A-Z,0-9	Port of entry name at the creation node for the job creating this data set.
RESERVED	38	8				Reserved for IBM's use.
NDHTOUSR	40	8	char	none	A-Z,0-9	Security session owner userid.

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHTOGRP	48	8	char	blanks	A-Z,0-9	Security session owner groupid.

RSCS Section

Table 19 describes the format of the RSCS section of the data set header.

Table 19. Fields in the RSCS Section of a Data Set Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHVLEN	0	2	binary	none	4-32764	Length of the RSCS section of the data set header.
NDHVTYPE	2	1	binary	none	none	The major type of this section. The RSCS section of the data set header is identified by X′87′.
NDHVMOD	3	1	binary	0	none	Modifier for the major type identifier. A modifier code of X′00′ is the modifier for the RSCS section.
NDHVFLG1	4	1	byte			Reserved for IBM's use
NDHVCLAS	5	1	char	none	A-Z 0-9	The CP spool file class.
NDHVIDEV	6	1	binary	none	none	The originating CP device type. (See the CP DEVTYPES macro for valid values.)
NDHVPGLE	7	1	binary	none	none	The 3800 virtual page length.
NDHVDIST	8	8	char	blanks	none	The CP distribution code.
NDHVFNAM	10	12	char	blanks	none	The CP file name.
NDHVFTYP	1C	12	char	blanks	none	The CP file type.
NDHVPRIO	28	2	binary	50	0-99	The RSCS transmission priority as specified on the TAG command.
NDHVVRSN	2A	1	binary	none	none	The version number of the RSCS system that created the header.
NDHVRELN	2B	1	binary	none	none	The release number of the RSCS system that created the header.
NDHVTAGR	2C	136	char	none	none	The TAG record as specified on the TAG command.
NDHVDSNO	B4	2	bin	0	none	The List processor data set number.

POWER Section

Table 20 describes the format of the POWER section of the data set header.

Table 20. Fields in the POWER Section of a Data Set Header

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHPLEN	0	2	binary	none	4-32764	The length of the POWER section of the data set header.

Table 20. Fields in the POWER Section of a Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHPTYPE	2	1	binary	none	none	The major type of this section. The POWER section of the data set header is identified by a X'86'.
NDHPMOD	3	1	binary	0	none	Modifier for the major type identifier. A modifier code of X'00' indicates the POWER section of the data set header.
NDHPFLG1	4	1	byte			Reserved for IBM's use.
NDHPIDEV	5	1	binary	0	none	The POWER device type.
NDHPPRIO	6	1	char	none	none	The output priority. This field is defaulted to the priority specified in the POWER generation.
NDHPDISP	7	1	char	D	none	The output disposition. Valid values are D (delete after printing), K (keep), H (hold), and L (leave).
NDHPUSER	8	16	char	blanks	none	This field contains any specified user information. It is used for printing on a separator page.
NDHPJBSF	18	1	binary	0	none	The job suffix number, which is created for segmented output. It is similar to the spin data set number.
NDHPSYID	19	1	char	zero	none	The system qualifier for a shared spool configuration. It is used for printing on a separator page.
NDHPNSEP	1A	1	binary	0	none	The number of separator pages that should be printed for this output.
NDHPOPTN	.1	1	bits	0	none	The COPYSEP parameter specified for the output.
NDHPPART	1C	2	char	blank	none	The partition identifier where the job executed. It is used for printing on a separator page.
	1E	2				Reserved for IBM's use
NDHPRCFM	20	1	binary	0	none	Record Format
Bit NDHGPRCSC		.1	80 bit	0	none	SCS print format
Bit NDHGPRCBM		.1	40 bit	0	none	BMS mapping format
Bit NDHGPRC32		.1	20 bit	0	none	3270 format
Bit NDHGPRCAP		.1	10 bit	0	none	APA (CPDS) format
Bit NDHGPRCES		.1	08 bit	0	none	Escape mode format
Bit NDHGPRCAS		.1	04 bit	0	none	ASA control characters
NDHGPRCMC		.1	02 bit	0	none	Machine control characters
NDHPJNUM	22	2	binary	0	none	The job number.
NDHPCOMP	24	4	char	blank	none	The compaction table name used for output destined to RJE/SNA.

Table 20. Fields in the POWER Section of a Data Set Header (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHPPASS	28	8	char	blank	none	The password that has been specified for this output.
NDHPSETP	30	68	char	blank	none	The default SETPRT parameter list.
NDHPSTRT	74	8	char	blank	none	The time (in STCK format) when output spooling started.

Output Processing Text Block

The output processing textblock (OPTB) is a fixed area followed by a sequence of Output Processing Text Units (OPTUs). Because the number of OPTUs and their sequencing is arbitrary, the appearance of any specific OPTU is optional. You can supply installation-defined defaults when OPTUs with values necessary to process data do not appear at a destination.

OPTB Structure Rules

The section below represents the rules for building an OPTB. These rules have been written in the Backus normal form metalanguage (BNF) format, where elements of the structure are indicated in brackets (such as <ELEMENT>) and the definition of the element appears at the right (:= definition). The sequence of elements allowed is indicated by explicit sequences of bracketed items. Alternative sequences are delimited by the "|" symbol, which means "OR".

Table 21. Rules for Building an OPTB

Element	Definition
<DSHOPT>:	= Output processing section of the data set header = < FIXED> <OPTB>
<FIXED>:	= Fixed portion of the output processing section starting with NDHSLEN through NDHSGPID
<OPTB>:	= <PREFIX> <OPTUS>
<PREFIX>:	= Fixed portion of the OPTB starting with NDHSPRID through the RESERVED field at offset X'36'
<OPTUS>:	= <OPTU> | <OPTUS> <OPTU>
<OPTU>:	= <KEY> <COUNT> <PARMS>
<KEY>:	= Two-byte key defined in keyword definition table
<COUNT>:	= Two-byte count of number of parms (1-255)
<PARMS>:	= <PARM> | <PARM> <PARM>
<PARM>:	= <LL> <DATA>
<LL>:	= Two-byte data length (1-128)
<DATA>:	= Sequence of exactly <LL> bytes

OPTU structure definition

A unique 2-byte registered NJE key identifies OPTUs. Each OPTU includes a specific type and maximum number of data elements that represent keyword parameter values. The number of these parameters included in the OPTU is specified in a 2-byte count field that follows the key and precedes the parameter list:

KEY	COUNT	LENGTH1	data1	LENGTH2	data2	. . .	. . .
key	cc	11	data	11	data	. . .	. . .

key

Specifies the 2-byte registered keyword identifier. All values not defined are reserved for IBM's use and should not be specified.

cc(1-16383)

Specifies the 2-byte count of the number of values supplied for the keyworkd parameter.

A count of 0 is used to indicate either a missing positional] parameter or a defaulted parameter. For a count of 0, no data elements should follow the count field.

ll (1-16383)

Specifies the two-byte length of the parameter value. A length of 0 indicates a null value.

For compatibility with MVS systems, the parameter length is restricted by the protocols to 128 bytes. Lengths greater than 128 will be correctly stored and forwarded by JES, but cannot be used at an MVS destination.

data

parameter value associated with the key.

OPTU Key Definitions

The suggested defaults listed in the following table may be used when the OPTU is not contained in the OPTB.

Table 22. OPTU Key Definition Table

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0001	1	1	Reserved	Reserved - private JES2/JES3.
0002	4	4	Font	Set and ready by AS/400, JES2, and JES3.
0003	1	2	CKPTLINE	Binary: range 0-32767. Maximum number of lines contained in a logical page. This value is used to determine when to take checkpoints for printed output or SNA data sets. Installation defaults may be used. (Not supported by VM or AS/400.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0004	1	2	CKPTPAGE	Binary: range 1-32767. Number of logical pages to be printed or transmitted before the next output data set checkpoint is taken. This value represents the number of pages transmitted as a single SNA chain to an SNA work station. (Not supported by AS/400.)
0005	1	2	CKPTSEC	Binary: range 1-32767. Number of seconds that may elapse between printer checkpoints. Installation defaults may be used. (Not supported by VM or AS/400.)
0006	1	1	Reserved	Reserved - private JES2/JES3.
0007	1	8	COMPACT	Symbolic name from 1-8 alphameric characters long, used to determine the compaction table when sending the SYSOUT data set described by this control statement to an SNA remote terminal. This specification overrides any remote device-defined compaction table. Installation defaults may be used. (Not supported by VM or AS/400.)
0008	1	1	CONTROL	X'80' - Force single space. X'40' - Force double space. X'20' - Force triple space. X'10' - Use first character in line as CC. Installation defaults may be used. (Not supported by VM.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0009	1	1	Copies	Set and read by AS/400, JES2, and JES3 for the number of copies to print.
000A	8	1	Reserved	Reserved - private JES2/JES3.
000B	8	1	Reserved	Not used at output nodes.
000C	1	17	Reserved	Reserved - private JES2/JES3.
000D	1	4	Reserved	Reserved - private JES2/JES3.
000E	1	4	Reserved	Reserved - private JES2/JES3.
000F	1	1	Reserved	Reserved - private JES2/JES3.
0010	1	8 for JES2/JES3 10 for AS/400	FORM	Set and Read by AS/400, JES2, and JES3. AS/400 uses the system default of the *STD type if the value of the first parameter is 0.
0011	1	8	Reserved	Reserved - private JES2/JES3.
0012	1	1	INDEX	Binary: range 1-31. Indicates the data set indexing print position offset (to the right) for the 3211 printer. (Not supported by VM or AS/400.)
0013	1	1	Reserved	Not used at output nodes.
0014	1	1	LINDEX	Binary: range 1-31. Indicates the data set indexing print position offset (to the left) for the 3211 printer. (Not supported by VM or AS/400.)
0015	1	1	Reserved	Reserved - private JES2/JES3.
0016	1	4	Reserved	Reserved - private JES2/JES3.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0017	1	1	Reserved	Reserved - private JES2/JES3.
0018	1	8 for JES2/JES3 16 for AS/400	PRMODE	Set and read by **AS/400** and **JES3** for the data stream type. Possible values for AS/400: **'SCS'** - Standard Character String with no SOSI processing **'SCS1'** - SCS with SOSI1 processing **'SCS2'** - SCS with SOSI2 processing **'SCS3'** - SCS with SOSI3 processing **'IPDS'** - IPDS (intelligent printer d s) with no SOSI processing **'AFPDS'** - AFPDS (advanced function printing d s) with no SOSI processing **'LINE'** - LINE with no SOSI processing **'SOSI1'** - LINE or PAGE with SOSI processing **'SOSI2'** - LINE or PAGE with SOSI2 processing **'SOSI3'** - LINE or PAGE with SOSI3 processing **'PAGE'** - PAGE with no SOSI processing **'ASCII'** - USERASCH with NO SOSI processing **'ASCIICS'** - SCS ASCII with no SOSI processing

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0019	1	1	PRTY	Binary: range 0-255 (255 is hishest priority.) Represents the priority of the output data set for output queuing. (Not supported by VM.) AS/400 print file priority ranges from 1 to 9 with 1 representing the highest priority. • **AS/400** sets this field by converting the AS/400 print file priority to the PRTY value through the algorithm: (X=AS/400 print file priority) PRTY keyword value=(9-X)*31 • **AS/400** reads this field by converting the PRTY value to the AS/400 print file priority through the algorithm: (X=AS/400 print file priority) If X < 248, then AS/400 priority=9-(x/31) Else, If X > = 248, then AS/400 priority=9-(248/31)
001A	1	1	TRC	**AS/400, JES2, and JES3** read and set this field for table reference characters. In AS/400: X'80'-No TRC exists. X'40'-TRC exists.
001B	1	4	Reserved	Reserved - private JES2/JES3.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
001C	1	8	WRITER	1-8 character name (consisting of any combination of alphamerics, #, @, or $) of an installation-written program in the system library that is to write the output data set. (Not supported by VM.)
001D	1	1-6 1-8 for AS/400	FORMDEF	Specifies the 1-6 character (A-Z, 0-9, #, @, or $) member name of the installation's form definition library that contains information which the page-mode printer uses to print a data set. The first two characters of the member name are pre-defined IBM standards and prefixed to the name specified here. The members can contain the following information: • Overlays to be invoked during output processing. • Location on the page where the overlays are placed. • Suppression to be activated for specified page formats. **AS/400** reads and sets this field if the FORMDEF mane for the spooled file on the AS/400 begins with 'F1'.
001E	-	-	Reserved	Reserved for IBM's use.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
001F	1	1-6 1-8 for AS/400	PAGEDEF	Specified to the 1-6 character (A-Z, 0-9, #, @, or $) member name of the installation's page definition library that contains information which the page-mode printer uses to print a data set. The first two characters of the member name are pre-defined by IBM standards and prefixed to the name specified here. The members can contain the following information: • Logical page size and width. • Desired fonts and page segments. • Definition of multiple page types or formats. • Definition of lines within a page (for example, line origin, carriage controls, spacing, fonts, and suppression control). • Definition of multiple logical pages on a physical page and line-skipping instructions. **AS/400** reads and sets this field if the PAGEDEF name for the spooled file on the AS/400 begins with 'P1'.
0020	-	-	Reserved	Reserved for IBM's use.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0021	2	1-3	PIMSG	Mixed data format. Level of printing system-generated messages by a functional subsystem. See "PIMSG" on page 114 for details. (Not supported by AS/400)
0022	1	4	THRESHLD	Binary: range 1-9999999. The maximum size for the SYSOUT data set before a new unit of work is created on a data set boundary. The size is based on the number of records multiplied by the number of copies. (Not supported by JES2, VM, or VSE.)
0023	1	1	DPAGELBL	Binary: range X'40' - X'80' Specifies whether or not data pages are labeled with the security label associated with the data set. • X'40' - (YES) Specifies that security labels are to be printed. • X'80' - (NO) Specifies that security labels are not to be printed (the application may generate its own security labels). (Not supported by VM or AS/400.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0024	1	1	SYSAREA	Binary: range X'40' - X'80' Specifies whether a System Defined User Printable Area (UPA) is to be in effect when printing output. The System Defined UPA is a subset of the physical page and only Security Overlays (used for B1 security labeling) can print outside of it. • X'40' - (YES) Specifies that the System Defined UPA is in effect for printed output. • X'80' - (NO) Specifies that the System Defined UPA is not in effect for printed output.
0025	-	-	Reserved	Reserved for IBM's use.
0026	1	0-60	ROOM	EBCDIC text characters. Specifies the room identification to be associated with the SYSOUT for use by the installation. (Not supported by AS/400.)
0027	4	0-60	ADDRESS	EBCDIC text characters. Specifies the delivery address for SYSOUT. (Not supported by AS/400.)
0028	1	0-60	BUILDING	EBCDIC text characters. Specifies the building location associated with the SYSOUT. (Not supported by AS/400.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0029	1	0-60	DEPT	EBCDIC text characters. Specifies the department associated with the SYSOUT for use by the installation to assist in distributing SYSOUT. (Not supported by VM, VSE, or AS/400.)
002A	1	0-60	TITLE	EBCDIC text characters. Specifies a description of the SYSOUT that can be placed on separator pages by the installation. (Not supported by VM, VSE, or AS/400.)
002B	1	1	OUTDISP	**JES2** sets and uses this field. The second byte is not relevant once SYSOUT is created and transmitted. You can set the bytes to the following values: • X'80' - WRITE - Output is processed and purged. • X'40' - HOLD - Hold until released; goes to WRITE. • X'20' - KEEP - Output is processed and held. • X'10' - LEAVE - Hold until released; goes to KEEP. • X'08' - PURGE - Output deleted without processing.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
002C	1	1	AOUTDISP	This option specifies the abnormal output disposition. You can set the bytes to the following values: • X'80' - WRITE - Output is processed and purged. • X'40' - HOLD - Hold until released; goes to WRITE. • X'20' - KEEP - Output is processed and held. • X'10' - LEAVE - Hold until released; goes to KEEP. • X'08' - PURGE - Output deleted without processing.
002D	1	0-60	NAME	EBCDIC text characters. Specifies the preferred name to be associated with the SYSOUT for use by the installation. (Not supported by AS/400.)
002E	1	0-60	USERLIB	EBCDIC text characters. Specifies the libraries containing the Advanced Function Printer (AFP) resources that PSF uses when processing data sets. (Not supported by VM or AS/400.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
002F	1	0-60	NOTIFY	EBCDIC text characters. Specifies the user(s) to be identified by PSF when the output has finished printing. (Not supported by VM, VSE, or AS/400.)
0030	-	2	PAGERANGE	Specifies the range of pages to print this file. Odd numbered pages are always starting pages. If the parameter length is zero, the starting page is the first page of the document. An '-1' tells the system to use the ending page value in the next parameter as the starting page. Even numbered pages are always ending pages. If the parameter length is zero, the ending page is the last page of the document. No special values exist for the ending page. (Only supported by AS/400.)
0031	16	1-60	USERDATA	Specifies the installation specific feature for which it has been defined. If you specify two or more values, they must be enclosed by parentheses. If your installation does not define a use for this keyword, the information is syntax checked, stored as part of the output descriptor's information, and ignored.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
0032 to 200A	-	-	Reserved	Reserved for IBM's use.
200B	2	1-41	FILEID	This option specifies where the file to be printed or associated resource files can be found. A print file can be either line or page mode records. Page mode data sets must have each record preceded by X'5A'. See "FILEID" on page 112 for syntax details. **AS/400** only reads this field. (Not supported by MVS.)
200C 2019	-	-	Reserved	Reserved for IBM's use.
201A	1	1	BIN	Binary: range 1-2. Specifies the media bin from which sheets are fed for printing as follows: • **1** - indicates that all sheets are to be fed from the primary input source. • **2** - indicates that all sheets are fed from an alternate input source. Overrides the value in the active FORMDEF. **AS/400** reads and sets this field. (Not supported by MVS or VSE.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
201B	1	1	DUPLEX	Binary: range X'00'-X'02' Specifies whether printing is to be done in simplex or duplex mode. This parameter overrides the value in the active FORMDEF and applies during the printing of the entire print data set. • X'00' - (NODUPLEX) Specifies standard simplex printing; printing will occur on only one side of the physical page. • X'01' - (DUPLEX) Specifies that printing is to occur on both sides of the physical page. This permits the print data set to be bound on its side. • X'02' - (DUPLEX TUMBLE) Specifies that printing will occur on both sides of the physical page such that the top of the back side is oriented with the bottom of the front side. Tumble duplexing permits the print data set to be bound at the top. **AS/400** sets and reads this field. (Not supported by MVS or VSE.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
201C	1	1	OPTIMIZE	Binary: range X'00'-X'02' Specifies whether resource (page segments, fonts, etc.) loading is to be optimized to determine the best time to load resources into the printer. This will improve the performance of the loading and printer processes at the cost of more storage and the CPU used during the optimization analysis. • X'00' - (NOOPTIMIZE) Indicates that resources are to be loaded into the printer on a page-by-page basis. • X'01' - (OPTIMIZE) Specifies that all possible resources are to be loaded prior to the printing of the data set. • X'02' = (OPTIMIZE FONTS) Specifies, that in addition to resource optimization, only those characters of a font that are referenced in the data set are loaded in the printer. This option is not defined for double-byte fonts. The optimization function is not defined for line-mode or mixed-mode data sets. (Not supported by MVS, VSE, or AS/400.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
201D	1	1	VERIFY	Binary: range X'00'-X'01' Specifies that the data stream contained in the data set be syntax checked at the destination. **0=NO, 1=YES**. For example, each AFPDS structured fields are checked for fields containing values within ranges, reserved fields not used, mandatory values being correct, etc. (Not supported by MVS, VSE, or AS/400.)
201E	4		TRACE	Mixed Format. Allows for the tracing of the internal operation of the print server as a particular spool file is being processed. See "TRACE" on page 114 for additional details. (Not supported by MVS, VSE, or AS/400.)

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
201F	32	2	SHEETS	Binary: range 1-32767 This option overrides a FORMDEF specification and will cause only the sheets specified to be printed. A sheet represents the physical page, not the logical page. The numbering begins with the first sheet (sheet 1) and may or may not have a relationship with the page number printed on the pages. The sheet numbers are specified in begin/end pairs. Each number and each range must be listed in ascending order from left to right. (Not supported by MVS, VSE, or AS/400.)
2020	3	1-8	DUMP	Mixed format. Allows for the DUMPing of the internal operation of the printing subsystem (e.g., Print Services Facility™). See "DUMP" on page 112 for additional details. (Not supported by MVS, VSE, or AS/400.)
2021	-	-	Reserved	Reserved for IBM's use.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
2022	1	1	DATACK	Binary: range X'00' - X'82' This option specifies whether the printer will block print-positioning and invalid-character errors. The term "block" means the printer will not return error messages to PSF, even if data is lost. • X'00' - (BLOCK) Specifies that the printer will block (not report) print-positioning and invalid-character errors. • X'80' - (UNBLOCK) Specifies that the printer will not block (will report) both print-positioning and invalid-character errors. • X'81' - (BLKCHAR) Specifies that the printer will block invalid-character errors. The printer will report print-positioning errors. • X'82' - (BLKPOS) Specifies that the printer will block print-positioning errors. The printer will report invalid-character errors. AS/400 sets and reads this field. (Not supported by VSE.)
2023 to 3EFF	-	-	Reserved	Reserved for IBM's use.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
3F00	1	1-128	CICSDATA	Character: alphameric and national. A sequence of from 1 to 128 bytes passed by the CICS® application that created the output request. This data is then associated with the output data set within the spool subsystems. Mapped by CICS. (Not supported by MVS, VM, or AS/400.)
3F01 to 3FFF	-	-	Reserved	Not used in NJE.
4000	1	10	AS4NAM	1-10 EBCDIC text characters. User's name for a file. Reserved - private AS/400.
4001	1	2	AS4DLN	Binary: 4079|512 Spool database record length. Reserved - private AS/400.
4002	1	6	AS4DR	1-6 EBCDIC text characters. Special device requirements. Reserved - private AS/400.
4003	1	2	AS4MFW	Binary Maximum printer file forms width. Reserved - private AS/400.
4004	1	2	AS4LN1	Binary Output record length 1. Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4005	1	2	AS4SEP	Binary: 0-9 Number of file separators. Reserved - private AS/400.
4006	1	4	AS4MRC	Binary Maximum number of records allowed. Reserved - private AS/400.
4007	1	20	AS4FIL	1-20 EBCDIC text characters. Qualified print file name. Reserved - private AS/400.
4008	1	6	AS4BIT	1-6 EBCDIC text characters. OS/400® Network flags. Reserved - private AS/400.
4009	1	15	AS4ACC	1-15 EBCDIC text characters. Job accounting code. Reserved - private AS/400.
400A	1	8	AS4TKN	1-8 EBCDIC text characters. Token for graphics file. Reserved - private AS/400.
400B	1	30	AS4SEC	1-30 EBCDIC text characters. Security classification text. Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
400C	1	3	AS4DIW	4.2 packed numeric. Width of drawer 1. Reserved - private AS/400.
400D	1	3	AS4DW2	4.2 packed numeric. Height of drawer 1. Reserved - private AS/400.
400E	1	3	AS4D2W	4.2 packed numeric. Width of drawer 2. Reserved - private AS/400.
400F	1	3	AS4D2H	4.2 packed numeric. Height of drawer 2. Reserved - private AS/400.
4010	1	8	AS436T	1-8 EBCDIC text character. Time of Day (TOD) file is opened. Reserved - private AS/400.
4011	1	10	AS4UDT	1-10 EBCDIC text character. User-supplied data. Reserved - private AS/400.
4012	1	12	AS4DOC	1-12 EBCDIC text character. Document name. Reserved - private AS/400.
4013	1	64	AS4FDR	1-64 EBCDIC text character. Folder name. Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4014	1	20	AS4PGM	1-20 EBCDIC text character Qualified program name. Reserved - private AS/400.
4015	1	8	AS4PRC	1-8 EBCDIC text character S/36 procedure name. Reserved - private AS/400.
4016	1	2	AS4AFW	Binary Alternate forms width Reserved - private AS/400.
4017	1	2	AS4AFL	Binary Alternate forms length Reserved - private AS/400.
4018	1	2	AS4AL1	Binary Alternate lines per inch in 1440's of an inch Reserved - private AS/400.
4019	1	3	AS4DEL	1-3 EBCDIC text characters of YINIAIC • A=absolute fidelity • C=content fidelity Print Spool Facility (PSF) defined. Reserved - private AS/400.
401A	1	2	AS4PR1	Binary Primary record length Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
401B	1	2	AS4PFL	Binary Forms length in lines Reserved - private AS/400.
401C	1	2	AS4PFW	Binary Forms width in characters Reserved - private AS/400.
401D	1	2	AS4PLI	Binary • '0003'X - 3.0 LPI • '0004'X - 4.0 LPI • '0006'X - 6.0 LPI • '004B'X - 7.5 LPI • '0008'X - 8.0 LPI • '0009'X - 9.0 LPI Lines per inch Reserved - private AS/400.
401E	1	2	AS4POF	Binary 1-255 Overflow line number. Reserved - private AS/400.
401F	1	1	AS4PAL	Binary • '00'X - Do not Align Form • '80'X - Align Form Aligning Form Reserved - private AS/400.
4020	1	1	AS4PFT	Binary • '00'X - Truncate • '80'X - Fold Fold or Truncate Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4021	1	2	AS4PUP	1-2 EBCDIC characters Unprintable character substitution Reserved - private AS/400.
4022	1	2	AS4CPI	Binary • **'0005'X** - 5.0 CPI • **'000A'X** - 10.0 CPI • **'000C'X** - 12.0 CPI • **'000F'X** - 15.0 CPI • **'0085'X** - 13.3 CPI • **'00A7'X** - 16.7 CPI • **'0012'X** - 18.0 CPI • **'0014'X** - 20.0 CPI Characters per inch Reserved - private AS/400.
4023	1	13	AS4FT	1132 EBCDIC characters Font identifier **Note:** If specified, the valid range for the number of parameters is 0-1. If zero is specified, use *DEVD for the font and point size. Reserved - private AS/400.
4024	1	1	AS4QLT	1 EBCDIC character • **'00'X** - Standard • **'01'X** - Draft • **'02'X** - Near letter quality • **'03'X** - Use device default Print quality value Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4025	1	1	AS4FFA	1 EBCDIC character • **'00'X** - Use divide default • **'01'X** - Use continuous forms • **'02'X** - Use manual cut sheet mode • **'03'X** - Use automatic cut sheet mode Form feed attachment Reserved - private AS/400.
4026	1	2	AS4DWR	Binary • **'0001'X** - Use drawer 1 • **'0002'X** - Use drawer 2 • **'0003'X** - Use enveloper drawer 1 • **'0013'X** - Use drawer 3 • **'00FF'X** - Use value in the FORMDF Drawer value Reserved - private AS/400.
4027	1	2	AS4TXT	1-2 EBCDIC characters Text utility flags Reserved - private AS/400.
4028	1	1	AS4DTA	1 EBCDIC character File contains DBCS data Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4029	1	1	AS4EXN	1 binary number • '00'X - Do not use DBCS extension characters • '80'X - Use DBCS extension characters Use of DBCS extension characters Reserved - private AS/400.
402A	1	1	AS4KCP	1 EBCDIC character • '00'X - Print at half of CPI. • '01'X - 6.7 DBCS CPI. • '05'X - 5.0 DBCS CPI. • '06'X - 6.0 DBCS CPI. • '0A'X - 10.0 DBCS CPI. DBCS characters per inch Reserved - private AS/400.
402B	1	1	AS4RRT	1 binary number • '00'X - Do not rotate DBCS characters • '80'X - Rotate DBCS characters Rotate DBCS characters Reserved - private AS/400.
402C	1	4	AS4CHI	1 EBCDIC text character Character set and code page ID Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
402D	1	2	AS4PRG	2 binary numbers • **'0000'X** - Printer default • **'FFFF'X** - Perform computer output reduction • **'FFFE'X** - Automatic computer output reduction • **'0168'X** - Rotate 0 degrees • **'005A'X** - Rotate 90 degrees • **'00B4'X** - Rotate 180 degrees • **'010E'X** - Rotate 270 degrees Page rotate Reserved - private AS/400.
402E	1	2	AS4JFY	2 binary numbers • **'0000'X** - No justification • **'0032'X** - 50% justification • **'0064'X** - 100% justification Justification Reserved - private AS/400.
402F	48	16	AS4FE	1-16 EBCDIC text characters Font equivalence table Reserved - private AS/400.
4030	1	20	AS4FDN	1-20 EBCDIC text characters Qualified form definition name Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4031	1	9	AS4RST	1-9 EBCDIC text character • **0** - Use the starting page in the page range. • **-1** - Use the ending page in the page range. • **-3** - Use the next page after the last page printed. Restart Reserved - private AS/400.
4032	1	2	AS4MUL	2 binary numbers • **0** - Print 1 page per side. • **2** - Print 2 pages per side. • **4** - Print 4 pages per side. Multiup Reserved - private AS/400.
4033	1	1	AS4UOM	1 EBCDIC text character • **'00'X** - Inches • **'00'X** - Centimeters Unit of measure Reserved - private AS/400.
4034	1	28	AS4FOV	1-28 EBCDIC text characters Font side overlay Reserved - private AS/400.
4035	1	28	AS4BOV	1-28 EBCDIC text characters Back side overlay Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4036	1	20	AS4PGD	1-20 EBCDIC text characters Qualified page definition name Reserved - private AS/400.
4037	1	13	AS4CH	1-13 EBCDIC text characters Channel values Reserved - private AS/400.
4038	1	9	AS4PSZ	1-9 EBCDIC text characters Page size Reserved - private AS/400.
4039	1	8	AS4FMG	1-8 EBCDIC text characters Front margin Reserved - private AS/400.
4040	1	8	AS4BMG	1-8 EBCDIC text characters Back margin Reserved - private AS/400.
4041	1	40	AS4FCI	1-40 EBCDIC text characters Font character set Reserved - private AS/400.
4042	1	20	AS4CDF	1-20 EBCDIC text characters Coded font Reserved - private AS/400.

Table 22. OPTU Key Definition Table (continued)

Key	Maximum Count	Length Range	External Keyword	Data Values Allowed and Description
4043	1	20	AS4DCF	1-20 EBCDIC text characters DBCS (Double-Byte Character Set) coded font Reserved - private AS/400.
4044-4FFF	-	-	RESERVED	Reserved for AS/400.
8005	1	0-124	IPADDR	IP address extracted from the DEST = 'IP:ip-address' format of the DEST = keyword. It is supported by JES2 and JES3.

Note:

1. Except where otherwise noted, the OPTU Keys are supported on VM, and MVS systems.
2. It is acceptable for implementations to ignore keys for which no support exists; however, it is not acceptable to flag as an error any key defined herein.
3. Alphameric characters are A-Z, a-z, and 0-9.
4. VSE does not use any OPTU keys. Any program may define its own OPTU keys using VSE/POWER. VSE/POWER passes any OPTU keys to the local program or to any node via the network.

Detailed OPTU Key Specifications

The tables in this section use captions that are explained by the following:

Offset
> Offset into OPTU

Length
> OPTU sub-field length in bytes

Type Value for parameter
- 'N' = binary number
- 'C' = characters
- 'B' = Bit
- 'X' = Hexadecimal value

Values
> Parameter values

Key OPTU identifier

Count Number of parameters

Parm Length
        Length of parameter in bytes

DUMP

Allows for the DUMPing of the internal operation of the print server.

Table 23. Print Server Dumping

Parameter	Offset	Length	Type	Values
KEY	0	2	N	X'2020'
COUNT	2	2	N	3
PARM LENGTH	4	2	N	0
REASON CODE	-	-	-	-
PARM LENGTH	6	2	N	7
MESSAGE ID	8	7	C	MMMxxxa
PARM LENGTH	15	2	N	1
MESSAGE COUNT	17	1	N	0-255

REASON CODE
> A 4 byte hex reason code. This subparameter is currently not supported for NJE and the length for the subparameter is always set to zero indicating that the parameter is not specified.

MESSAGE ID
> Identifies a particular message generated by the printing subsystem.
>
> In VM systems, a PSF message identifier is represented by a 10-character field with the form CCCMMMxxxa. Message id is specified as MMMxxxa, dropping the three letter PSF component id, where 'MMM' is the PSF module id, 'xxx' is the PSF message number and 'a' is the action code of the message. Messages will be monitored for the issuance of the specified message id, and PSF will terminate the print data set with a DUMP if it is encountered.

MESSAGE COUNT
> A binary count between 0 and 255 which identifies the occurrence of the message. If message count is specified as 12, then PSF will terminate the data set with a DUMP on the 12th occurrence of the specified message. This parameter is only applied to monitoring for messages. A message count of zero with no reason code indicates no dump.

FILEID

Specifies VM file identifiers for various files used during output processing.

Table 24. File identifiers used during output processing

Parameter	Offset	Length	Type	Values
KEY	0	2	N	X'200B'
COUNT	2	2	N	2
PARM LENGTH	4	2	N	0
FILECODE	6	1	N	1-11
PARM LENGTH	7	2	N	40
FILEID	9	36		(as follows)

Table 24. File identifiers used during output processing (continued)

Parameter	Offset	Length	Type	Values
FILENAME		8	C	CMS file name
FILETYPE		8	C	CMS file type
SEND CODE		1	N	0-2
USERID		8	C	CMS userid
DISK		3	C	000-FFF
PASSWORD		8	C	
EDISK		4	C	0000-FFFF

FILE CODE
- 01 = VM FORMDEF file
- 02 = VM PAGEDEF file
- 03 = VM CODED FONT file
- 04 = VM OVERLAY file
- 05 = VM PAGESEG file
- 06 = VM PRINT file, with CC (no TRC)
- 07 = VM CODE PAGE file
- 08 = VM FONT CHARACTER SET file
- 09 = VM PRINT file, with CC and TRC
- 10 = VM PRINT file (no CC, no TRC)
- 11 = VM PRINT file, with TRC (no CC)

Note: AS/400 only reads codes 01, 02, 09, and 11.

SEND CODE

0 = SYSDISK
   Read the specified file from the SFCM resource disk. When SYSDISK or SEND are specified, the userid, disk address and password do not need to be specified.

1 = SEND
   Send the file to the SPOOL.

2 = LINK
   File will be read by the SFCM from the specified disk. When LINK is specified, the userid, disk address and password must also be specified.

USERID
   CMS userid that owns the file.

DISK
   Disk address of the minidisk which contains the file.

PASSWORD
   Password to the minidisk.

EDISK
   Extended disk address of the minidisk which contains the file. (Optional.)

PIMSG

Level of printing system-generated messages by a functional subsystem.

Table 25. Functional Subsystem Messages

Parameter	Offset	Length	Type	Values
KEY	0	2	N	X'0021'
COUNT	2	2	N	1-2
PARM LENGTH	4	2	N	2
PIMSG/ MESSAGES	6	1	X	X'80', X'40'
PARM LENGTH	7	2	N	2
MESSAGE COUNT	9	2	N	0-999

PIMSG/MESSAGES

X'40'
> specifies that the system is to print the messages generated by a functional subsystem. This is the default specification.

X'80'
> specifies that the system is not to print the messages generated by a functional subsystem.

MESSAGE COUNT
> Maximum number of messages that are to be printed. A zero value means that all messages should be printed.

TRACE

Allows for the tracing of the internal operation of the SFCM and PDM virtual machines as a particular spool file is being processed. More than one of the parameters: INPUT, OUTPUT, CALL, IOCMD, STATE, etc. may be specified. Use of this option is restricted to those users identified to the printing subsystem as privileged. Specification by non-privileged users will result in a warning message issued with the printed document.

Table 26. Trace IDs

Parameter	Offset	Length	Type	Values
KEY	0	2	N	X'201E'
COUNT	2	2	N	4
PARM LENGTH	4	2	N	1
TRACE CODE	6	1	X	X'80', X'40'
PARM LENGTH	7	2	N	7
MESSAGE ID	9	7	C	MMMxxxa
PARM LENGTH	16	2	N	1
MESSAGE COUNT	18	1	N	1-255
PARM LENGTH	19	2	N	3
STATE	21	2	B	(see the following page)

TRACE CODE

X'80' = NO
   Specifies no tracing should be done.

X''0' = YES
   Specifies tracing should be done

MESSAGE ID
Identifies a particular PSF message identifier as documented in the PSF messages. A PSF message identifier is represented by a 10-character field with the form 'CCCMMMxxxa'. Message is specified as MMMxxxa, dropping the three letter PSF component id, where 'MMM' is the PSF module id, 'xxx' is the PSF message number and 'a' is the action code from the message. Messages will be monitored for the issuance of the specified message id and tracing started with the specified options if it is encountered. If specified as X'FFFFFFFF', tracing is done for the entire SFCM and PDM operation.

MESSAGE COUNT
A 1 to 255 binary count which identifies the occurrence of the message. If message count is specified as 12, PSF will start tracing on the 12th occurrence of the specified message. This parameter is only applied to monitoring for messages.

STATE
(bits numbered left to right)

Bit 0=1 (INTOKEN) - Each CPDS structured field will be recorded to the TRACE output file before being processed (SFCM parameter).

Bit 1=1 (INREAD) - Each record of CPDS or line data read from disk, spool, or created by the SPM will be recorded. In the PDM, each record of CCWs or IPDS read from disk will be returned (SFCM, PDM).

Bit 2=1 (OUTTOKEN) - Each CCW or IPDS control created will be recorded (SFCM).

Bit 3=1 (OUTWRITE) - Each record of IPDS or CCWs created will be recorded (SFCM).

Bit 4=1 (CALL) - The name of the caller and the return address will be recorded for each module invocation made (SFCM,PDM).

Bit 5=1 (IOCMD) - Each CCW or IPDS command will be recorded before being issued to the printer (PDM).

Bit 6=1 (STATE) - Records each state change in the SFC or SPM state machines (SFCM).

Bit 7=1 (ALLOCATE) - Records each new block of storage allocated and its allocation pointers (SFCM).

Bit 8=1 (FREE) - Records each freed block of storage and its allocation pointers (SFCM).

Bit 9=1 (PAGE) - Records each page record read or written (SFCM, PDM).

Bit 10=1 (FONTTAB) - Records contents of internal font table whenever changed. Done before each page is printed. Also, records device font table information (PDM).

Bit 11=1 (OVLYTAB) - Records contents of internal overlay table whenever changed. Done before each page is printed (PDM).

Bit 12=1 (PSEGTAB) - Records contents of internal segment table whenever changed. Done before each page is printed (PDM).

Bit 13=1 (CHKPTTAB) - Records contents of internal checkpoint table whenever changed. Done after each page is printed, at the beginning and end of the document, and in error recovery situations. Also records device page number counter information (PDM).

Bit 14=1 (LIST) - May be specified along with the INTOKEN, INREAD, OUTTOKEN, OUTWRITE, ALLOCATE, FREE, PAGE, and/or IOCMD parameters. When specified with the ALLOCATE or FREE parameters, LIST will cause all blocks on the allocated or free lists, respectively, and their storage allocation pointers to be recorded to the TRACE output file. When specified with the INTOKEN, INREAD, OUTTOKEN, OUTWRITE, PAGE, or IOCMD parameters, LIST will record the entire input record.

Bit 15=1 (PPCC) - After each SNA LU 6.2 communications verb issued to the PPCC, the PPCC internal trace of module calls and returns for that PPCC verb will be recorded. These modules are not traced by the "CALL" keyword because PPCC is a separate component (3820 PDM).

Bit 16=1 (EMSGSNAP) - Records a single snap of the 3820 internal trace table, the SNA driver I/O control block and the 3820 global work area when message type E or S occurs (3820 PDM).

Bit 17=1 (3820TBLS) - Records the contents of the data set information table and the data set tracker table whenever they are modified (3820 PDM).

Bits 18-23 are reserved for future use (set to 0).

Error Handling

Destination nodes that do not recognize OPTUs received in an Output Processing Section may either reject the SYSOUT (X'B0' RCB) or accept the SYSOUT and subsequently ignore the data, print it (with unpredictable results), or hold the SYSOUT. A message describing the problem must be sent to the NJHNTYN node and NJHNUSID in the job header. Destination nodes that process OPTUs must adhere to the following rules for error cases:

Invalid text block format
A system-dependent action may be taken, such as rejecting the file or substituting an entire set of default parameters. A message must be sent to either the destination or originating node.

Keyword parameter values out of range
Specified defaults must be used unless overridden by the installation. No message is required.

Invalid keyword identifier
Invalid keywords must not be processed. They may be ignored, or the file may be rejected. A message should be issued at the destination node, but is not required unless the file is rejected.

Note: Not all products support error handling.

Installation-Defined Section

This section is identified by an identifier field of B'11xxxxxx' and a modifier field of the installation's choice. This section is not used by any IBM products, but will be passed through the network.

Beyond the four-byte definition, this section is the responsibility of the installation. The section length is limited to 32764 bytes. The combined length of all sections in the header is limited to 32764 bytes, although product implementations may be more restrictive. The length must be reflected in the NDHULEN field in the front of the control record section header. The following depicts a sample eight-byte field:

Table 27. Fields in the Data Set Header Extension Section

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NDHULEN	0	2	binary	none	4-32764	Length of the section.
NDHUTYPE	2	1	binary	none	C0-FF	The major type of this section.
NDHUMOD	3	1	binary	none	none	Modifier for the major type identifier.
NDHUCODE	4	4	char	none	none	This could be used for a GUIDE/SHARE installation code, or some other unique identifier.

Chapter 5. Job Trailer

A job trailer indicates the end of an NJE job. The job trailer is divided into several sections depending on the networking facility where the NJE job originated. All networking facilities require a job trailer prefix section and a job trailer general section.

Table 28 describes which sections in the trailer are built and used by each subsystem.

Table 28. Job Trailer Sections used by Each Subsystem

Section	JES2	JES3	RSCS	POWER	AS/400
"Job Trailer General Section"	built and used	built and used	built and used	built and used	built and used
"Accounting Section"	built and used	built and used	built and used	not built or used	not built or used
Product Specific Sections	none	none	none	none	none

Job Trailer Prefix Section

The job trailer begins with the standard four-byte control record prefix containing the combined length of all sections. Table 29 describes the format of the job trailer prefix section.

Table 29. Fields in the Job Trailer Prefix Section

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTLEN	0	2	binary	none	8-32K	Length of all sections in the job trailer.
NJTFLAGS	2	1	binary	0	none	Reserved for IBM's use.
NJTSEQ	3	1	binary	0	none	The sequence indicator, which contains valid information after the header is segmented for transmission. The information specifies: • The position of the segment within the record. • More record segments follow.
		.1	80 bit	0		High order bit means that more segments follow.
		.7	7F bits	0		Low order 7 bits contain the segment number starting with zero (0).

Job trailer general section

The job trailer general section is identified by an identifier field of X'00' and a modifier field of X'00'. Table 30 describes the format of the job trailer general section.

Table 30. Fields in the job trailer general section

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTGLEN	0	2	binary	none	4-32764	Length of the job header general section.
NJTGTYPE	2	1	binary	none	none	The type of section. This value must be X'00' for the job trailer general section.
NJTGMOD	3	1	binary	0	none	Modifier for the type of section. This field contains X'00' to identify this as the job trailer general section.
NJTGFLG1	4	1	bits	0	none	Reserved for IBM's use.
NJTGXCLS	5	1	char	A	A-Z 0-9	The actual execution class of the job. **AS/400 and JES3** set this field to zero and do not use it. **RSCS and POWER** do not use the execution class.
	6	2				Reserved for IBM's use.
NJTGSTRT	8	8	binary	0	none	The time of day the job started execution in System/370 TOD clock format (GMT). **JES2** sets the time of day and uses this field in SMF type 26 records. **AS/400, JES3, RSCS, and POWER** set this field to zero and do not use it.
NJTGSTOP	10	8	binary	0	none	The time the job completed execution in System/370 TOD clock format (GMT). **JES2** sets the field to zero or the default value for spin data sets. **AS/400, JES3 and RSCS** set the field to zero and do not use it. **POWER** sets the field but does not use it. For spin output, all trailers except the last are zero.
	18	4				Reserved for IBM's use.

Table 30. Fields in the job trailer general section (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTGALIN	1C	4	binary	0	none	The total number of print lines for this job or job segment at all locations. This is not multiplied by the number of copies. **AS/400** sets this field to the number of records if it is a print file, but does not use it. **JES2** sets the field to zero or the default value if SYSOUT is sent before the job finishes executing. **JES3** sets the field to the number of lines written to spool and uses it for SMF accounting. **RSCS** sets the field to the number of records in the file (from TAGRECNM) if the file is a print file. RSCS does not use this field. **POWER** sets the field to the number of records produced by the job. For NJE jobs that contain a spin-off data set, all trailers except for the last are zero.
NJTGACRD	20	4	binary	0	none	The total number of card images produced for this job or job segment at all locations. This is not multiplied by the number of copies. **AS/400** sets this field to the number of records if it is a punch file or SYSIN , but does not use it. **JES2** sets the field to X'00' for NJE jobs that contain spin-off data sets. **JES3** sets the field to the number of punch cards written to spool and uses it for SMF accounting. **RSCS** does not use this field. RSCS sets the field to the number of records in the file (from TAGRECNM) if it is a punch file (SYSOUT) or SYSIN. **POWER** does not use this field. It is set to the number of records produced by the total job. For spin output, all trailers except the last are zero.
	24	4				Reserved for IBM's use.

Table 30. Fields in the job trailer general section (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTGIXPR	28	1	binary	0	0-15	The initial requested execution selection priority. **AS/400** does not use this field; it is set to AS/400 transmission priority as shown in NJHGPRIO. **JES2** uses this field but does not set it. The low-order 4 bits are shifted to the high-order four bits of JCTIPRIO by the SYSOUT receiver. **JES3** defaults this field but does not use it. **RSCS** does not use this field. It is set to the RSCS transmission priority as shown in NJHGPRIO. **POWER** sets this field from the PRI= parameter on the * $$ JOB statement.
NJTGAXPR	29	1	binary	0	0-15	The actual execution selection priority used. **AS/400** does not use this field; it is set to AS/400 transmission priority as shown in NJHGPRIO. **JES2** sets this field, but does not use it. **JES3** sets this field to the default, but does not use it. **RSCS** does not use this field. It is set to the RSCS transmission priority as shown in NJHGPRIO. **POWER** defaults this field, but does not use it.

Table 30. Fields in the job trailer general section (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTGIOPR	2A	1	binary	0	0-255	The initial job priority for output selection. **AS/400** does not use this field; it is set to AS/400 transmission priority as shown in NJHGPRIO. **JES2** sets the range (0-15) when the SYSOUT job is selected for transmission. (JES2 sets the field to 1 for spin data sets if the job is still executing.) JES2 does not use this field. **JES3** uses the greater value of NJTGIOPR or NJTGAOPR for the job's priority for received output (unless greater than 15, in which case 15 is used). This field is set from the job's priority for outgoing SYSOUT. **RSCS** does not use this field. It is set to the RSCS transmission priority as shown in NJHGPRIO. **POWER** defaults this field but does not use it.
NJTGAOPR	2B	1	binary	0	0-255	The actual output selection priority used. **AS/400** does not use this field; it is set to AS/400 transmission priority as shown in NJHGPRIO. **JES2** defaults this field, but does not use it. **JES3** uses the greater value of NJTGIOPR or NJTGAOPR for the job's priority for received output (unless greater than 15, in which case 15 is used). This field is set from the job's priority for outgoing SYSOUT. **RSCS** does not use this field. It is set to the RSCS transmission priority as shown in NJHGPRIO. **POWER** defaults the field but does not use it.
NJTGCOMP	2C	1	binary	0	0-255	Job completion indicator.
NJTJCAB		.1	80 bit			NJTGCODE contains an abend code.
NJTJCCC		.1	40 bit			NJTGCODE contains a completion code.
NJTGCCDE		.6	2f BIT			Specific completion type.
NJTGCUNK			000000			No completion information.
NJTGCNRM			000001			Job ended normally. NJTGCODE contains highest step cc.
NJTGCECC			000010			Job ended by cc. NJTGCODE contains terminating step cc.
NJTGCJCL			000011			Job had a JCL error.

Table 30. Fields in the job trailer general section (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTGCCAN			000100			Job was cancelled.
NJTGCABN			000101			Job ABENDed.
NJTGCCAB			000110			Converter ABENDed.
NJTGCSEC			000111			Security error.
NJTGCEOM			001000			Job ABENDed in end-of-memory processing.
NJTGCODE	2D	3	binary	0	0-FFFFFF	Completion code. If NJTJCAB is set, contains system abend code in first 12 bits and user abend code in last 12 bits. If NJTJCCC is set, contains applicable step completion code.

Accounting Section

The following defines the accounting section. This section is identified by an identifier field of X'89' and a modifier field of X'00'.

Note: These statistics are for the entire job, not just for an individual SYSOUT data set.

Table 31. Fields in the Job Trailer Accounting Section

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTS	0	0	label	none	none	The start of the accounting section.
NJTSLEN	0	2	binary	none	none	Length of the accounting section including the two bytes used to hold the length.
NJTSTYPE	2	1	binary	X'89'	none	The identifier for the accounting section.
NJTSMOD	3	1	binary	X'00'	none	The modifier for the accounting section, which must be '00'.
NJTSAPAG	4	4	binary	none	none	The actual number of the "begin page" structured fields. **JES3** sets this field to the actual number of pages and uses it in SMF accounting. **POWER** does not set or use this field.
NJTSABYT	8	4	binary	none	none	The actual number of bytes. **JES3** sets this field to the actual number of bytes and uses it in SMF accounting. **POWER** does not set or use this field.

Installation-Defined Section

This section is identified by an identifier filed of B'11xxxxxx' and a modifier field of the installation's choice. This section is not used by any IBM products, but will be passed through the network.

Note: Beyond the four-byte header, this definition is the responsibility of the installation. The section length is limited to 32764; the combined length of all sections in the header is limited to 32764, although product implementations may be more restrictive. The length must be reflected in the NJTULEN field in the front of the control record section header. The following depicts a sample eight-byte field:

Table 32. Fields in the Job Trailer User Section

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NJTULEN	0	2	binary	none	4-32764	Length of the section.
NJTUTYPE	2	1	binary	none	C0-FF	The major type of this section.
NJTUMOD	3	1	binary	none	none	Modifier for the major type identifier.
NJTUCODE	4	4	char	none	none	This could be used for a GUIDE/SHARE installation code, or some other unique identifier.

Chapter 6. NJE Nodal Message Records (NMR)

A node in a network uses a nodal message record (NMR) to transmit commands and messages. Command responses are treated as messages in the nodal message record. The RCB associated with the NMR contains X'9A'.

The NMR record consists of a header containing control information followed by the message or command text. Some of the fields in the header are used differently depending on whether the record contains a message, formatted command, or an unformatted command.

The rules for sending messages and commands in a network are different than those for sending files. NMRs are not stored on spool at an intermediate node. If the NMR cannot be sent to the next node, it is discarded and only message responses are sent back to the user at the originating node. (The originating node may not receive any information about an NMR if disastrous errors occur at any of the network nodes during its transmission.)

When an NMR can be sent to a node, it is sent directly. After a node processes a command, the node generates and sends a message to the originating node to inform the person issuing the command that it has been processed.

Identifying the Contents of an NMR

To understand the contents of an NMR, you must first examine the contents of the NMRFLAGC and NMRTYPEF flags. The NMRFLAGC flag contains valid information only if the NMRTYPEC flag contains a 1. Use Table 33 to determine the contents of an NMR.

Table 33. Identifying the Contents of an NMR

NMRFLAGC	NMRTYPEF	NMR Contents
1	0	Unformatted command
1	1	Formatted command
0	N/A	Message

Interpreting the contents of the NMROUT field

The contents of the NMROUT field at offset X'0D' are determined by examining the following bits in NMRFLAG at offset X'00':

- NMRFLAGW -- 2 bits into NMRFLAG
- NMRFLAGT -- 3 bits into NMRFLAG
- NMRFLAGU -- 4 bits into NMRFLAG

Use the following table to determine the contents of NMROUT:

Table 34. Interpreting the NMROUT Field

NMRFLAGW	NMRFLAGT	NMRFLAGU	NMROUT Contents
0	0	0	MCS routing codes for the message.

Table 34. Interpreting the NMROUT Field (continued)

NMRFLAGW	NMRFLAGT	NMRFLAGU	NMROUT Contents
0	0	1	The console identifier.
0	1	0	Userid.
1	0	0	The remote workstation identifier.
1	1	0	Userid/remote workstation identifier. (NMRout type)

If NMRFLAGW and NMRFLAGT are both set, the origin node could not determine whether the value of NMROUT is an interactive userid or a remote workstation identifier. In these cases, the destination node must determine the source so they can transmit to the proper location.

JES3 checks the Console Status Table (CST) for a match with a remote workstation identifier. If one is not found, the source is presumed to be a TSO/E userid.

JES3 issues an IATXCNDB macro to determine if a remote console with the given name exists. If a console with the specified name does not exist, the source is presumed to be a TSO/E userid.

Format of an NMR that contains an unformatted command

Unformatted commands are sent in NMRs and can be distinguished from other NMRs because the NMRFLAGC flag is set on and the NMRTYPEF flag is set off.

Table 35. Fields in the unformatted command NMR

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NMRFLAG	0	1	byte	0	none	Flag byte
bit NMRFLAGC		.1	80 bit	1	none	If on, the NMR contains a command.
bit NMRFLAGW		.1	40 bit	0	none	If on, the NMROUT field contains the remote number. **AS/400** does not use the bit. **JES3** sets and does not use the bit. **RSCS** uses but does not set the bit. **POWER** does not use or set the bit.

Table 35. Fields in the unformatted command NMR (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
bit NMRFLAGT		.1	20 bit	0	none	If on, the NMROUT field contains the originating userid. **AS/400** uses this bit. **JES3 and POWER** do not use or set the bit. **RSCS** does not use this bit.
NMRFLAGU		.1	10 bit	0	none	If on, NMROUT contains the originating console identifier. **AS/400** does not use this bit. **POWER** does not use or set this bit. **JES3** sets the bit it if is transmitting an NMR that contains a command. It is not used when JES3 receives an NMR that contains a command. **RSCS** does not use the bit.
bit NMRFLAGR		.1	08 bit	0	none	If one, the console is remote-authorized only. **AS/400** does not use this bit. **JES3** sets but does not use the bit. **POWER and RSCS** do not use or set this bit.
bit NMRFLAGJ		.1	04 bit	0	none	If on, the console is not job-authorized.
bit NMRFLAGD		.1	02 bit	0	none	If on, the console is not device-authorized..
bit NMRFLAGS		.1	01 bit	0	none	If on, the console is not system-authorized.
bit NMRTYPEX		.4	FO bits	0	none	Reserved for IBM's use.
bit NMRTYPE5		.1	10 bit	0	none	If on, the sending subsystem is included in NMRMSG.
bit NMRTYPE4		.1	08 bit	0	0	Used to indicate NJE security signature. See /SIGTP/.
bit NMRTYPET		.1	04 bit	0	0	Used to indicate NJE security signature. See/SIGTP/.
bit NMRTYPEF		.1	02 bit	0	0	Always 0 for unformatted commands.
bit NMRTYPED		.1	01 bit	0	none	Reserved for IBM's use.
NMRML	3	1	binary	1	1-132	Length of the information in NMRMSG.
NMRTO	4	9	mixed	none	none	Describes the destination node for the command with the following two fields.
NMRTONOD		8	char	blanks	none	The destination node for the command.

Table 35. Fields in the unformatted command NMR (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NMRTOQUL		1	binary	0	none	The system identifier for the destination node. It is used to identify the system that should process the command. **JES3** sets but does not use the byte. **RSCS and POWER** set it to the default value but do not use the byte.
NMROUT	D	8	char	zeros	none	The originating userid, remote id, or console id depending on the setting of NMRFLAG. **AS/400** uses this field. **JES3** sets the originating id to either the remote workstation or a JES3 console where the *SEND command was issued. **Note:** The format for consoles is the same as for messages: NMRDESC=0000, NMRROUT=JES3 console number, NMRCNID=00000000. **RSCS** initializes the field to the originating userid. It always contains the originating id when either NMRFLAGT or NMRFLAGW is on.
NMRFM	15	9	mixed	none	none	The following two fields describe the originating node.
NMRFMNOD		8	char	blanks	none	The originating node.
NMRFMQUL		1	binary	0	none	The system identifier of the originating node. **JES3** sets the system identifier to the default and uses the field. **AS/400, RSCS and POWER** set it to the default value and use this field.
NMRMSG	1E	132	mixed	none	none	The command text.
NMRLEVEL	1	4 high-order bits	binary	0	none	Networking products do not use these bits for commands. The bit should be set to X'70'.
NMRPRIO		4 low-order bits	binary	0	none	The output priority. Networking products do not use these bits for commands. The bit should contain X'07'.
NMRTYPE	2	1	byte	0	none	Flag byte

Format of an NMR that Contains a Formatted Command

Formatted commands are sent in NMRs and can be distinguished from other NMRs because flag NMRFLAGC and NMRTYPEF are both on.

JES2 sends formatted commands. AS/400, RSCS, and JES3 support formatted commands for input only, and change them to equivalent AS/400. RSCS, or JES3 commmands prior to processing. Therefore, in the following section, the AS/400, RSCS, and JES3 descriptions indicate only the usage of the commands.

POWER does not send out formatted commands. If POWER receives an NMR that contains a formatted command, the NMR is discarded and the user at the originating node is not notified that the command has been discarded.

Table 36. Fields in the Formatted Command NMR

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NMRFLAG	0	1	byte	0	none	Is a flag byte
NMRFLAGC	0	.1	80 bit	1	none	If on, the NMR contains a command.
NMRFLAGW	0	.1	40 bit	0	none	If on, the NMROUT field contains a JES2 remote number. **AS/400 and JES3** do not use this bit. **RSCS** does not use or set this bit.
bit NMRFLAGT	0	.1	20 bit	0	none	If on, the NMROUT field contains a userid. **AS/400 and JES3** do not use the bit. **RSCS** does not use the bit.
bit NMRFLAGU	0	.1	10 bit	0	none	If on, the NMROUT field contains the originating console identifier. **AS/400, JES3 and RSCS** do not use this bit.
bit NMRFLAGR		.1	08 bit	0	none	If on, the console is remote-authorized only. **AS/400, JES3 and RSCS** do not use this bit.
bit NMRFLAGJ		.1	04 bit	0	none	If on, the console is not job-authorized.
bit NMRFLAGD		.1	02 bit	0	none	If on, the console is not device-authorized.
bit NMRFLAGS		.1	01 bit	0	none	If on, the console is not system-authorized.
NMRLEVEL	1	4 high-order bits	binary	0	none none	Does not apply to NMRs that contain commands.
NMRPRIO		4 low-order bits	binary	0	none	Does not apply to NMRs that contain commands.
NMRTYPE	2	1	bits		None	
bit NMRTYPEX		.4	F0 bits	0	none	Reserved for IBM's use.
bit NMRTYPE4		.1	08 bit	0	none	Always 0 for commands.
bit NMRTYPET		.1	04 bit	0	none	Always 0 for commands.
bit NMRTYPEF		.1	02 bit	0	1	On for formatted commands.
bit NMRTYPED		.1	01 bit	0	none	Not applicable to commands.
NMRML	3	1	binary	1	1-132	Length of the information in NMRMSG. The length includes NMRFM, even if it is not specified.
NMRTO	4	9	mixed	none	none	This field is comprised of the following two fields:
NMRTONOD		8	char	blanks	none	The destination node for the command.

Table 36. Fields in the Formatted Command NMR (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NMRTOQUL		1	binary	0	none	The system identifier for the destination node. It identifies the system that should process the command. **AS/400** does not use this field. **JES3 and RSCS** do not use or set this field.
NMROUT	D	8	char	zeros	none	The originating userid, remote id, or console id depending on the setting of NMRFLAG.
NMRFM	15	9	mixed	none	none	The following two fields describe the originating node.
NMRFMNOD		8	char	blanks	none	The originating node.
NMRFMQUL		1	binary	0	none	The system identifier of the originating node.
NMRMSG	1E	variable	mixed	none	none	The formatted command.
NMRFOP		1	binary	none	none	The action the destination node should take with the NMR that contains the formatted command. The possible values are: 1. NMRFOPD - the NMR contains a command to display a job. 2. NMRFOPC - the NMR contains a command to cancel a job. 3. NMRFOPA - the NMR contains a command to release a job. 4. NMRFOPH - the NMR contains a command to place a job in operator hold. 5. NMRFOPR - the NMR contains a command to reroute the job.
NMRFFLG	1F	1	byte	0	none	Flags or opcode modifiers. **JES2** initializes the field to zeros.
bit NMRFFLGO		.1	80 bit	0	none	If for a cancel command, the output will be purged. If for a route command, output will be rerouted.
bit NMRFFLGD		.1	40 bit	0	none	This flag indicates that the node should cancel job execution and issue a dump. This field is mutually exclusive with NMRFFLGO when NMRFFLGO is on for cancel. **JES3** does not use this bit.
		.6	3F bits	0	none	Reserved for IBM's use.
NMRFJID	20	2	binary	none	none	The job identifier of the job to be processed by the NMR. **JES2** initializes the field to zeros. It is set only if specified by the $G operator command. **JES3** does not use the original job number field.
NMRFORGN	22	8	char	blanks	none	The originating node where the job was issued.

Table 36. Fields in the Formatted Command NMR (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NMRFJNAM	2A	8	char	blanks	none	The job name of the job to be processed. **JES3** uses the job name or number that the operator would use on an operator command. **RSCS** uses the first 4 characters as the spool identifier.
NMRFD	32	8	char	blanks	none	The destination for the route command.
NMRFR	3A	8	char	blanks	none	The remote name for the route command if not implied by NMRFD.
NMRFJNO	42	4	binary	none	1-4B	Expanded job number ('4B' = 4,294,967,295).

Format of an NMR that Contains a Message

Messages are sent in NMRs and can be distinguished from other NMRs because flag NMRFLAGC is off. The format of the NMR varies depending on the settings of bits in NMRFLAG and NMRTYPE. Table 38 describes the format of an NMR that contains a message.

Interpreting the Contents of Field NMRMSG

To determine the contents of the NMRMSG field at offset X'1E', you must also examine the contents of the NMRTYPE4 field at offset X'02' and the NMRTYPET field at offset X'00'. Use Table 37 to determine the contents of NMRMSG.

Table 37. Interpreting Field NMRMSG

NMRTYPE4	NMRTYPET	NMRMSG Contents
0	0	Time stamp and the text of the message which starts at NMRMSG + 8.
0	1	The text of the message, which starts at NMRMSG.
1	0	Time stamp, userid, and the text of the message, which starts at NMRMSG + X'10'.
1	1	Userid and th etext of the message, which starts at NMRMSG + 8.

Table 38. Fields in the Messages NMR

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NMRFLAG bit NMRFLAGC bit NMRFLAGW	0	1 .1 .1	bits 80 bit 40 bit	0 0 0	none 0 none	Flag byte. 0 indicates the NMR contains a message. If on, the NMROUT field contains the remote number. **AS/400** sets this bit to 0, but does not use it. **RSCS** does not use or set the bit.

Table 38. Fields in the Messages NMR (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
bit NMRFLAGT		.1	20 bit	0	none	If on, the NMROUT field contains the destination userid. **AS/400** always sets and uses this bit. **JES3 and POWER** do not use the bit.
bit NMRFLAGU		.1	10 bit	0	none	If on, the NMROUT contains the destination console identifier. **AS/400** sets this bit to 0, but does not use it. **POWER** does not use this bit. **RSCS** does not use the bit.
bit NMRFLAGR		.1	08 bit	0	none	Not used for messages.
bit NMRFLAGJ		.1	04 bit	0	none	Not used for messages.
bit NMRFLAGD		.1	02 bit	0	none	Not used for messages.
bit NMRFLAGS		.1	01 bit	0	none	Not used for messages.

NMRLEVEL	1	.4	bits	0	none	The importance level of the NMR. The bit definitions are:
						
						X'10' Non-essential messages
						X'30' Normal messages
						X'50' Messages requiring delayed
						      operator action
						X'70' Essential messages
						X'80' Messages requiring immediate
						      operator action
						X'F0' Extremely important messages
						
						AS/400, JES3 and POWER set the importance level to X'70', but do not use it.
						
						RSCS does not use the importance level. The importance level is set to X'70' for notify messages and sets it to X'30' for command responses.

NMRPRIO	1	.4	bits	0	none	The output priority. **AS/400, JES3 and POWER** set the output priority to X'07', but do not use it. **RSCS** sets the output priority to X'07'.
NMRTYPE	2	1	byte	0	none	Flag byte.
bit NMRTYPEX		.4	F0 bits	0	none	Reserved for IBM's use.
bit NMRTYPE4		.1	08 bit	0	none	If on, the sending userid is included in NMRMSG.
bit NMRTYPET		.1	04 bit	0	none	If on, the timestamp is not imbedded in NMRMSG.
bit NMRTYPEF		.1	02 bit	0	none	Not used for NMRs that contain a message.

Table 38. Fields in the Messages NMR (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
bit NMRTYPED		.1	01 bit	0	none	If on, the NMR contains a delete operator message (DOM) request. DOM is a function of Multiple Console Support (MCS). This bit is not set or used by any system.
NMRML	3	1	binary	1	1-148	Length of the information in NMRMSG, including the userid if it is present. It does not include the timestamp if it is present.
NMRTO	4	9	mixed	none	none	The following two fields describe the destination node and system identifier for the command.
NMRTONOD		8	char	blanks	none	The destination node for the message.
NMRTOQUL	C	1	binary	0	none	The qualifier for the destination node. **AS/400, RSCS and POWER** set but do not use the qualifier code.
NMROUT	D	8	char	zeros	none	The destination userid, remote id, or console id depending on the setting of NMRFLAG.
NMRUCM	D	1	binary	0	none	This field contains the MCS console identifier when NMRFLAGU is set. **AS/400, JES3, RSCS, and POWER** do not use or set this field.
NMRUCMA	E	1	binary	0	none	This field contains the MCS console area when NMRFLAGU is set. **AS/400, JES3, RSCS, and POWER** do not use or set this field.
NMRLINET	F	2	binary	0	none	The line spacer for a multi-line WTO when NMRFLAGU is set. **AS/400, JES3, RSCS, and POWER** do not use or set this field.
NMRDESC	D	2	binary	0	none	This field contains the MCS descriptor codes when NMRFLAGT and NMRFLAGU are not set. **AS/400 and JES3** does not use or set the field for messages. **RSCS** does not use the descriptor codes because it does not use MCS. **POWER** sets the descriptor codes to X'0800' if a destination userid was not specified.
NMRROUT	F	2	binary	0	none	This field contains the MCS routing codes when NMRFLAGT and NMRFLAGU are not set. **JES3** does not set the routing codes for messages. When receiving an NMR that contains a message, this field may contain a JES3 console number. **RSCS and POWER** sets the field to X'4100' if the destination userid is not specified.

Table 38. Fields in the Messages NMR (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NMRCONID	11	4	binary	0	none	The extended console identifier. **JES3** does not set or use the field.
NMRRMT	D	8	char	none	none	This field contains the remote name when NMRFLAGW is set. **JES2** uses the form 'Rnnnn' for the remote name.
NMRUSER	D	8	char	none	none	This field contains the destination userid when NMRFLAGT is set.
NMRFM	15	9	mixed	none	none	The following two fields describe the originating node and the qualifier.
NMRFMNOD		8	char	blanks	none	The origin node.
NMRFMQUL		1	binary	0	none	The qualifier for the originating node. **AS/400** sets this filed to the default value, but does not use it. **JES3** sets but does not use the qualifier. **RSCS and POWER** do not use or initialize the field.
NMRMSG	1E	1 -148	mixed	none		May contain the following: • 8 bytes that contain the timestamp when the message was issued. • 8 bytes that contain the userid of the operator or user that sent the message. • Up to 132 bytes of message text.
NMRECSID	1E	8	char	blanks	none	If NMRTYPE4 is set, this field contains the originating userid followed by the message text. Otherwise, this field does not exist. **JES3** does not use or set the field. **RSCS** sets and uses the field for messages that originate from a VM user.

Chapter 7. Networking Connection Control Records (NCCR)

The following types of NCC records are defined in NJE as specified in the NCCSRCB field:

- I - Initial signon
- J - Response signon
- K - Reset signon
- L - Concurrence signon
- M - Add connection
- N - Subtract connection
- B - Signoff

Only JES2 uses the NCC record types K, M, and N. Only JES2 and JES3 use the NCC record type L. JES3 uses the NCC record type L only for TCP/IP NJE.

Initial Signon Record

The primary NJE node sends an initial signon record to initiate a session or connection. The initial signon record is the only data in the transmission buffer and is not compressed or compacted. If a session between two nodes is already active and additional signon records are received, the subsequent signon records are ignored.

Table 39. Fields in an NJE Initial Signon Record

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCRCB	0	1	binary	X'F0'	none	All path manager RCBs contain X'F0'.
NCCSRCB	1	1	char	none	none	The type of connection control record. Initial signon records contain a C'I'.
NCCIDL	2	1	char			Length of the initial signon record from the RCB to the end of the initial signon record.
NCCINODE	4	8	char	none	none	The EBCDIC name of the primary node.
NCCIQUAL	B	1	binary	0	1-7	The member number of the node.
NCCIEVNT	C	4	binary	0		Not used for initial signon records.
NCCIREST	10	2	binary	0		Partial node-to-node resistance. **JES2** uses and sets this field. **AS/400** sets this field to X'00C8'. **RSCS** sets this field to X'0000'. **JES3 and POWER** sets this field to X'00C8'.
NCCIBFSZ	12	2	binary			The maximum size of the transmission buffer that the sending node can receive (must be greater than 300 bytes).
NCCILPAS	14	8	char	blanks		The password for the line for nodes using BSC protocols.
NCCINPAS	1C	8	char	blanks		The password for the node.

Table 39. Fields in an NJE Initial Signon Record (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCIFLG	24	1	flag	bits		This field contains X'00' for an initial signon record.
NCCIFLGS	24	X'40' bit	flag	0		Indicates that a secure signon protocol is to be used.
NCCIFEAT	25	4	binary	0		Defines space for up to 32 features. **JES2** only supports NCCIMDR (bit X'1'). **RSCS** supports at least one of these flags. **JES3, BDT, POWER, and AS/400** do not support them.
NCCIPREP		.1	80 bit	0		BSC/CTC quiesce options. **RSCS** uses and sets this flag. **JES2** does not use or set this flag.
NCCITRM		.1	40 bit	0		SNA termination options. **RSCS and JES2** do not use or set this flag.
			20 bit			Reserved for IBM's use.
NCCIPACK		.1	10 bit	0		Multiple NJE headers may be sent in a single buffer. **RSCS and JES2** use and set this flag.
NCCIRIF		.1	8 bit	0		Request to exchange records may be omitted after first object sent in a stream. **JES2** does not use or set this bit. **RSCS** uses and sets this bit.
NCCIMRCB		.1	04 bit	0		Mixed RCBs may be sent in a single buffer. **JES2** does not use or set this bit. **RSCS** uses and sets this bit.
NCCINOS		.1	02 bit	0	none	This system supports 'non-output' SYSOUT data. JES2 uses this bit.
NCCISSIN		.1	01 bit	0		This system supports spanned SYSIN data. When you use NCCISSIN, the valid range for NDHCLREC is 0-32767.
	26	3	binary	0		Reserved for IBM's use.
NCCIPRAW	14	8	hex	0		Unencrypted string 1 for NJE secure signon

Response Signon Record

A node that receives an initial signon record responds with a response signon record if a session or connection is not already established. The response signon record is the only data in the transmission buffer and is not compressed or compacted.

Table 40. Fields in an NJE Response Signon Record

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCRCB	0	1	binary	X'F0'		All path manager RCBs contain X'F0'.
NCCSRCB	1	1	char	none	none	The type of connection control record. Response signon records contain a C'J'.
NCCIDL	2	1	binary			Length of the response signon record from the RCB to the end of the extension.
NCCINODE	3	8	char	none	none	The EBCDIC name of the secondary node.
NCCIQUAL	B	1	binary	0	1-7	The member number of the node. (1 indicates the member is a JES2 node.)
NCCIEVNT	C	4	binary	0		Normal signon sequence is 0 if high end sends or secondary trunk of multiple trunk connection. Next higher sequence if low end sends and the trunk is primary. Predefined connection is X'FFFFFFFF'.
NCCIREST	10	2	binary	0		Partial node-to-node resistance.
NCCIBUFSZ	12	2	binary			The size of the largest transmission buffer the NJE secondary node can receive.
NCCILPAS	14	8	char	blanks		The password for the line.
NCCINPAS	1C	8	char	blanks		The password for the node.
NCCIPRAW	14	8	hex	0		Un-encrypted string 2 for NJE secure signon
NCCIPENC	1C	8	hex	0		Encrypted string 1 for NJE secure signon
NCCIFLG	24	1	bits	0		Flag byte.
NCCIFLGM		.1	80 bit			On for multi-trunk responses.
		.6	3F bits			Reserved for IBM's use.
NCCIFEAT	25	4	binary	0		Defines space for up to 32 new features. See Table 39 on page 137 for bit definitions.

Reset Signon Record

A reset signon record is sent to increment the CES value and/or to change the resistance. The NJE secondary node responds to the request by sending a reset signon record instead of a response signon record. The reset signon record can be included with other NJE records in the transmission buffer but is always placed at the beginning of the transmission buffer. The data in the NJE record is not compressed or compacted.

Table 41. Fields in an NJE Reset Signon Record

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCRCB	0	1	binary	X'F0'		All path manager RCBs contain X'F0'.
NCCSRCB	1	1	char	none	none	The type of connection control record. Reset signon records contain a C'K'.
NCCCDL	2	1	binary			Length of the response signon record from the RCB to the end of the concurrence signon record.

Table 41. Fields in an NJE Reset Signon Record (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCCEVNT	3	4	binary			0 if the record is sent by the high end or secondary trunk of a multi-trunk connection. (A reset record cannot be sent from a low end on a normal multi-trunk signon sequence).
NCCCPENC	12	8	hex	0		Encrypted password 2 for NJE secure signon
	7	3				Reserved for IBM's use.
NCCIREST	10	2	binary	0		Partial node-to-node resistance.

Concurrence Signon Record

A node sends a concurrence signon record when an acknowledgement is expected from a response or reset signon record. The concurrence signon record can be included with other connection records in an NJE record. The data in the NJE record is not compressed or compacted.

Table 42. Fields in a Concurrence Signon Record

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCRCB	0	1	binary	X'F0'		All path manager RCBs contain X'F0'.
NCCSRCB	1	1	char	none	none	The type of connection control record. Concurrence signon records contain C'L'.
NCCCDL	2	1	binary			Length of the response signon record from the RCB to the end of the concurrence signon record.
NCCCEVNT	3	4	binary			The contents of the CCCEVNT field from the concurrence signon record.
	7	2				Reserved for IBM's use.
NCCCPENC	12	8	hex	0		Encrypted password 2 for NJE secure signon
NCCCREST	10	2	binary			The total node-to-node resistance.

Add/Subtract Connection Record

Add/subtract connection records inform other members of a network of the connection status between two nodes. JES2 is the only networking product that sends add/subtract connection records. Members of a MAS configuration may have identical node names and can be distinguished by the member names. The add/subtract connection record can be included with other connection records in an NJE record. The data in the NJE record is not compressed or compacted.

Table 43. Fields in an Add/Subtract Connection Record

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCRCB	0	1	binary	X'F0'		All path manager RCBs contain X'F0'.
NCCSRCB	1	1	char	none	none	The type of connection control record. Subtract connection records contain C'N'. Add connection records contain C'M'.

Table 43. Fields in an Add/Subtract Connection Record (continued)

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCADL	2	1	binary			Length of the response signon record from the RCB to the end of the add/subtract connection record.
NCCANODA	3	8	char			The name of the low-end node.
NCCAQULA	B	1	binary	1	1-7	Member number for the low-end (if shared spool).
NCCANODB	C	8	char			The name of the high-end node.
NCCAQULB	14	1	binary	1	1-7	Member number for the high-end (if shared spool).
NCCAEVNT	15	4	binary			The connection event sequence.
NCCAREST	19	2	binary		0-8000	The total node-to-node resistance.

Terminating a Session or Connection

Signoff records inform the other node that the connection or session between two nodes is terminated.

Table 44. Fields in a Signoff Record

Field Name	Offset in Hex	Length	Type of Data	Default Value	Valid Range	Description
NCCRCB	0	1	binary	X'F0'		All path manager RCBs contain X'F0'.
NCCSRCB	1	1	char	none	none	The type of connection control record. Signoff records contain C'B'.

Chapter 8. Presentation Services

NJE packages data in NJE records within transmission buffers. It uses one buffer format for non-SNA transmissions (BSC and CTC) and another for SNA transmissions. This chapter describes the formats of those buffers and the data within those buffers.

Non-SNA Buffer Format

Transmission Buffer Size

The minimum size of a transmission buffer is 300 bytes which allows the transmission buffer to contain a complete 256-byte record, compression bytes, and BSC control bytes. The maximum size of a transmission buffer is 32K. A BSC connection can use a smaller buffer size than a CTC connection because the time to retransmit and the probability of errors on a BSC connection are greater than on a CTC connection. Each job header, data set header, and job trailer must be the first record in its transmission block.

Initial signon record flag NCCIPACK allows you to send multiple NJE headers in a single buffer. In all other cases, you must send the job header, data set header, and job trailer in separate buffers.

Compressing Data for BSC and CTC Transmissions

For BSC and CTC NJE transmission, data compression is always used. Compression reduces the length of records for transmission by removing blanks and duplicate characters. String control bytes (SCBs) are used to compress data before placing each record into the transmission buffer.

BSC and CTC NJE transmissions use compression only; compaction (reducing the length of records by representing certain 8-bit characters with only 4 bits) is **not** used.

BSC and CTC Buffers

A variable number of NJE records can be combined into a transmission buffer. Each record in the buffer is comprised of a series of character strings and prefixed by a string control byte (SCB). Each NJE record is prefixed by a record control byte (RCB) and a subrecord control byte (SRCB).

To control the flow of individual streams, a function control sequence (FCS) is added to each transmission buffer. Finally, a block control byte (BCB) is added as the first character of each transmission buffer for error detection and correction.

Following is the layout of a typical transmission buffer for BSC and CTC communications.

```
Characters

┌──────────┐
│   BCB    │   - block control byte
│          │     (block and sequence control information)
├──────────┤
│   FCS    │    - function control sequence
├──────────┤
│   FCS    │    - function control sequence (continued)
├──────────┤
│   RCB    │   - record control byte for record 1
├──────────┤
│  SRCB    │   - subrecord control byte for record 1
├──────────┤
│   SCB    │   - string control byte for record 1
├──────────┤
│          │
│   data   │    - character string
│          │
├──────────┤
│   SCB    │   - string control byte for record 1
├──────────┤
│          │
│   data   │    - character string
│          │
├──────────┤
│   SCB    │   - terminating SCB for record 1 (end-of-record)
├──────────┤
│   RCB    │   - record control byte for record 2
├──────────┤
│  SRCB    │   - subrecord control byte for record 2
├──────────┤
│   SCB    │   - string control byte for record 2
└──────────┘
         .
         .
         .
┌──────────┐
│   SCB    │   - terminating SCB for last record
├──────────┤
│   RCB    │   - transmission buffer terminator
└──────────┘
```

Figure 15. Format of a Transmission Buffer for BSC and CTC

Block control byte (BCB)

Every BSC buffer begins with a block control byte (BCB) which contains inbound
and outbound buffer sequence counters. These counters are used to synchronize
inbound and outbound transmissions and to detect and correct sequence errors.
Table 45 shows the BCB bit definitions.

Table 45. BCB bit definitions

Binary	Meaning
1...	Must be 1
1xxx	Control information as follows:
1000 cccc	Normal block
1001	Bypass sequence count validation (sometimes called "BCB ignore bit")
1010 cccc	Reset expected block sequence count to cccc.

Table 45. BCB bit definitions (continued)

Binary	Meaning
1011	Reserved for IBM's use
11xx	Reserved for IBM's use

After a BSC line completes initialization, each directly-attached node initializes the inbound and outbound BCB counters to 0000 and maintains it in modulo sixteen. The outbound BCB count is incremented by one each time an acknowledgement (ACK) is received. The inbound BCB count is also updated to reflect the number of transmission buffers that the node has received. When a node receives a BCB the outbound count should be one more than the last BCB it received (the inbound count). If so, data transmission is normal and no data has been lost.

If the count in the BCB is not what is expected, the receiving node must indicate an error by sending an RCB indicating a BCB sequence error. When the sequence error is received, the node receiving it terminates the connection because error recovery is not possible.

If a duplicate BCB is received, the system receiving it assumes the last transmission buffer it sent has been lost and its last transmission buffer that did not receive an acknowledgement must be re-sent. The duplicate buffer is discarded. This allows the NJE network to recover from errors rather than terminating the connection by indicating a sequence error.

Because the BCB is also used to detect lost blocks, a null block containing a BCB should always be sent as an acknowledgement rather than using a DLE ACK0 as the acknowledgement. Figure 16 and Figure 17 on page 146 clarify the importance of using the DLE as an acknowledgement.
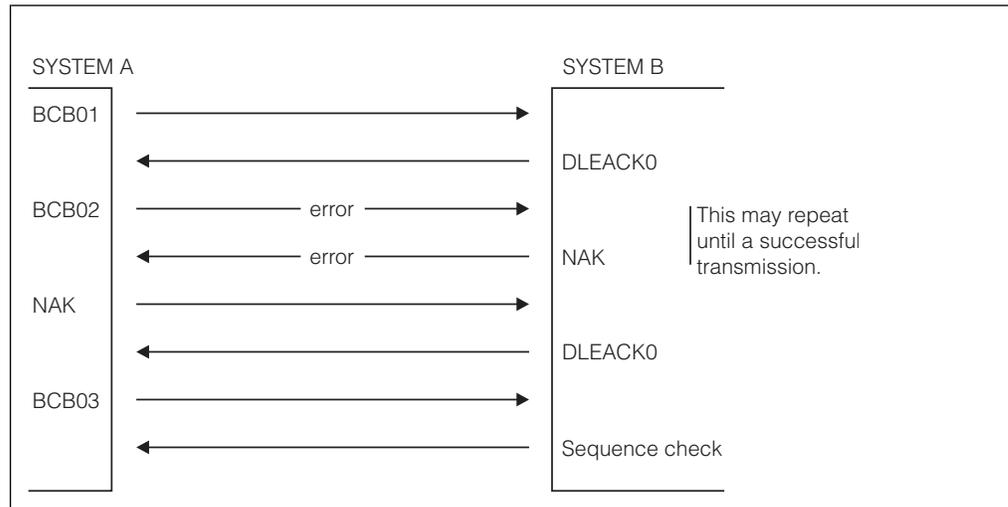
Figure 16. Results of Not Transmitting Null Records.
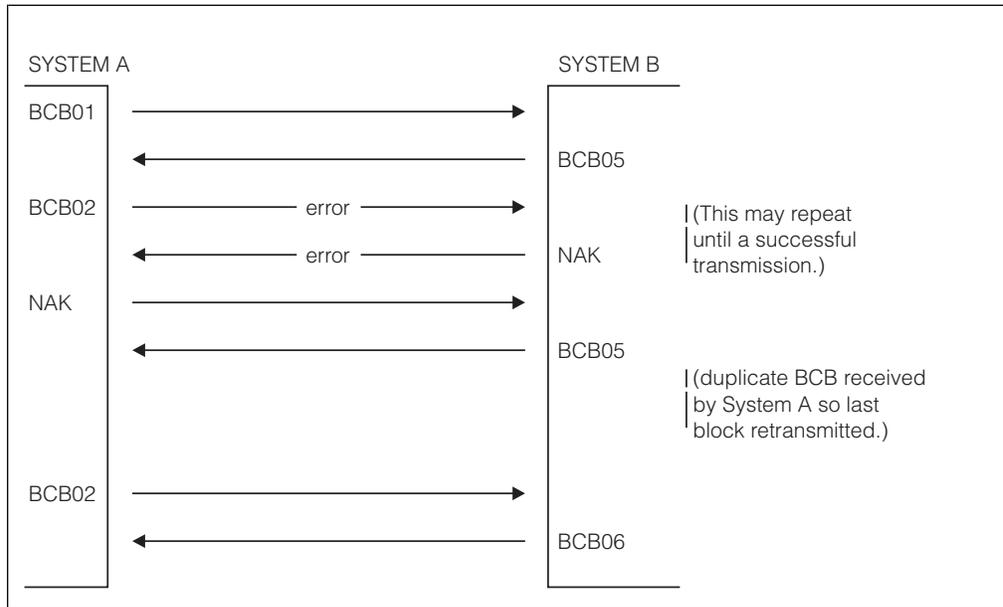
Figure 17. Correct Recovery with Null Records

Use Table 46 to determine the state a node is placed in and the action it takes after receiving a transmission block.

Table 46. Transmission Block Handling State Table

Current state of node	Event	Action node takes	New state of node after action is taken
State 1: Waiting for a response	Receives a data block		Checks the BCB
	Receives DLE ACK0	Transmits last non-NAK block of data	Waits for a response
	Receives a NAK	Retransmits last non-NAK	Waits for a response
	Receives an error (not CE-DE status)	Increments error count	Checks error count
State 2: Checking a BCB	Corrects the BCB	Transmits last non-NAK block of data	Waits for a response
	Detects a duplicate BCB	Retransmits the last non-NAK	Waits for a response
	Detects an incorrect BCB	Transmits a sequence error	Waits for a response
State 3: Recovering from an error and waiting for a response	Receives a data block		Checks the BCB
	Receives DLE ACK0	Transmits last non-NAK block of data	Waits for a response
	Receives a NAK	Retransmits last non-NAK	Waits for a response
	Receives an error (not CE-DE status)	Increments error count	Checks error count
State 4: Checking error count	Error count is not exceeded	Transmit a NAK	Recovers from the error and waits for a response

Function Control Sequence (FCS)

Following the BCB are two bytes, called function control sequence (FCS), used by each receiver to control inbound flow. There is one bit to hold (off) or enable (on) *each* of the eight inbound streams and one bit to hold/enable *all* inbound streams.

The latter bit is referred to as the "wait-a-bit". Table 47 defines the binary representation and the meaning of bits in the FCS.

Table 47. FCS Stream Bit Definitions

Binary	Meaning
1... .... 1...	Must be 1... .... 1... ....
.0..	Normal state
.1..	Suspend all stream transmission (Wait-a-bit)
..rr .... ..rr	Reserved for IBM's use
.... 1...	Function stream identifier for SYSIN stream 1
.... .1..	Function stream identifier for SYSIN stream 2 or SYSOUT stream 7
.... ..1.	Function stream identifier for SYSIN stream 3 or SYSOUT stream 6
....1	Function stream identifier for SYSIN stream 4 or SYSOUT stream 5
.... 1...	Function stream identifier for SYSIN stream 5 or SYSOUT stream 4
.... .1..	Function stream identifier for SYSIN stream 6 or SYSOUT stream 3
....1.	Function stream identifier for SYSIN stream 7 or SYSOUT stream 2
....1	Function stream identifier for SYSOUT stream 1
.... .1..	Function stream identifier for console tasks

The FCS bit assignments cause the BSC protocol restriction of a maximum of eight SYSIN and SYSOUT streams concurrently in each direction.

The compressed transmission records are placed in the buffer after the FCS bytes. Each transmission buffer contains transmission records that belong to the same NJE job. NJE protocols do not allow transmission records from different streams to be placed in the same transmission buffer.

The system receiving an FCS containing Wait-a-bit that is on, should delay responding. The delay can be in the range of one or two seconds if the Wait-a-bit was sent with a null record. A delay of greater than two seconds is not recommended because a timeout will occur if the response is not received within three seconds. If the data was sent with the wait-a-bit, the system should respond immediately. The immediate response allows the system requesting the delay to continue to transmit data as fast as possible. The approach of always delaying causes transmission delays whenever a system has no data to transmit, but cannot receive data.

The node receiving the FCS containing a Wait-a-bit that is on should respond with either a:
- DLE ACKO
- Null buffer.

IBM recommends responding with null buffers. If a DLE ACK0 is sent, the node receiving it responds to the DLE ACK0 based on the last FCS that was sent. (This

is the normal response to a DLE ACK0). NJE networking does not allow a node to transmit a data record in response to a wait-a-bit record.

String Control Byte (SCB)

In BSC and CTC NJE transmissions, the record segments are first compressed. This compression results in string control bytes (SCBs) interspersed in the stream data with duplicate character strings compressed. The resulting format always begins with an initial SCB and may have additional SCBs interspersed.

The SCB defines how many blanks or repeating non-blank characters are to be inserted by the receiver (up to 31) or the length of strings without any duplicate characters (up to 63 characters). An SCB **must** appear at least once every 64 bytes, and describe up to 63 bytes of following data. A "stand-alone" SCB with a value of zero, sometimes referred to as an "EOR SCB", is always placed at the end of each NJE record.

The SCB bit settings specify the compression parameters. SCBs are always used, whether or not the data is actually compressed; that is, even non-compressed data must be interspersed with SCBs that have the two high-order bits set (see Table 48).

Table 48. SCB Definition

2 high-order bits	Remaining bits	Meaning
00	00 0000	Indicates the end of the NJE record.
01	00 0000	Indicates the transmission of the stream should be terminated.
10	0b bbbb	Indicates that 'bbbbb' blanks should be inserted after the SCB.
10	1d dddd	Indicates that the single character following the SCB should be duplicated 'ddddd' times.
11	cc cccc	Indicates that 'cccccc' non-compressed characters (maximum of 63) follow the SCB. (The record is compressed; these characters are not.)

Record Control Byte (RCB)

Each logical record begins with a record control byte (RCB), which contains the stream identifier (type and number). In non-SNA transmissions, the end of the record is defined by a null string control byte (SCB) for compressed records, or by the data length byte for non-compressed records (signon, signoff, and path manager records).The initial signon record NCCIMRCB flag allows you to send mixed RCBs in a single transmission buffer. To use this flag, you must use the wait-a-bit only and ignore the FCS stream flags that indicate there is more tham one stream in a buffer. These messages cannot be avoided. See 2-93 for a description of this field.

If you do not use the initial signon record NCCIMRCB flag, your transmission buffer cannot contain records with different RCBs. The connection must be terminated (all streams) if a transmission buffer is received that contains an unexpected, unrecognized, or incorrect RCB. This includes:

- RCBs for streams that have not been started.
- RCBs for different streams in the same buffer.

- Undefined RCB values.

When the next compressed transmission record will not fit in the buffer, a special "stand-alone" RCB of X'00' (EOB) is placed after the last record and the buffer is truncated at this point for transmission.

Segments are never spanned across transmission buffers, however, uncompressed records may be.

Valid RCBs received out of sequence will also cause a connection to terminate. For example, a X'B0' receiver cancel with an unstarted stream referenced in the SRCB constitutes an error situation.

A request to start a stream that has already been started is handled differently. In this case, the request is rejected with the X'B0' (permission denied) RCB. The connection is not terminated; instead, the transmitting system can terminate the stream upon receipt of the X'B0' RCB. Table 49 depicts possible binary and hexadecimal values and their meanings in a RCB.

Table 49. RCB Definition

Binary	Hex	Meaning
0000 0000	00	End-of-block (BSC)
rrrr rrrr	01-8F	Reserved
1001 0000	90	Request to initiate stream (SRCB=RCB of stream to be initiated)
1010 0000	A0	Permission to initiate stream (SRCB=RCB of stream to be initiated)
1011 0000	B0	Negative permission or receiver cancel (SRCB=RCB of stream to be denied)
1100 0000	C0	Acknowledge transmission complete (SRCB=RCB of stream received)
1101 0000	D0	Ready to receive stream (SRCB=RCB of stream to be received)
1110 0000	E0	BCB sequence error
1111 0000	F0	General control record
1001 0001	91	Reserved
1rrr 0001	A1-F1	Reserved
1001 0010	92	Reserved
1rrr 0010	A2-F2	Reserved
1iii 0011	93-F3	Reserved
1iii 0100	94-F4	Reserved
1iii 0101	95-F5	Reserved
1iii 0110	96-F6	Reserved
1rrr 0111	97-F7	Reserved
1iii 1000	98-F8	SYSIN record
1iii 1001	99-F9	SYSOUT record
1001 1010	9A	Operator command/console message
1rrr 1010	AA-FA	Reserved

Table 49. RCB Definition (continued)

Binary	Hex	Meaning
1rrr 101	9B-FB	Reserved
1rrr 1100	9C-FC	Reserved
1rrr 1101	9D-FD	Reserved
1rrr 1110	9E-FE	Reserved
1rrr 1111	9F-FF	Reserved

i   represents 1 or 0 provided this produces a value within the range shown in the hex column.

iiii
     may be from 1 to 7 and corresponds to the stream number.

r   represents 1 or 0 provided this produces a value within the range shown in the hex column. All *r* values are reserved.

For complete system-dependent and product-specific information, see Appendix A, "System-Dependent Considerations," on page 219.

Subrecord Control Byte (SRCB)

The subrecord control byte (SRCB) is interpreted according to the RCB value, as shown in the following figures. The SRCB defines the record type (job header, data set header, job trailer, or user data record) plus SYSOUT carriage control type and user data spanning indicators (first, last, or middle segment). See "Format of Data Records Containing Spanned Data" on page 158 for additional information on records containing spanned data.

Table 50. SRCB Definition - Part 1

RCB	SRCB
00	None
90	RCB of stream to be initiated
A0	RCB of stream to be initiated
B0	RCB of stream to be cancelled or rejected
C0	RCB of completed stream
D0	RCB of ready stream receiver
E0	Expected count - BCB sequence error (received count is in BCB)
F0	An identification character as follows: • A = Reserved • B = Network SIGNOFF • C-H = Reserved • I = Initial network SIGNON – Must be only record in transmission buffer • J = Response to initial network SIGNON – Must be only record in transmission buffer • K = Reset network SIGNON • L = Accept (concurrence) network SIGNON • M = Add network connection • N = Delete network connection • O-R = Reserved for IBM's use • S-Z = Unused

Table 50. SRCB Definition - Part 1 (continued)

RCB	SRCB
91-94	Reserved

Note: F0 records are not compressed. Except where noted (I and J), multiple F0 records may be blocked in a single transmission. F0 records cannot be blocked with other records within a transmission.

Table 51. SRCB Definition - Part 2

RCB	SRCB
95-F5	Reserved
96-F6	Undefined
97-F7	Undefined
98-F8	NJE SYSIN control information as follows:
	• 1000 0000 - Standard record
	• 1100 0000 - Job header
	• 1110 0000 - Data set header
	• 1101 0000 - Job trailer
	• 1111 0000 - Reserved
	• 1111 0000 - Reserved for IBM's use
99-F9	NJE SYSOUT control information as follows:
	• 10cc 0000 - Carriage control type as follows:
	– 1000 0000 - No carriage control
	– 1001 0000 - Machine carriage control
	– 1010 0000 - ASA carriage control
	– 1011 0000 - CPDS page mode records (with carriage control)
	• 10cc ss00 - Spanned record control as follows:
	– 10.. 0000 - Standard record (not spanned)
	– 10.. 1000 - First segment of spanned record
	– 10.. 0100 - Middle segment of spanned record
	– 10.. 1100 - Last segment of spanned record
	• 11cc 0000 - Control record as follows:
	– 1100 0000 - Job header
	– 1110 0000 - Data set header
	– 1101 0000 - Job trailer
	– 1111 0000 - Reserved for IBM's use
9A	Operator Command/Message (NMR)
	• 1000 0000 (X'80')

Reason Control Bytes for a Rejected Stream

The reason bytes for 'B0', the RCB of a stream to be cancelled or rejected, are defined below. See Table 50 on page 150 for the complete list of Subrecord control byte definitions.

The following reason codes are defined, with additions needed for RSCS noted on the specific reason codes.

X'04'   Normal response to an ABORT

X'08'   Receiving system in SHUTDOWN

X'0C'   RIF received for an unsupported stream

> **X'0C04'**
>   Stream not supported (NR)
>
> **X'0C08'**
>   Stream drained
>
> **X'0C0C'**
>   Unknown stream (NR)
>
> **X'0C10'**
>   FCS conflict

X'10'   Resource lack

> **X'1004'**   Real storage (NR)
>
> **X'1008'**   Virtual storage
>
> **X'100C'**
>   Spool space
>
> **X'1010'**   CPU resources

X'14'   Operator command

> **X'1404'**   HOLD command for connection
>
> **X'1408'**   STOP command for connection
>
> **X'140C'**
>   FLUSH command for connection

X'18'   Protocol violation

> **X'1804'**   Last transmission not correctly terminated
>
> **X'1808'**   Compression/compaction error
>
> **X'180C'**
>   Records sent out of sequence
>
> **X'1810'**   Mixed RCBs within a buffer (M)
>
> **X'1814'**   Undefined RCB/SRCB combination

X'1C'   Datastream error

X'20'   File rejected by an installation-defined exit routine or by some other security check

KEY: (NR)   Not issued by RSCS
(R)Not issued when multiple RCB extension exists.

Figure 18. B0 Reason Bytes

Sending Sequences for 'B0' Reason Code

Table 52. Sending Sequence for a X 'B0' RCB on both BSC and SNA links.

Type	RCB	SRCB	SCB	LEN	Reason Byte
BSC	BO	9x	C2	--	rc1 rc2
SNA	BO	9x	05	01	rc1 rc2

SNA Buffer Format

The SNA data buffer is called a request/response unit (RU). The RU carries control information and data between logical units. A **control RU** contains a request or acknowledgement, while a **data RU** contains function management headers, or SCBs and data.

Transmission Buffer Size

A data RU may contain as many job networking records as can fit into the RU. The BIND parameters specified at the start of the session determine the size of a data RU. The maximum size of an RU can be 65535 bytes. The maximum length for the job networking record is 259 bytes (256 bytes of user data plus a 3-byte record identifier).

Compressing and Compacting Data for SNA Transmissions

SNA data transmission requires the use of compression. to reduce the length of records for transmission by removing blanks and duplicate characters. Compaction (reducing the length of records by representing certain 8-bit characters with only 4 bits) is optional and is controlled by NJE and SNA protocols. (See Session Initialization in Link Protocols for additional information on compaction and compression.)

The SNA string control byte (SCB) defines the beginning and end of compressed and compacted data. An SCB must begin each RU following the exchange of FM headers. See "String Control Byte (SCB)" on page 148 for additional information for the SCB.

SNA Transmission Buffer

SNA transmission buffers contain SCBs, RIDs, and data records. Each data record is prefixed with a 3-byte header, called a record identifier (RID), which identifies the type of record that follows. The purpose of the RID is similar to that of the RCB and SRCB in non-SNA NJE transmissions. The three bytes of the RID are as follows:

RIDRCB

See Table 54 on page 156 and Table 55 on page 156.

RIDSRCB

See Table 54 on page 156 and Table 55 on page 156.

RIDRLEN

Contains the length of the data. NJE networking uses RIDLEN to locate the beginning of the next data record in the transmission buffer.

Figure 19 on page 154 illustrates the format of a SNA transmission buffer.

```
Characters

┌──────────┐
│   SCB    │     - string control byte for up to 63 bytes
├──────────┤
│  RIDRCB  │     - record control byte for record 1 (RID byte 1)
├──────────┤
│ RIDSRCB  │     - subrecord control byte for record 1 (RID byte 2)
├──────────┤
│ RIDRLEN  │     - length field for record 1 (RID byte 3)
├──────────┤
│   data   │     - character string
│    .     │
│    .     │
│    .     │
├──────────┤
│   SCB    │     - at arbitrary points in the buffer,
│          │     but at least every 64 bytes
├──────────┤
│   data   │     - character data
│    .     │
│    .     │
│    .     │
├──────────┤
│  RIDRCB  │     - record control byte for record 2
├──────────┤
│ RIDSRCB  │     - subrecord control byte for record 2
├──────────┤
│ RIDRLEN  │     - length field for record 2
├──────────┤
│   data   │
│    .     │
│    .     │
│    .     │
└──────────┘
```

Figure 19. Format of a Transmission Buffer for SNA

String Control Byte (SCB)

In contrast with non-SNA use of SCBs, for SNA buffers, SCBs are used to
compress/compact the entire buffer; that is, the entire buffer is treated as data as
far as the SNA transport subsystem is concerned.

The SCB bit settings specify the compression and compaction parameters, but the
encoded bits are different for SNA than for non-SNA transmissions. SCBs are
always used, whether or not the data is actually compressed; that is, even
uncompressed data must be interspersed with SCBs that have the two high-order
bits set.

The one-byte SCB consists of a 2-bit description field, followed by a 6-bit count
field. This count is the number of characters that are described by this SCB.
Therefore, one SCB may replace up to 63 characters, or identify up to 63
intervening uncompressed characters before the next SCB. In all cases, a count of 0
is a reserved value. The SCB format is shown in Table 53.

Table 53. String Control Byte (SCB) Format for SNA

Description	Count	Usage
00	cccccc	No compressed characters follow
10	cccccc	Repeat blanks

Table 53. String Control Byte (SCB) Format for SNA (continued)

Description	Count	Usage
11 01	ccccc ccccc	Repeat the following non-blank character Compacted characters follow

The *only restriction* on the placement of SCBs is that an SCB must appear at least once every 64 bytes, and describe up to 63 bytes of following data. This restriction applies to non-SNA use of SCBs as well.

Compression SCB

Compression may be optionally indicated in the BIND for LU TYPE 0, however, compression is required for NJE using SNA. Two or more blanks and three or more non-blanks are compressed. For example, if 5 blanks were being compressed, the SCB would be B'10000101'. If 5 "A"s were being compressed, the SCB would be B'11000101' followed by the character A (or X'C1').

Compaction SCB

Compaction allows certain 8-bit character sequences to be represented in network transmission as 4 bits. The compacted characters are called master characters. When two master character are adjacent in a data stream, they are compacted from their normal 8-bit representation into 4 bits. Non-master characters may also be defined; these characters are not compacted, but when they are adjacent to master characters, they will not interrupt the compaction SCB in effect. All other characters are considered non-compactable and will be transmitted in their 8-bit form. Obviously, the non-compactable characters should be those that are least frequently used in the data stream.

Master and non-master characters are transmitted via FMH3 at session initialization.

Record Identifier (RID)

A record identifier (RID) is a 3-byte header that precedes every NJE SNA record. The RID identifies the type of record that follows. Its purpose is similar to that of the RCB and SRCB in BSC NJE.

The RID describes three types of records: network topology records, stream control records, and data records. For data records, the RID also contains the length of the data in that record. The maximum data length is 256 bytes.

The following figures depict the format of the three bytes of the RID:
1. RIDRCB - Table 54 on page 156
2. RIDSRCB - (depends on value of RIDRCB) Table 55 on page 156
3. RIDRLEN - (depends on value of RIDRCB) .

Note: Byte 2 of the RID, RIDSRCB, contains the same values as the SRCB for non-SNA records.

RID Record Control Byte (RIDRCB)

Table 54. RIDRCB: Byte 1 of the RID

Value	Record type	Meaning
X'90'	stream control	Request to allocate SYSIN/SYSOUT stream
X'A0'	stream control	Permission to allocate stream granted
X'B0'	stream control	Permission to allocate stream denied or receiver cancel
X'C0'	stream control	Acknowledge end of transmission
X'D0'	stream control	Receiver ready
X'F0'	network topology	An NJE topology record follows
B'1iii1000'	data	An NJE SYSIN stream data record follows
B'1iii1001'	data	An NJE SYSOUT stream data record follows
X'9A'	data	A nodal message record follows

Note: For all stream control records, the RIDSRCB is the RCB of the stream.

The *iii* in the value field for RIDRCB identifies the particular SYSIN or SYSOUT stream. Values of 1-7 binary are permitted; a value of zero is not allowed.

RID Subrecord Control Byte (RIDSRCB)
Notes:
1. For all stream control records, the RIDSRCB is the RCB of the stream.
2. For network topology records, RODSRCB contains the same values as the SRCB for non-SNA records.
3. For all data records, RIDSRCB contains the same values as the SRCB for non-SNA records.

Table 55. RIDSRCB: Byte 2 of the RID

RIDRCB	RIDSRCB
X'90'	RCB of SYSIN/SYSOUT stream to be allocated
X'A0'	RCB of SYSIN/SYSOUT stream for which permission to allocate has been granted
X'B0'	RCB of SYSIN/SYSOUT stream to be canceled or for which permission to allocate has been denied
X'C0'	RCB of SYSIN/SYSOUT stream for which end of transmission has been acknowledged
X'D0'	RCB of the initiated SYSIN/SYSOUT receiver
X'F0'	EBCDIC character identifying NJE topology record.
X'98'-X'F8'	Transmission end information X'00' - standard SYSIN stream end X'40' - SYSIN stream canceled by transmitter NJE SYSIN control information.
X'99'-X'F9'	Transmission end information X'00' - standard SYSOUT stream end X'40' - SYSOUT stream canceled by transmitter NJE SYSOUT control information.
X'9A'	X'80' (JES2 sends an SRCB of X'00' with NMRs.)

RID Subrecord Control Byte (RIDRLEN)

Table 56. RIDRLEN: Byte 3 of the RID

RIDRCB	RIDRLEN
X'90'	N/A (set to zero)
X'A0'	N/A (set to zero)
X'B0'	N/A (set to zero)
X'C0'	N/A (set to zero)
X'D0'	N/A (set to zero)
X'F0'	Length of network topology plus 3 (for RID)
X'98'-X'F8'	Length of data record - 1 [5]
X'99'-X'F9'	Length of data record - 1 [5]
X'9A'	Length of nodal message record - 1 [5]

Data Records

This section describes unspanned/spanned data format and trailing blank truncation. This information applies to both SNA and non-SNA NJE transmissions as described in the previous sections.

Format of Data Records Containing Unspanned Data

The record length (LRECL) includes the carriage control character (CCTL) and data. It does not include the record length field. The maximum size record that can be transmitted without using the spanned record format is 255 bytes, including the carriage control character.

```
0                1                          256
   +---------+------------------------------+
   |  LRECL  |      255 bytes of data       |
   +---------+------------------------------+
```
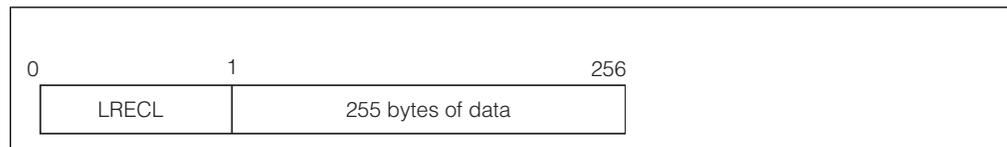
Figure 20. Data Record Without Carriage Control

If machine carriage control is used, any CCW operation code (opcode) may be sent in that record. Store-and-forward systems must forward all CCW opcodes intact. The destination may discard any records that it cannot process.

```
0              1            2                   256
   +---------+----------+----------------------+
   |  LRECL  |   CCTL   |   254 bytes of data  |
   +---------+----------+----------------------+
```
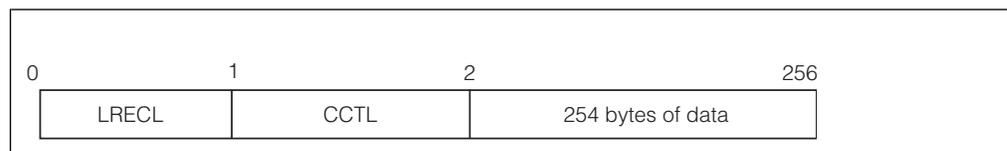
Figure 21. Data Record With Carriage Control

For NJE SYSIN jobs, the default record length and format for each record is a 80 bytes fixed. SYSIN data records do not contain carriage control.

---

5. For the last three values shown in the figure, a value of zero in RIDRLEN indicates that no data follows. The values X'01' - X'FF' represent record lengths of 2-256. Because NJE records begin with a one-byte length, a record length of 1 need not be represented.

Format of Data Records Containing Spanned Data

Spanned record support allows records longer than 255 bytes to be transmitted. For transmission, the records are broken into data segments of less than 256 bytes. The maximum size of a spanned record is 32760 bytes. All segments contain the segment length (SEGL) at the beginning. The total length of the logical record (LRECL) is transmitted following the SEGL in the first segment. (All other segments contain only the SEGL and data.)

The byte used by the SEGL is not included in the SEGL value. The two bytes used by the LRECL are also not included in the SEGL. A sample spanned record transmission is shown in Figure 22.
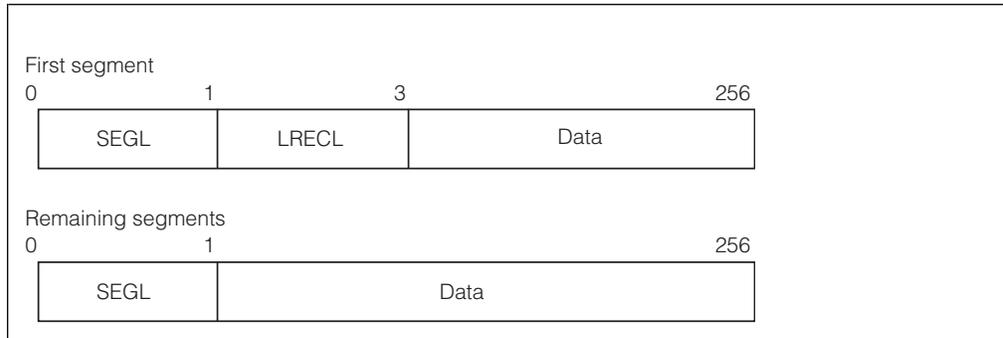
Figure 22. Spanned Data Record
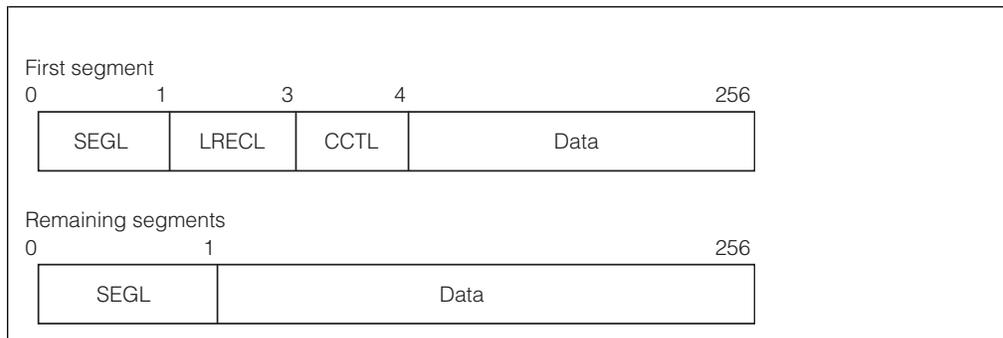
Figure 23. Spanned Data Record With Carriage Control

Each segment after the first can contain 255 bytes of data. The first may contain a maximum of 253 bytes of data (including the carriage control).

If machine carriage control is used, any CCW opcode may be sent in that record. Store-and-forward systems must forward all CCW opcodes intact. The destination may discard any records that it cannot process.

Trailing Blank Truncation

NJE allows the truncation of trailing blanks prior to transmission. The original LRECL is used to reconstruct the record. Any segmentation done for the purposes of controlling transmission buffers (or RUs) is done after blank truncation of the logical record.

Compression and compaction may also be applied for transmission, but these have reconstruction mechanisms independent of the original LRECL.

Further efficiency may be obtained by truncating blanks at the end of individual segments of spanned records prior to compression or compaction. In this case, the original SEGL is used to restore the segment. The concatenated segments are then used to build the original record. If required, the reconstructed record is padded with blanks to match the original 2-byte LRECL.

Chapter 9. Stream Control

This chapter describes the multi-leaving functions of NJE. After NJE records are placed into transmission buffers, the buffers are sent in a stream to a remote node. Each buffer begins with a 2-byte RCB/SCRB control field that defines the stream (SYSIN 1-7 or SYSOUT 1-7) to which this packet belongs. BSC and CTC packets begin with a BCB and two FCS bytes. The first SCB (compressed data) follows the RCB/SRCB field. Because SNA packets are compressed after you append the RCB/SRCB, the RCB/SRCB is only a logical beginning: the packet on the link really begins with an SNA SCB byte.

NJE protocols support two features of file transmission to expedite the flow of multiple files in each direction and minimize delays:

- Multi-leaving -- concurrently transferring multiple streams on the same BSC connection or SNA session.
- Full duplexing -- transferring a stream in one direction while receiving an unrelated stream on the same line in the other direction.

The following notes any differences in multi-leaving between the protocols when performing the following tasks:

- Providing BSC and CTDC protocols for stream control
- Providing SNA protocols for stream control
- Initiating stream transmission
- Using full duplex with BSC and SNA protocols
- Ending stream transmission normally
- Ending stream transmission abnormally
- Avoiding unnecessary stream transmission.

Providing BSC and CTC Protocols for Stream Control

BSC protocols allows a total of 8 NJE SYSIN or NJE SYSOUT jobs in either direction an NMR, and a control stream. BSC and CTC connections perform "pseudo" full duplexing because each node must flip-flop between the send and receive states. Both nodes are synchronized.

You should not send an **R I** record to a destination node with FCS code that suspends the stream. If the sending node violates this protocol, the destination node should respond with a **R C** request. The restrictions on the number of streams that may be concurrently active are documented in "Function Control Sequence (FCS)." Some products implement the same restrictions for SNA stream control, although NJE does not impose them.

Providing SNA Protocols for Stream Control

SNA protocols allow the concurrent flow of 28 NJE SYSIN and/or SYSOUT jobs in either direction, an NMR, and a control stream. All NJE records flow on logical streams, which correspond to transmitter and receiver pairs on the network link. NJE records are controlled by destination indications in the record identifier (RID). For information about the contents of the RID, see Record Identifier (RID) in Presentation Services.

NJE logical streams must be initiated before any data can be sent. An SNA stream control record must be the only record in the request/response unit (RU).

Initiating Stream Transmission
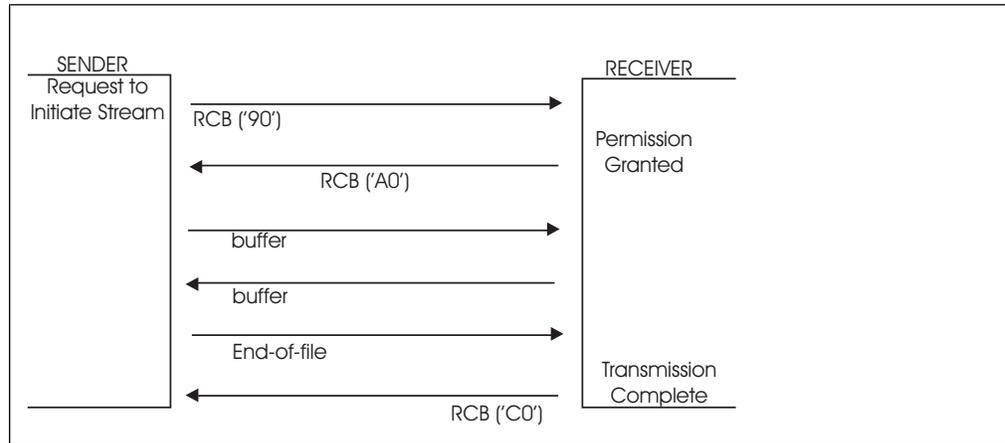
Figure 24. Protocols Used to Transmit a Stream

When a node is ready to transmit a stream, it prepares a control record to initiate the stream. A request initiate (**R I**) RCB with a value of X'90' indicates that a request has been made to initiate a stream. The receiving node receives the request and prepares a control record to indicate whether or not it can receive the stream. If the node can receive the stream, the receiving node responds by preparing an RCB with a value of X'A0'. If the stream is unwilling or unable to receive the new stream, it sends a receiver cancel (**R C**) RCB with a value of X'B0'

The sending node cannot send any transmission buffers until the proper control records are exchanged. The sender may only send a transmission buffer after the receiver has responded with a permission to initiate a stream (RCB X'A0'). Only one job may be sent for each request to initiate a stream, unless the multi-job stream option is in effect. If the receiving node receives a request to initiate a stream that is already active, the receiving node should return an **R C** record (RCB X'B0').

Using Full Duplex with BSC and SNA Protocols

The BSC protocol synchronizes nodes because both sides of a conversation must alternate between sending and receiving nodes. The receiving node can tell the sending node to hold one or more streams, while allowing the remaining streams to continue. When temporary link failures occur, the receiving node must notify the sending node so it can retry the transmission.

The SNA protocol is asynchronous; each end is in send and receive state concurrently. Because VTAM paces the stream flow at the session level, individual streams cannot be held as they are during BSC. However, because the transmission subsystem takes care of all errors, the receiving node can ignore temporary link failures.

The SNA asynchronous protocol is less error-prone than the BSC protocol for full duplexing, and for using long-delay links or satellite links.

Ending Stream Transmission Normally

The sending node sends an end-of-file control record when transmission for the stream is complete, as shown in Figure 25
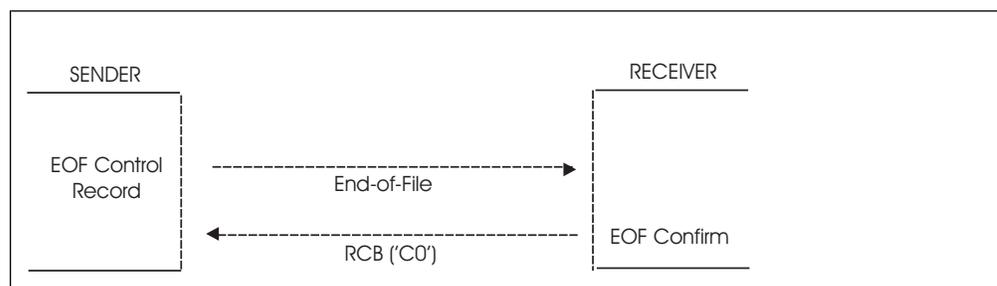
Figure 25. Protocols Used to Transmit a Stream

When transmission complete (T C or RCB X'CO') is received, the sender may purge all copies of the job on his own system. It may not do so before RCB X'CO' is received. At this point the stream is logically closed. It is also valid to receive an RCB X'BO' in response to an end-of-file when the receiver wishes to abort the file.

Table 57. EOF Specification for Both BSC and SNA Links

Link	RCB	Stream ID	SRCB	SCB	LEN
BSC	98-F8 99-F9	SYSIN SYSOUT	00	00	--
SNA	98-F8 99F9	SYSIN SYSOUT	00	00	00

Ending Stream Transmission Abnormally

When a problem arises during the transmission of jobs, the sender and the receiver must agree on what to do with the cancelled job. If both the sender and receiver purge the job, it cannot be recovered. If the sender saves the job and retransmits it to the receiver later, it could cause job duplication if the receiver has not purged the originally cancelled job. Job duplication has become a serious problem in networking.

Use the following transport service protocols to ensure accurate communication across nodes and network integrity.

NJE supports two transport protocols, the Less Reliable Transport Service (LRTS) and the More Reliable Transport Service (MRTS). The following describes the use of both protocols.

Less Reliable Transport Service

If the receiving node enters an operator cancel command or detects stream errors, it can refuse the stream by responding with an RCB X'BO' on a receiver cancel (**R C**) record.

Either the sender or the receiver may conclude the transmission of a job before the end-of-file or transmission complete control record is sent. This abnormal termination may be caused by any problem either the sender or receiver has with a particular job *after* the initiating protocol has completed (that is, the receiving node has sent a permission granted RCB X'A0' control record).

Note: When the sending node uses SNA protocols, a RIDSRCB of X'40' rather than an SCB of X'40' indicates the sending node initiated stream end. (The SCB is described in String Control Byte in Presentation Services.)

The following figure displays the abnormal end LRTS protocol when the sender initiates ending the session for the receiving node.
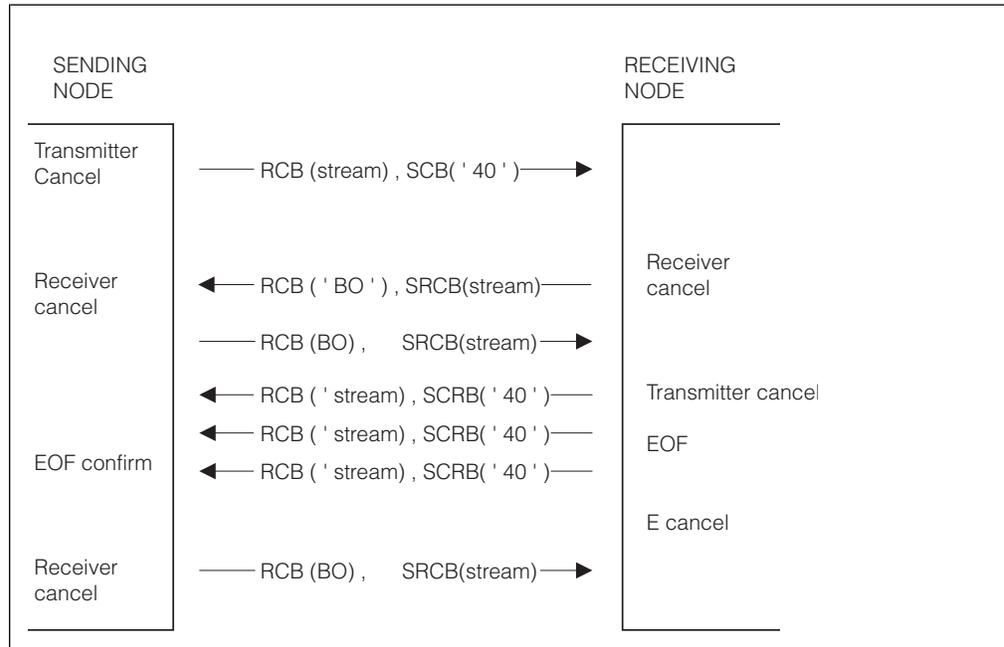
Figure 26. Sending Node Initiates Stream End

No additional data may be sent on this stream until the receiver cancel control record is received, and that in turn implies the stream has been closed. A new request for permission must be sent before another job is transmitted. If the system problems caused the ending of the stream, rather than a user or operator cancel request, a copy of the abnormally ended job should be kept in the sender's spool space for transmission at a subsequent time.

Abnormal termination protocol is as follows:
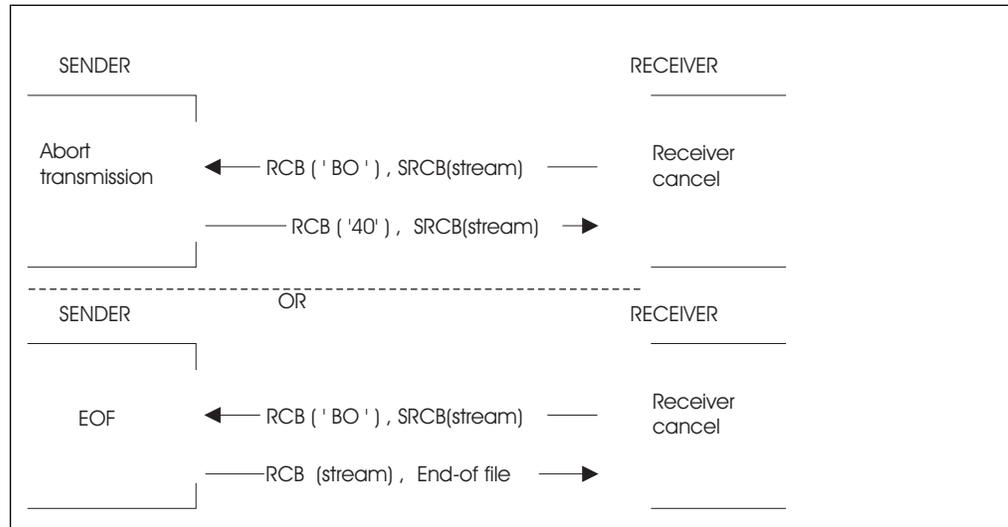
Figure 27. Receiving Node Initiates Stream End

Only after the protocol described above has been followed may the sender attempt to transmit a new job in the completed stream. To avoid retransmitting the same job again, operators should place it into a hold state. Under most circumstances, if the receiver ends the job once, it is ended during subsequent attempts to retransmit the job.

More Reliable Transport Service

A transmitting node initiates the MRTS protocol by setting the NCCIMRTS flag (X'04') of the NCCIFEAT field during initial signon. To specify this field, see 2-92. To accept a transmission stream, the receiving node responds by setting the same value in the response signon record.

The More Reliable Transport Service uses the same procedures found in the Less Reliable Transport Service, with the following additional restrictions:

* Any **R C** requires a **T C** response: EOF is not a valid response.
* Only one **R C** can be sent in a stream. A sending node discards all subsequent records sent until the receiving node responds with a **T C** record received.
* An EOF record has only one correct response: **E C**.

   If the receiving node sends an EOF at the same time it receives an **R C**, the EOF is ignored.
* A receiving node terminates all connection with a sending node that ignores an **R C**.

For how to code particular protocol records, see the tables below. Note that reason codes are indicatee within the SCB byte for BSC links and the Length byte for SNA links.

Table 58. Transmitter Cancel for SNA and BSC Abnormal Terminations

Link	RCB	Stream ID	SRCB	SCB	LEN
BSC	98-F8 99-F9	SYSIN SYSOUT	40	00	--
SNA	98-F8 99-F9	SYSIN SYSOUT	40	--	00

Table 59. Receiver Cancel for SNA and BSC Abnormal Terminations

Link	RCB	Stream ID	SRCB	SCB	LEN
BSC	B0	SYSIN SYSOUT	98-F8 99-F9	00 C2	--
SNA	B0	SYSIN SYSOUT	98-F8 99-F9	--	00 01

Table 60. EOF Confirm for SNA and BSC Abnormal Terminations

Link	RCB	Stream ID	SRCB	SCB	LEN
BSC	99-F9	--		00	00
SNA	C0	--	98-F8	00	00

Because the operating systems handle **R C** control records differently, users may follow the LRTS protocol. If JES2 or RSCS are the subsystems at the sending node, the transmitter is held if an **R C** control record is received. POWER and JES3 drain the NJE jobs that are waiting to be sent and requeue the NJE jobs so that the job can be resent.

Only after the protocol described above has been followed may the sender attempt to transmit a new job in the completed stream. To avoid retransmitting the same job again, operators should place it into a hold state. Under most circumstances, if the receiver ends the job once, it is ended during subsequent attempts to retransmit the job.

Avoiding Unnecessary Stream Transmissions

You can avoid one level of exchange between two nodes by following the multi-job stream option described below. If nodes exchange the signon concurrence flag -- NCCIRIF -- in the signon records exchanged at initialization, you can eliminate the X'90'-X'AO' exchange after the receiver sends back the first X'C0'. This method of transmission is recommended for sessions over satellite links because it reduces the propagation delay.

Both systems using the signon concurrence flags for this method of transmission to work.

Systems using the signon concurrence flag can end the transmission of a job for a number of reasons, as described in the 'B0' reason codes on 3-12. The receiving node can send an **R C** record (RCB X'B0') in response to a request to initiate (RCB X'90'). In this instance, the transmitting node should consider the receiving node drained until it receives a receiver online request (RCB X'D0'). If the receiving node does not wish to receive an object but is in the midst of transmission, it can send a cancel transmission (RCB X'B0') with a reason control byte to explain the reason. Then the transmitting node can place that file in a HOLD state and reopen the stream to send another object.

Each job in the stream still ends with an EOF travelling from the sender node to the receiving node. To end transmission, the receiver must send an **E C** request (RCB X''C0''). The receiver can also send an **R C** (RCB X''B0'') in response to the EOF. This record terminates the stream and rejects the last job sent.

Note that the sending node should notify the operator before deleting the stream from his system.

Under the MRTS protocol, the **R C** is treated as a cancel request for both the last job and this stream. In this case, the sending node **must** respond with a **T C** request before restarting the stream with an **R I**.

Under the LRTS protocol, the **R C** is treated as an EOF response.

Chapter 10. NJE Connection Control

Depending on the networking facility and the type of protocol a node uses to transmit a transmission buffer, a node can establish a connection with another node in the network.

If one of the nodes attempting to establish a connection has JES2 installed, the network path manager (NPM) influences the protocols used to establish the connection. JES2 installations can include the CONNECT initialization statement in their initialization streams to specify a node does not use NPM. See "Establishing a Connection or Session With NPM" on page 171 for the protocols used if you are establishing a connection with a node that has JES2 installed. See "Establishing a Connection Without NPM" on page 170 for the protocols used if you are establishing a connection with nodes that *do not* have JES2 installed. The Signon Concurrence Feature is an option your installation can implement. "Signon Concurrence Feature" describes the feature.

Signon Concurrence Feature

The signon concurrence feature is an optional element of the signon protocol. It allows two systems to determine that each are able to work in an extended mode on the communications line running between them.

External Interface

A method must be provided to allow the system programmer to define the extended features to be used on each line. Optionally, the networking system could assume all the features are to be used if the system at the other end concurs.

Any system with extended features should set the bit stating that an extended feature is available and must check the bit in the response from the other side. When a specific bit is set by both systems, then the corresponding extended NJE feature is used between them.

Implementation Example

Each system has a systems supported features word (SSFW) in a common area. This word contains the bit mask describing all the additional features that this system is able to support. In addition, each system adds an authorized features word (AFW) to a control block that is unique to each line. This word contains the features that are to be enabled for the given line.

At signon, the implementing system performs logic equivalent to the following:
- Primary
  1. Copy the SSFW into NCCIFEAT before transmitting the I record.
- Secondary
  1. Copy NCCIFEAT from the I record into the AFW.
  2. AND the AFW with the SSFW, storing result in the AFW.
  3. Copy the AFW into NCCIFEAT of the J record.
- Primary
  1. Copy NCCIFEAT from the J record into the AFW.

2. AND the AFW with the SSFW and store the result in the AFW.
3. Compare the AFW with NCCIFEAT. If the fields are different, terminate the line or use the features indicated by NCCIFEAT.

The previous assumes the NCCI is expanded before processing is performed. Systems not supporting this feature will have NCCIDL set to a value too small to include the NCCIFEAT bytes. The system receiving such an I or J record must then assume that none of the optional features are to be used on the session.

Protocol Description

The system transmitting the I record (primary) must set the features bits in the I record showing the features that are both present and desired. (The system programmer may have the option of not enabling a given feature on a given line.)

When an I record is received, the secondary node must look at the extended features bits and for any features that it is also willing to use and must transmit the corresponding feature bit in the J record. The secondary node should then use the features. Upon receiving the J record, the primary node should then examine the features bits and use any features that are specified. See Initial Signon Record in Formats of NJE Transfer Units for a description of the actual bit definitions.

Establishing a Connection Without NPM

The nodes first attempt to establish a connection by exchanging a set of control records. The control records vary depending on the type of protocols the nodes will use to transmit the stream.

The NJE primary node always sends the initial signon record. If BSC or CTC protocols are being used, the NJE primary node is the node that receives the DLE ACK0 BSC control record. If SNA protocols are being used, the NJE primary node is the node with the higher node name. The SRCB that precedes the initial signon record contains an identifier of C'I'. Initial Signon Record in Formats of NJE Transfer Units describes the format of the initial signon record.

The node receiving the initial signon record, called the NJE secondary node, responds with a response signon record. The SRCB that precedes the response signon record contains an identifier of C'J' The J record contains a connection event sequence of minus one (X'FFFFFFFF'). Figure 28 describes the format of the response signon record.

Either End (Primary) — Initiate Signon (I) → Other End (Secondary)

Response (J) (CES=X'FFFFFFFF') ←
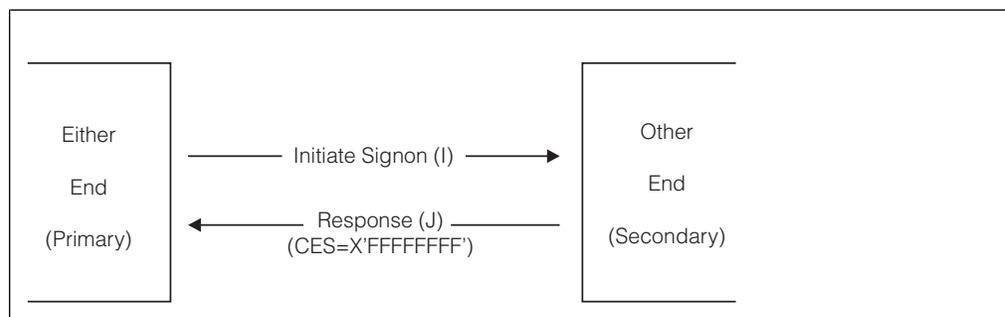
Figure 28. Connection Protocols used by the Non-Path Manager Nodes

Establishing a Connection or Session With NPM

A node that has JES2 installed uses the Network Path Manager (NPM) to establish a connection or session between nodes. NPM is also responsible for:

- Promulgating connection information to the other members.
- Maintaining information about which lines should be used to reach a given node.
- Informing other subcomponents which nodes should be reached over a given line.
- Providing routing information for jobs and messages.
- Processing NJE signon and connection/disconnection records from other Network Path Managers.
- Selecting a line based on specified resistances.

Nodes that use NPM may have code parameters on initialization statements that prohibit connect and disconnect records from being sent for some connections. The following are examples of connections where the installation may prevent connect and disconnect records from being sent:

- **Private connections**. Connection records are created for the directly connected node. For private connections no other nodes are notified of the nodes existence.
- **Non-path manager connections** Connection records are not created when a node using NPM signs onto a node not using NPM.

Signon Connection Protocol Between Two Nodes With NPM

The protocol for directly connecting two NJE nodes depends upon the capabilities of the two Network Path Managers, the number of lines between the two systems, and the installation-supplied names of the nodes. A unique connection in an NJE network has 4 basic parts:

1. The identification of the system with the low EBCDIC node name (low end).
2. The identification of the system with the high EBCDIC node name (high end).
3. The Connection Event Sequence (CES). The CES is the high-order four bytes from the time of day clock.
4. The resistance of the connection.

The CES is a binary value that increases each time the low end system initiates or allows a connection. Since the value is always increasing, NPMs can decide what information is the most recent, discarding any old connection information. When CES values are assigned, the path manager ensures that the sequence does not go above the current TOD clock value; therefore, the value could possibly overflow in 143 years from the base TOD clock time (same as TOD clock overflow). Because the low end determines the CES, protocols depend upon which end initiates a connection. It should be noted that even though a line may be leased, no assumption is made that a particular node is at the other end until it identifies itself via the connection protocol. To keep the CES consistent throughout the network, it is necessary that all systems using the path manager use Greenwich Mean Time (GMT) settings in their TOD clocks, and that these clocks are properly synchronized. This also affects the accuracy of the reader start time and other SMF information.

Full Primary Trunk Protocol (Low End Initiation)

This signon protocol is used on BSC and TCP/IP connections only. Figure 29 on page 172 illustrates low end-initiated full connection protocol. Note that the low

end cannot concur with a primary connection; that is the responsibility of the high end.
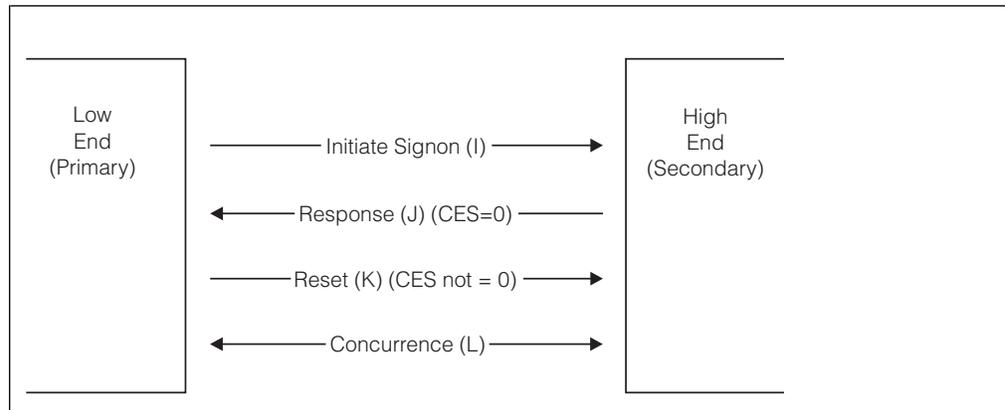
Figure 29. Normal Connection (Low Initiates)

Full Primary Trunk Protocol (High End Initiation)

The high end-initiated connection permits a slightly abbreviated protocol. This signon protocol is always used on SNA sessions but can also be used on BSC or TCP/IP connections.   Figure 30 illustrates high end-initiated full connection protocol.
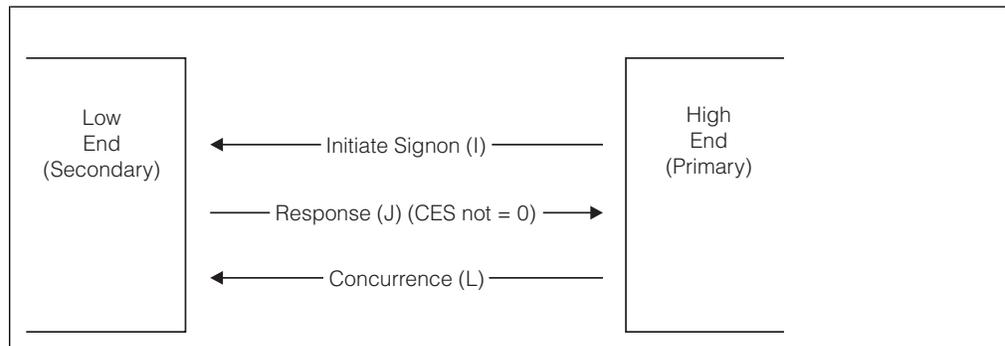
Figure 30. Normal Connection (High Initiates)

Full Secondary Trunk Protocol

A secondary trunk is a line directly connecting two systems already directly connected when the secondary connection is made and the new line resistance is not less that the original. Because this does not represent a new connection, no CES is assigned and no distinction is made between low end and high end. In this case, the multi-trunk flag must be set on in the response (J) record. If the new line resistance is less than the original, this becomes the primary trunk with the CES not equal to zero. Figure 31 on page 173 illustrates secondary full trunk connection protocol.
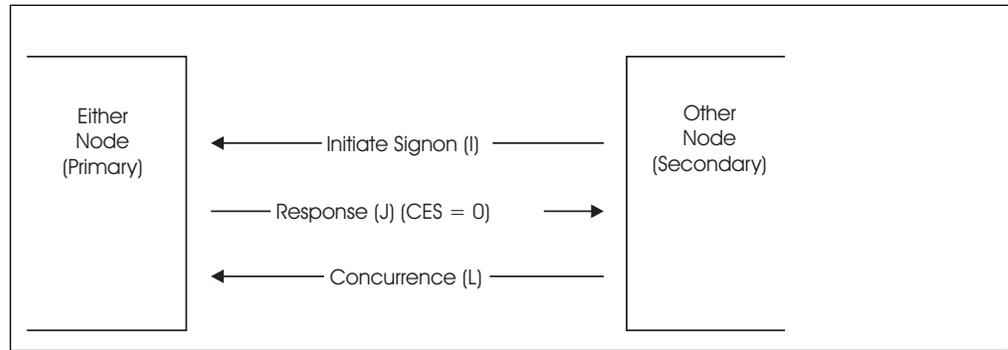
Figure 31. Secondary Trunk Connection

Full Reset Trunk Protocol

If the low end of a connection determines that the primary trunk of a multi-trunk connection is no longer valid, a reset connection protocol is initiated. The trunk over which the reset control record is transmitted is usually the new primary trunk. The CES value will be set to indicate primary or secondary. Other conditions may cause a reset to be initiated from either end; however, the high end must never require the low end to answer the reset. Figure 32 illustrates primary assignment reset protocol.
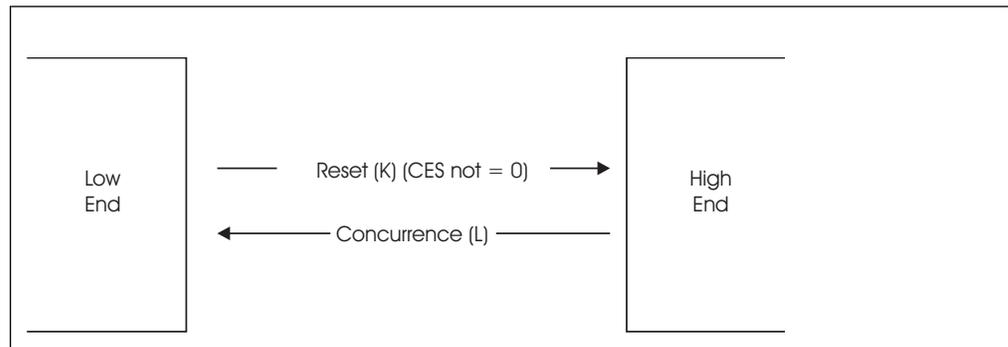
Figure 32. Trunk Reset Protocol

Secure signon Protocol

Establishing a secure NJE signon with NPM: "Secure" signons occur in very much the same way as the connections previously described; however, the line and node passwords are not used. Instead, DES-encrypted signon values are used as follows:

The initial signon ("I") record contains a random 8-byte string (s1) in field NCCIPRAW. The receiving node uses DES-encryption that uses a session key to encrypt this value and returns the value (e_s1) to the initiating node in the response signon ("J") record in field NCCIPENC. The initiating node then uses his session key to DES-encrypt its initial value, and compares the result to the value sent by the secondary node.

Similar verification is done by the secondary node; the secondary node sends its random string (s2) in the field NCCIPENC of the response signon record ("J"); the DES-encrypted value (e_s2) is returned in field NCCCPENC of the reset ("K") or concur ("L") signon record.

Figure 33 illustrates this principle for a high-end initiated, full primary trunk protocol. The signon is similar for a low-end initiated full primary trunk signon, except that the second encrypted value (e_s2) is exchanged in the "K" record, rather than the "L" record.

```
                    I (s1)
         Low    <------------------      High

         End    ------------------>      End
                    J (e_s1, s2)
      (Secondary)                      (Primary)
                <------------------
                    L  (e_s2)
```
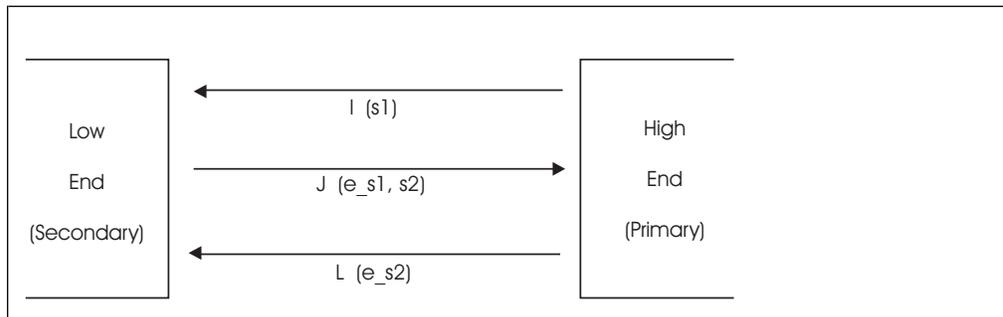
Figure 33. Secure connection (high initiates)

Secure signon Protocol - secondary trunk

For signons using the secure protocol, a minimum of 3 records are required to exchange both random strings and their DES-encrypted value. When using a secure signon, the secondary trunk protocol includes an additional "L" record in order to accommodate all required passwords.

Figure 34 illustrates this principle.

```
                    I (s1)
        Either  <------------------     Either

         End    ------------------>      End
                    J (e_s1, s2)
                    (CES=0)
       (Primary)                      (Secondary)
                <------------------
                    L  (e_s2)
```
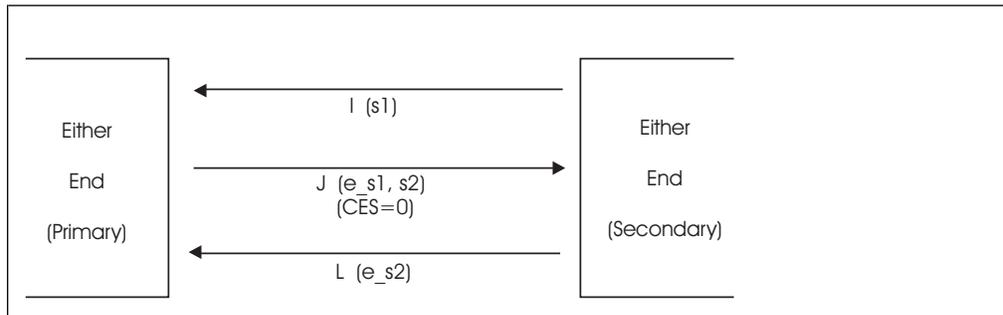
Figure 34. Secure connection (secondary trunk)

Signon Protocols Between A Node With NPM and A Node Without NPM

The protocols used to signon between a node using NPM and a node that doesn't use NPM are the same as those used to signon between nodes that do not use NPM. See "Establishing a Connection Without NPM" on page 170. The node not using NPM discards all NPM records other than B, I or J.

All nodes that do not use the NPM must be defined to each node in the network. Figure 35 on page 175 illustrates the protocols used to connect nodes using the nodal path manager.
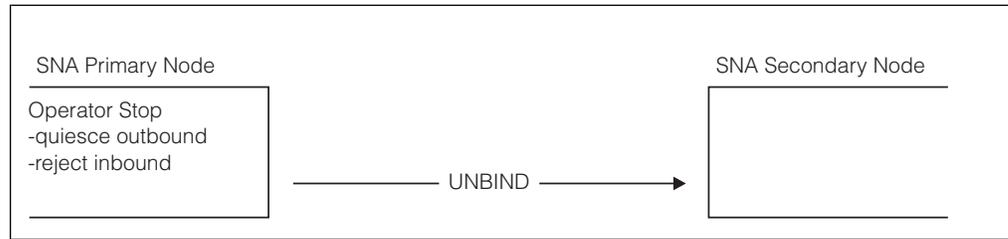
Figure 35. NPM Connection Protocols

Secure signon: For signons using the secure protocol, a minimum of 3 records are required to exchange both random strings and their DES-encrypted value. When using a secure signon, the non-path manager protocol includes an additional "L" record in order to accommodate all required passwords.

Figure 36 illustrates secure connection (non-path manager trunk)
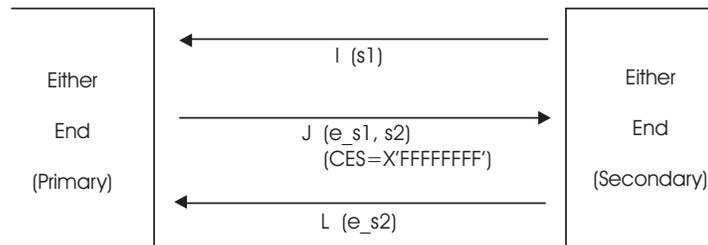
Figure 36. Secure connection (non-path manager trunk)

Connection Status Information

Whenever a dynamic connection is agreed upon, each path manager involved sends an "add connection" control record to systems not involved in the connection over all other NJE lines. The add connection control record is used by receiving path managers to determine best paths to nodes within the network. Each NPM node forwards the add connection control records. If a connection is already known (CES indicates that the control record received is not new), the record is ignored and not forwarded to other network nodes.

Disconnections are promulgated to the members of the network using a "subtract connection" control record. Disconnections may cause nodes formerly reachable via the disconnected line to no longer be available to the system. In this case, dependent connections are automatically determined by each system experiencing the disconnection or receiving the subtract connection control records.

Add and subtract connection control records may be blocked in the buffer with reset and concurrence control records. This is common when a new trunk is established and complete pictures of the network are traded by the systems involved, or when a disconnect is received by a JES. See Networking Connection Control Records (NCCR) in NJE Transfer Units for the format of the control records.

Disconnections

When an NJE line has disconnected, the path manager then clears its own reachable nodes in its tables, validates the queues, and notifies attached nodes of the disconnection.

Terminating a Session or Connection

Normal disconnection of an NJE node (using BSC or CTC) occurs when the initiating system sends a final signoff control record (RCB X'F0', SRCB C'B').

After this record is transmitted, the sending system prepares the line for signon or drains the line, as required. The receiving system should prepare for signon after receiving a signoff. Abnormal disconnection of an NJE node occurs when the initiating system prepares the line for signon or draining the line, as required. Either type of disconnection may occur when all transmitting and receiving functions are idle or when the functions are active.

Normal disconnection of an SNA node uses a TERMSESS if it is the secondary node, or CLSDST if it is the primary node.

Chapter 11. Link Protocols

Bisynchronous Communications Lines

Initialization

BSC connections are established by an exchange of NJE signon records. There is an initiate signon sent by the primary node, followed by a response signon sent by the secondary node. Before the signon records can be exchanged, the two nodes must agree on which one will be primary and which will be secondary. If the nodes cannot agree on which node will be the primary and which will be the secondary, then the nodes are in contention.

BSC connections are initialized in one of the following two ways:

1. The following figure illustrates when there is not any contention between two nodes when signing on.
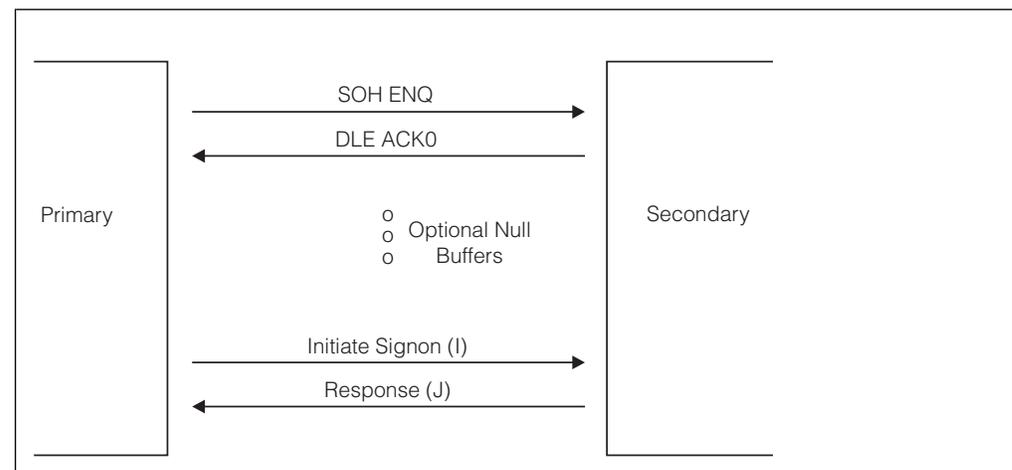
Figure 37. BSC Initialization

The roles may be predefined at each node by the two operators or system programmers such that one end will be secondary and the other primary whenever connection establishment is attempted. In this case, the secondary always begins by reading from the line while the primary begins by writing an SOH ENQ to the line.

Note: Only JES2 and RSCS provide the facility to be pre-defined as a secondary node.

2. The other way that the roles may be established is for both nodes to begin as primary, both sending SOH ENQ. The contention for primary is resolved by the collision of SOH ENQs in one of the two directly-connected communication controllers (CCs).
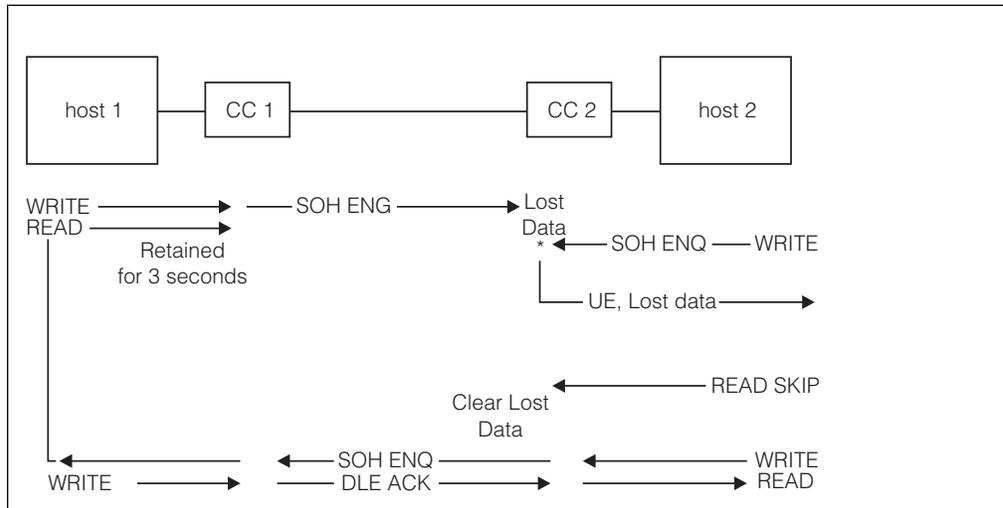
Figure 38. BSC Signon Contention

If inbound data arrives at a communication controller (CC) and a READ is not active, a lost data condition is set in the controller. If a WRITE is subsequently issued, the CC responds with Unit Exception, Lost Data. The lost data condition remains and can only be cleared with a READ command. READ SKIP is issued to clear the lost data condition and another WRITE SOH ENQ is issued. At side 1, the READ stays up for 3 seconds and is waiting for the SOH ENQ to come in. Note that the side that first sends an SOH ENQ to the other controller and causes the collision becomes the secondary.

With a satellite connection between the two communication controllers there is no longer a direct connection between them and it is possible for the two SOH ENQ to "pass in the night". Without the collision detection in one of the communication controllers, both SOH ENQs get through and both nodes believe they are secondary. Each waits for an initial signon. Because of this possibility, the primary/secondary roles must be pre-defined.

Table 61 shows the channel commands used in BSC initialization.

Table 61. BSC Channel Commands

Opcode	Definition	Comments
X'2F'	Disable	This CCW causes the adapter to be reset. If this is a switched communications line, the circuit will be terminated.
X'23'	Set Mode	Conditions for this adapter. For NJE, a byte of X'00' should be used.
X'27'	Enable	Causes a switched line to wait for a connection to be made. On a non-switched line, this will immediately terminate with normal ending status if the communications equipment is operational. If the line is not operational, a unit check with sense data of timeout or intervention required will occur. It also initializes the adapter to accept read and write commands.
X'01'	Write	Writes data to the adapter.
X'02'	Read	Reads data from adapter.
X'06'	Prepare	Causes the adapter to wait for synchronization.

Table 61. BSC Channel Commands (continued)

Opcode	Definition	Comments
X'29'	Dial	Enables the adapter and automatically issues a call to the receiver.

Initialization Error Recovery

Initialization error recovery is in effect until either the DLE ACK0 is received properly from the other side, or an SOH ENQ is properly received and a DLE ACK0 is transmitted. During initialization, errors are responded to in one of three ways:

1. Terminate the line.
2. Retransmit the SOH ENQ.
3. Issue a read with a large CCW count and retransmit the SOH ENQ.

An ending condition of unit exception on the write CCW occurs when data has been transmitted by the other side and a read was not active to receive the data. In this case, a read CCW with a large count and the skip bit should be issued to clear any data that is pending in the adapter. Then, the write should be reissued.

An ending condition of unit check with sense data of either command reject or intervention required should cause the line to be terminated. If a command reject occurred, either the adapter was not enabled, or an invalid CCW was executed. In either case, a software error has probably occurred and retrying would most likely result in a loop.

An intervention required can occur for many reasons. The problems can range from the telephone hanging up on a switched line, to an intermittent clock within the local modem. In any case, retrying the write will usually result in the intervention required happening again. If a unit check with sense data of time out, data check, or lost data is received, the SOH ENQ should be retransmitted. If successful status is received, but the data is not SOH ENQ or DLE ACK0, the SOH ENQ should be retransmitted. These conditions are summarized in the table below.

Table 62. BSC Initialization error recovery

Unrecognized data	CCW and Sense data	Rewrite SOH ENQ
Device Status	*CCW*	*Recovery Action*
Unit	Write	Read Skip - Rewrite SOH ENQ ENQ
Exception	Read	Rewrite SOH ENQ
Device Status	*Sense Data*	*Recovery Action*
Unit Check	80 Command reject	Terminate
Unit Check	40 Intervention required	Terminate
Unit Check	20 Bus out check	Terminate
Unit Check	10 Equipment check	Terminate
Unit Check	08 Data check	Rewrite SOH ENQ
Unit Check	04 Overrun	Rewrite SOH ENQ
Unit Check	02 Lost data	Rewrite SOH ENQ
Unit Check	01 Time out	Rewrite SOH ENQ

Error Recovery

Post-initialization error recovery is more complex than error recovery during initialization. Those errors that require the line to be terminated do not change. Other errors require a negative acknowledgment (NAK (X'3D')) to be transmitted. The NAK notifies the other side that either a block was received incorrectly, or no block was received within the three seconds required by the BSC adapter. Reception of a NAK is also considered an error and requires the last transmission that was not a NAK transmission to be resent. The error conditions and recovery actions are shown in the table below. For JES3 processing acceptions, see A-19.

Table 63. BSC Error Recovery

SOH ENQ	CCW and Sense data	Restart line
NAK		Retransmit last non-NAK transmission
Unrecognized data		Write NAK
Device Status	*CCW*	*Recovery Action*
Unit	Write	Read Skip - Write NAK
Exception	Read	Write NAK
Device Status	*Sense data*	*Recovery Action*
Unit Check	80 Command reject	Terminate
Unit Check	40 Intervention required	Terminate
Unit Check	20 Bus out check	Terminate
Unit Check	10 Equipment check	Terminate
Unit Check	08 Data check	Write NAK
Unit Check	04 Overrun	Write NAK
Unit Check	02 Lost data	Write NAK
Unit Check	01 Time out	Write NAK

Termination

Normal termination occurs when a signoff record is sent. There is no special BSC character sequence sent for termination and no response is expected to the signoff record. If a read is chained to the write of the signoff, the read will time out. After the signoff record is written, a disable should be issued to disable the communications adapter.

Normal Sequences

DLE ACK0 (X'1070') is a positive acknowledgment to the previous block. At the line protocol level, a DLE STX (X'1002') is also a positive acknowledgment to the previous block. The DLE ETB that signals the end of the transmission block must be sent by command chaining a separate write CCW for those two bytes of data to the write CCW used to transmit the transmission block. Otherwise, a command reject will occur.
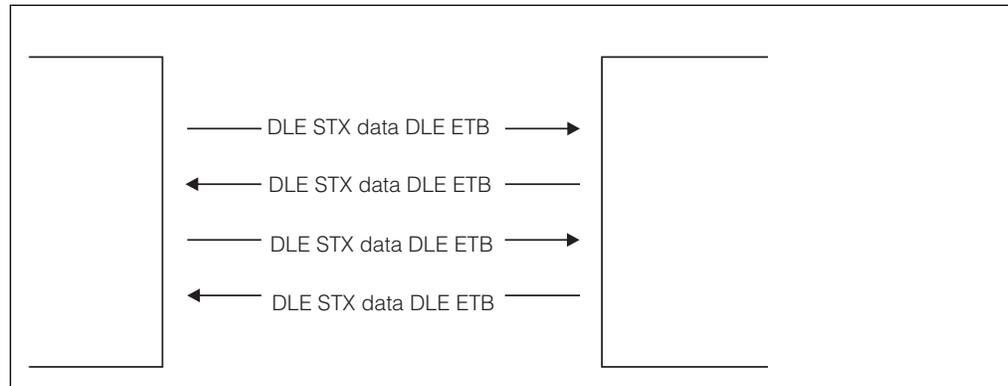
Figure 39. BSC Normal Communication

V.27 Modem Contention Resolution

Certain modems used in support of CCITT V.27 protocols for BSC communications require special signon contention resolution protocols.

These protocols require that, external to the NJE protocols, there be a way to identify to the program supporting this new protocol, the "mode" of the node that is about to participate in a connection. There are three modes:

- Primary
- Secondary
- Contention

For V.27 signon, a given node must be identified relative to another specific node as either "primary" or "secondary". The other node must have the opposite designation. This information need not be available to the programs supporting the protocols except as a parameter on invocation. Hence, two system operators may establish an arbitrary relationship prior to session initialization, or the product may elect to maintain tables.

V.27 Signon by the Primary Node

This scenario assumes that the node attempting signon has been identified as the primary node.

```
START:
        Allow operator intervention to stop this
        process, otherwise continue until 'DLE ACK0' is received.

        Issue   WRITE (cc)    SOH ENQ
               READ

        If ANY SERIOUS ERROR (e.g. device not available)
              -Terminate immediately
        If UNIT EXCEPTION (contention)
              -Issue message
              -Terminate  (contention not allowed)
        If TIMEOUT
              -Retry immediately the same CCW chain up to 5 times
              -If retry works, go to step CKREAD
              -Otherwise, set timer for 2 minutes
              -When timer expires, retry the original CCW
                 sequence up to 5 times

CKREAD: (Successful I/O completion)
        See if data read is 'DLE ACK0'
        If so, send 'I' signon record
         ... normal protocols follow
        If not 'DLE ACK0', go back to START
```

Figure 40. V.27 Modem Protocols: Primary Node Signon

V.27 Signon by the Secondary Node

This scenario assumes that the node attempting signon has been identified as the
secondary node.

```
START:
        Allow operator intervention to stop this
        process, otherwise continue until 'SOH ENQ' is received.

        Issue   PREPARE (cc)
               READ
        If ANY SERIOUS ERROR (e.g. device not available)
              -Terminate immediately

CKREAD: (Successful I/O completion)
        See if data read is 'SOH ENQ'
        If so, SEND 'DLE ACK0'
               READ
         ... normal protocols follow
        If not 'SOH ENQ', go back to START
```

Figure 41. V.27 Modem Protocols: Secondary Node Signon

Channel-to-Channel Adapter

NJE also supports communication by a channel-to-channel (CTC) adapter as a
communication line. The support is for 360-mode operations over block
multiplexor channels.

Note: POWER supports only virtual channel-to-channel adapters.

To properly operate as a NJE device, the UCWs for the CTC adapter should be plugged non-shared. (This is a hardware option on block multiplexor channels.) The UCW is used to store the address for performing the channel reconnection. If the UCW for the CTC adapter is not plugged non-shared, the CTC adapter will not disconnect while waiting for the control CCW to complete. This will cause the channel to remain busy for a significant length of time during each I/O operation.

Initialization

To begin communication, a SENSE device command (X'14') is issued by side A to determine the state of side B of the adapter. If a byte of X'00' is returned, it indicates that side B has not yet been activated and no command is active. The correct response for side A is to issue a control command (X'07') chained to a read (X'02'). This serves two purposes. First, the control command signals an attention to side B. Second, the control state remains in side A so that when side B does subsequently start, it will detect that side A is already active. The control CCW causes a channel disconnect on a block multiplexor channel, so the channel is available for use with other devices while waiting for side B of the CTC adapter to respond.

When side B starts, it issues a sense device command, detects the control state (X'07') on side A, and writes a SYN NAK (X'323D'). This allows side A to become the primary, which then responds with a SOH ENQ (X'012D'). If side B wishes to be the primary rather than the secondary, it must send a SOH ENQ rather than a SYN NAK. The secondary then responds with a DLE ACK0 (X'1070'), as with a BSC line. (See CTC Adapter Initialization in Appendix A for JES3-specific processing; see RSCS in Appendix A for RSCS-specific processing.)
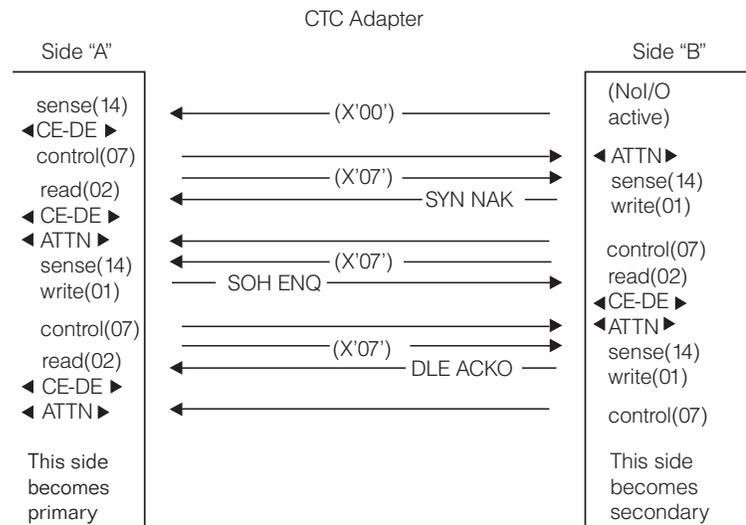
CTC Adapter

Side "A"

sense(14)
◄CE-DE ►
control(07)
read(02)
◄CE-DE ►
◄ATTN ►
sense(14)
write(01)

control(07)
read(02)
◄CE-DE ►
◄ATTN ►

This side
becomes
primary

(X'00')

(X'07') — SYN NAK

(X'07') — SOH ENQ

(X'07') — DLE ACKO

Side "B"

(NoI/O
active)

◄ATTN►
sense(14)
write(01)

control(07)
read(02)
◄CE-DE ►
◄ATTN►
sense(14)
write(01)

control(07)

This side
becomes
secondary

Note: CE-DE is a channel end, device end status in the CSW.
      ATTN is attention status in the CSW.

Figure 42. CTC Adapter Initialization

Error Recovery

No error recovery is used on the CTC adapter. When running in compatibility mode, the CTC adapter cannot generate any status bits other than attention, busy, channel end, or device end. If any channel errors occur, it is considered proper to

signal connection termination to the higher levels. The session must then be restarted by normal initialization protocol.

Prior to the advent of channels that buffer status, attention and busy could only occur if the two systems had lost synchronization. A non-shared 30xx subchannel may cause an attention and/or busy status. This is because the subchannel has already buffered the attention from the other side's I/O request and intends to present it. In channels that do not buffer status, the I/O would have been sent to the CTC adapter, which would have suppressed the attention status and accepted the command. The only option in this case is to reissue the I/O.

On newer, buffered channels, a condition exists where the attention busy status is caused by having another device cause contention. Therefore, the attention busy may occur in normal operation and should not be a termination condition. (See CTC Adapter Attention and Busy Status in Appendix A for system-specific handling of this condition.)

Termination

Normal termination occurs when a signoff record is sent. There is no special character sequence sent for termination and no response is expected after a signoff record is sent. In this case, the standard CTC program should terminate with the write CCW. (See CTC Adapter Termination Deviations in Appendix A for JES3 deviations.)

Normal Sequences

Instead of using CCWs designed for BSC line transmissions, the channel-to-channel adapter sequences are as those shown in Table 64.

Table 64. CTC Adapter Program

Opcode	Definition	Comments
X'14'	Sense	This CCW will cause other end of adapter control to fall through to the read CCW.
X'01'	Write	Writes data to the adapter.
X'07'	Control	Causes the channel program to wait until the other end of the adapter issues a SENSE. A block multiplex channel is freed for other programs.
X'02'	Read	Reads data from the adapter.

```
                              CTC Adapter

   read(02)                                              sense(14)
 ◄ CE-DE ►  ◄──────── DLE STX data ──────────            write(01)
 ◄ ATTN ►  ◄────────────────────────────────
   sense(14)                                             control(07)
   write(01)  ──────── DLE STX data ─────────►           read(02)
   control(07) ───────────────────────────────►        ◄ CE-DE ►
   read(02)                                             ◄ ATTN ►
 ◄ CE-DE ►  ◄──────── DLE STX data ──────────            sense(14)
 ◄ ATTN ►  ◄────────────────────────────────            write(01)
                                                         control(07)


   Note: CE-DE is a channel end, device end status in the CSW.
         ATTN is attention status in the CSW.
```
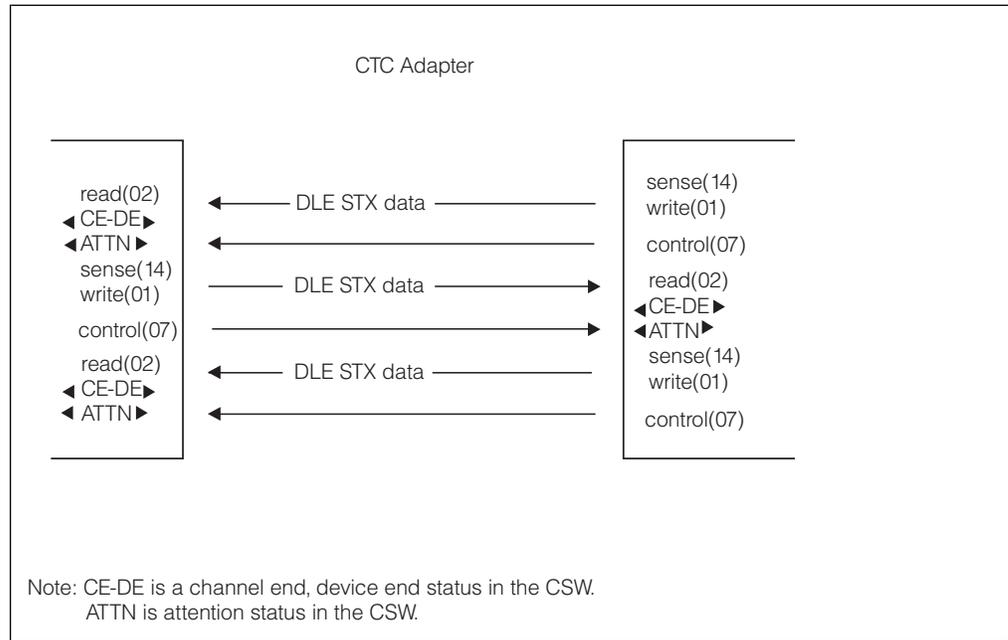
Figure 43. CTC Adapter Normal Sequences

PREPARE Mode Option: Suspend I/O (BSC and CTC Adapter)

This option (implemented only by RSCS Version 2) defines a method by which non-SNA links do not have to transmit to each other every one to two seconds. It is called "PREPARE mode" because it uses the PREPARE CCW on a BSC line. For the CTC adapters, the link is left without any I/O active during periods of inactivity.

Use of PREPARE mode is controlled by a signon concurrence bit. The signon concurrence feature is an optional element of the signon protocol. It allows two systems to determine that each are able to work in an extended mode on the communications line running between them. This causes no problems for back-level systems because this has already been handled by the signon concurrence protocols. The result is that PREPARE will never be used when communicating with back-level systems.

PREPARE could cause idle lines to appear connected when, in fact, the system at the other end of the line is no longer operational. This problem may be avoided by setting a timer during I/O suspension and attempting to wake up the other side after a fixed interval has passed. If the other side does not respond, the link must be taken down using normal error recovery procedures.

For PREPARE mode to work properly, both systems using it must send null buffers rather than DLE ACK0 when they have no data to transmit. Signon concurrence bit X'80' controls the use of PREPARE.

This requires no control block changes, but does rely on the signon concurrence feature of NJE. The X'80' bit in the first byte has been assigned to PREPARE.

Normal Sequences

Presently, when no data is being transmitted or received on a given link, a null-buffer is transmitted every one to two seconds. This is shown for BSC below. (The CTC adapter normal CCW chain is similar, except that the read CCW is preceded by a control command and the write by a sense.)
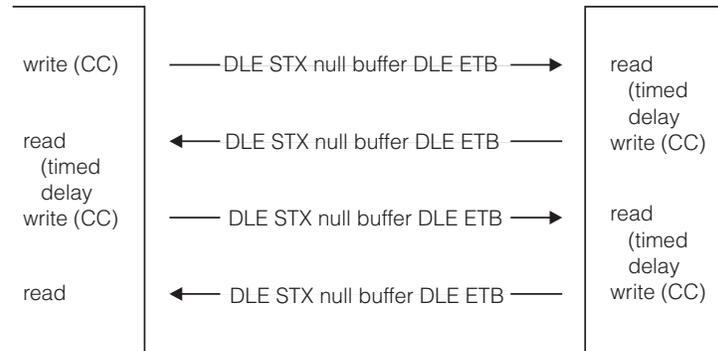
Note: (CC) indicates command chaining.

Figure 44. BSC Normal Communication

Requesting I/O Suspension

When the last non-null buffer is acknowledged and no files are in the process of being received or sent, one side (side A) may decide to suspend I/O activity. It does this by writing a null buffer to the other side (side B) with the terminating sequence DLE ETX instead of the normal DLE ETB. This buffer is sent with the appropriate I/O CCW sequence for CTC adapters or BSC. Thus, the write is command chained either directly (BSC) or indirectly (through a control for CTC adapters) to a read. If side B wants to accept the I/O suspension, it responds with another null buffer also ended by DLE ETX instead of DLE ETB. Unlike the normal I/O sequence, this write CCW is not chained directly to a read (BSC) or to a control, read (CTC adapter). If it does not want to suspend I/O, side B sends either a non-null data buffer or a null buffer terminated with DLE ETB with its normal I/O sequence. In this case, normal I/O sequences continue to be used by both sides.

As soon as side B sends the acknowledging null buffer, it can suspend I/O. Side A must also suspend I/O if it receives an acknowledging null buffer (with DLE ETX) immediately following the buffer it used to request suspension.

Note: The two null buffers with DLE ETX must be exchanged without an intervening buffer with DLE ETB in order for PREPARE mode to be used. If there is an intervening normal buffer, the second DLE ETX buffer will be considered a new request rather than an acknowledgment.

Suspension and Resumption of I/O

How communication is actually suspended and resumed with PREPARE mode differs for BSC and CTC adapters. This is described in the following separate sections.

BSC I/O Suspension

For BSC suspension, both sides issue a read CCW that is allowed to time out followed by the I/O sequence of a PREPARE CCW (X'06') command chained to a read CCW.

Both systems then wait until either one of them has something to send. When either of the systems has data to transmit to the other system, it must then issue a halt device (HDV) or halt I/O (HIO) to terminate the PREPARE. Then, before transmitting the actual buffer, it should transmit a DLE ENQ sequence. When a system waiting on a PREPARE receives any data, it must acknowledge the data with a null buffer before going back to wait on the PREPARE. This allows the system with the pending data to transmit the data and resume normal processing once data has been received. This sequence is shown in BSC Communication in PREPARE Mode.

If both systems attempt to initiate transmission at the same time, one of the DLE ENQs may be lost by the HDV, or may be lost when a read-skip is used to recover from a unit exception on the write. In this case, a contention resolution protocol is used (see "BSC Error Protocols" on page 190) to force resynchronization.

Note: During resumption, all null buffers are terminated with the normal DLE ETB sequence.

```
Requesting

write (CC)              DLE STX null buffer DLE ETX          read

                                                             Agreeing to
                                                             suspend

read                    DLE STX null buffer DLE ETX          write (CC)

                                                             read
                                                             timeout

read
(timeout)

PREPARE (CC)            read pending on both sides           PREPARE (CC)
                                  ...
                             No I/O Activity
                                  ...
data
arrives
cancel read

HDV Wakeup
write (CC)              (20 SYN) DLE ENQ                     read

read                   DLE STX null buffer DLE ETB           write (CC)

write (CC)             DLE STX null buffer DLE ETB           read


Note: (CC) indicates command chaining.
      (20SYN) indicates 20 SYN characters.
```

Figure 45. BSC Communication in PREPARE Mode

CTC Adapter I/O Suspension

During I/O suspension on a CTC adapter, no I/O sequences are issued so that the adapter is left with no I/O in progress during idle periods. A control CCW, which signals attention, indicates data transmission is to be resumed. The procedure, illustrated in CTC Adapter Communication in PREPARE Mode in Protocols Common to BSC and CTC Adapters, is as follows:

- I/O suspension is requested and agreed upon as described in "Requesting I/O Suspension" on page 186.
- Both sides then wait without any I/O outstanding.
- If either side wants to resume, it must first issue a control CCW which will signal an attention to the other side (B, for this example). The control will be command chained to a read.
- When the attention interrupt is detected by side B, it will issue its normal CTC adapter CCW chain (sense, write (null buffer), control, read).
- The sense causes the control issued by side A to complete, and the null buffer is written by side B and read by side A.
- Side A, which had data to send, can send the data by issuing the normal CCW chain as soon as the first "wake up" read completes.
- If both sides attempt to "wake up" at the same time, one of the control CCWs will get a busy response. When this happens, the normal CCW chain should be issued with the write for a null buffer. Normal synchronized operation will then commence.
- If one side has gone down during the PREPARE, the "wake up" control CCW will not complete and a selector channel will be tied up and unavailable for use by other devices. To prevent this, a timer could be set (for less than 30 seconds) at the beginning of I/O resumption. If the timer expired without the I/O completing, an HDV would be issued and the link taken down.

The following flows show the procedure graphically. In the diagrams:
- S = Sense CCW
- C = Control CCW
- R = Read CCW
- W = Write CCW
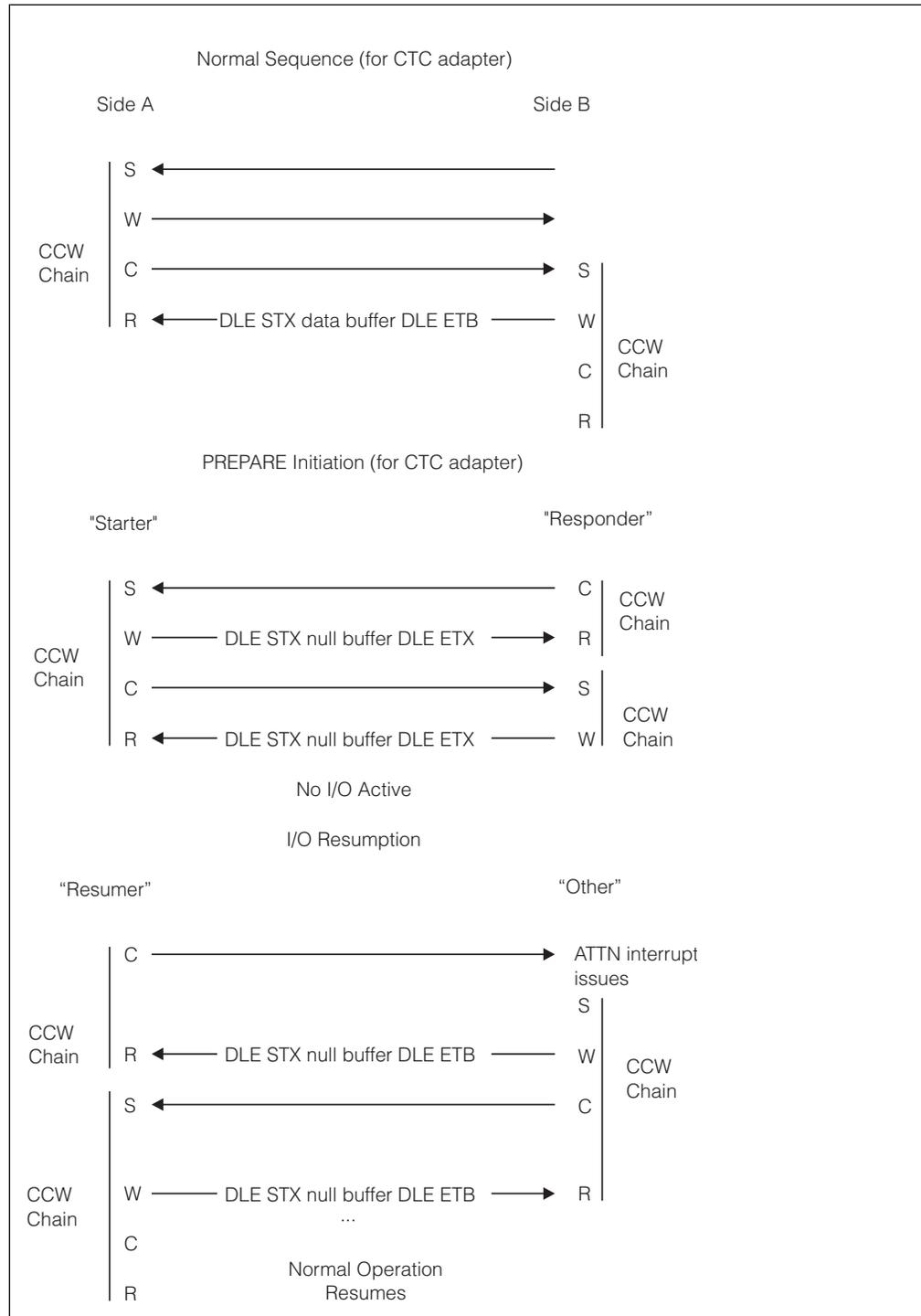
Figure 46. CTC Adapter Communication Procedure Shown Graphically

Protocols Common to BSC and CTC Adapters

During the I/O suspension period, either side may optionally set a timer for some reasonable interval (5 to 15 minutes) and, when the time expires, transmit a null buffer to determine if the other system is still operational. If the other side does not respond, the link is taken down by normal error recovery procedures.

Sense	◄──────────────────────────	Control
Requests PREPARE Write	── DLE STX null buffer DLE ETX ──►	Read
Control	──────────────────────────►	Sense Agrees to suspend I/O
Read	◄── DLE STX null buffer DLE ETX ──	Write
	No Activity	
Data Arrives (Wakeup) Control	──────────────────────────►	ATTN Sense
Read	◄── DLE STX null buffer DLE ETB ──	Write
Sense	◄──────────────────────────	Control
Write	── DLE STX null buffer DLE ETB ──►	Read
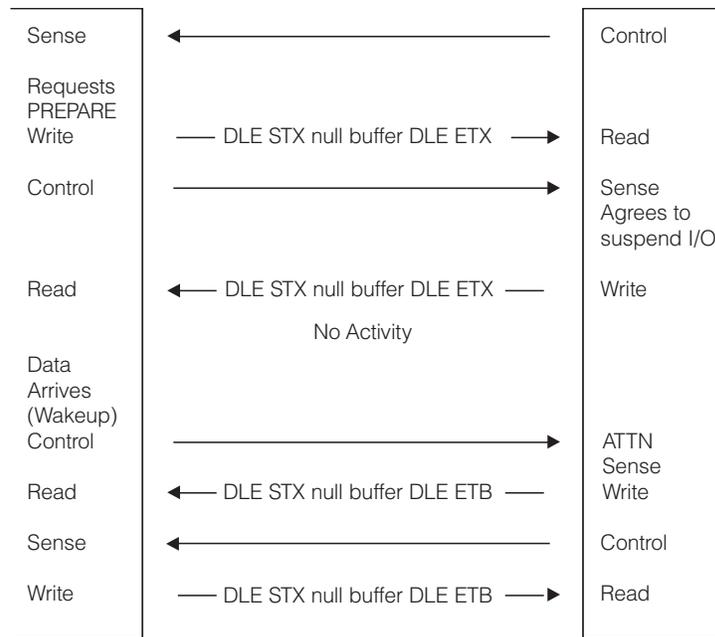
Note: (CC) indicates command chaining.

Figure 47. CTC Adapter Communication in PREPARE Mode

BSC Error Protocols

The symbols used in the flows that follow are:

- R = Read
- P = PREPARE
- W = Write
- S = Sense
- C = Control
- TO = Time Out
- cc = Command Chain

1. When a DLE ENQ for PREPARE resume receives a negative acknowledge, the condition is treated as a NAK to a NAK in normal processing. In this case, the last non-NAK is defined to be the last buffer (null or not) for which an acknowledgement has not been received. This buffer will be different depending on which side starts the PREPARE and which receives the NAK. The following example illustrates the different interpretations.

   Note: In the example, an actual buffer is sent in all cases where just the BCB is shown.
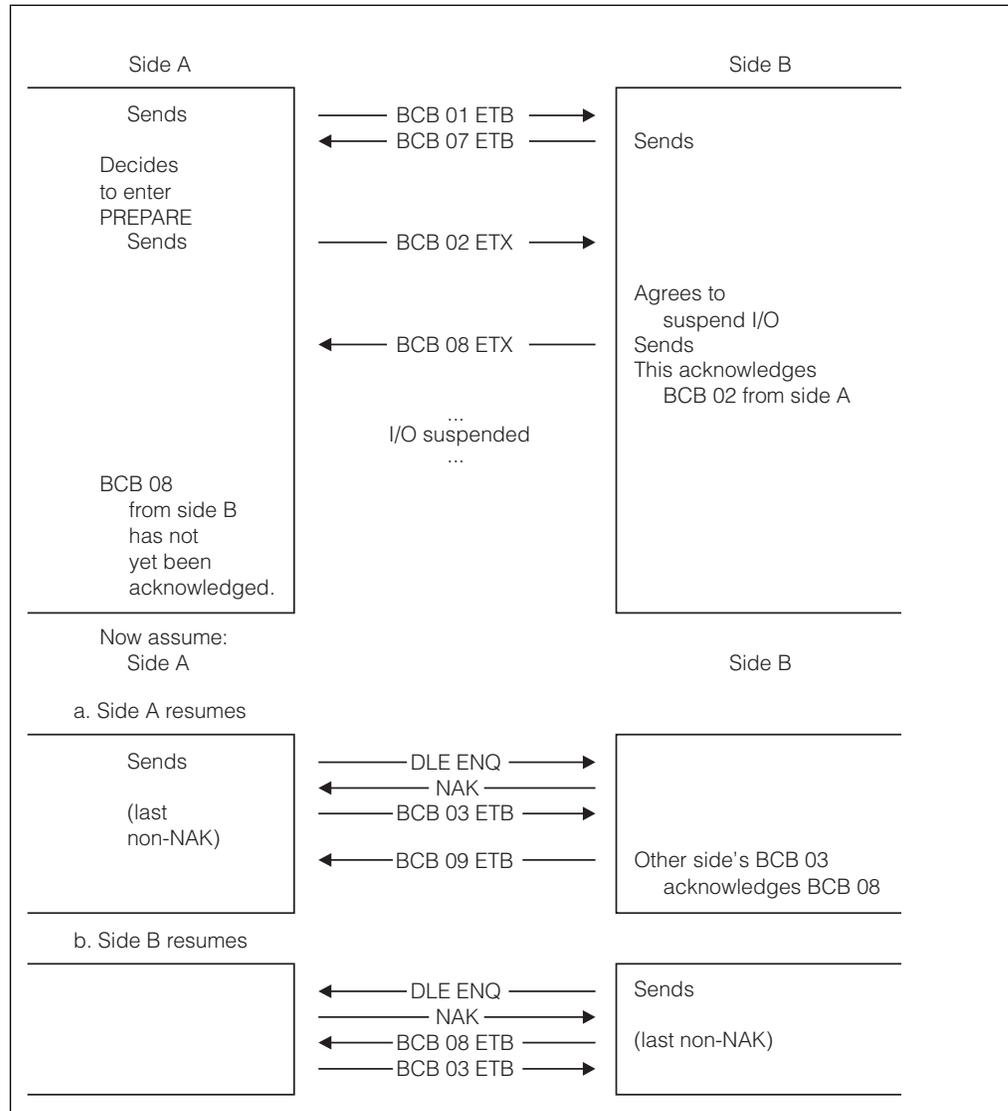
Figure 48. BSC Error Protocols

2. In the case of a wake up contention condition on a BSC line, one DLE ENQ
   may be lost. This situation is resolved by using the primary/secondary
   relationship between the nodes (established at signon) to determine which side
   sends data first.

   If one side receives a unit exception when it tries to write the DLE ENQ, that
   side uses the primary/secondary relationship in the same way as above after
   issuing a read skip to clear the contention.

   If one side receives DLE ENQ instead of data in response to its DLE ENQ, then
   that side waits (issues a read CCW) for DLE STX data buffer DLE ETB to be
   transmitted if it is the secondary, or sends DLE STX data buffer DLE ETB if it is
   the primary.

   The flow for contention is:

```
┌──────────────────────────────────────────────────────────────────────┐
│  Side A                                     Side B                     │
│  (Primary)                                  (Secondary)                │
│ ┌──────────┐                              ┌──────────┐                 │
│ │  P(cc)   │                              │  P(cc)   │                 │
│ │          │                              │          │                 │
│ │  R       │                              │  R       │                 │
│ │          │                              │          │                 │
│ │  HDV     │                              │  HDV     │                 │
│ │          │                              │          │                 │
│ │  W(cc)   │ ──── DLE ENQ ──▶  ◀── DLE ENQ ──── │  W(cc)  │           │
│ │          │                              │          │                 │
│ │  R       │ ◀───────────────────────────── │  R     │                │
│ │          │                              │          │                 │
│ │  W(cc)   │ ◀── DLE STX data buffer DLE ETB ──── │  R  │             │
│ │          │                              │          │                 │
│ │  R       │ ◀── DLE STX data buffer DLE ETB ──── │  W(cc) │          │
│ │          │           null buffer        │          │                 │
│ │  W       │ ◀──────────── or ─────────── │  R      │                 │
│ │          │           dat buffer         │          │                 │
│ └──────────┘                              └──────────┘                 │
└──────────────────────────────────────────────────────────────────────┘
```
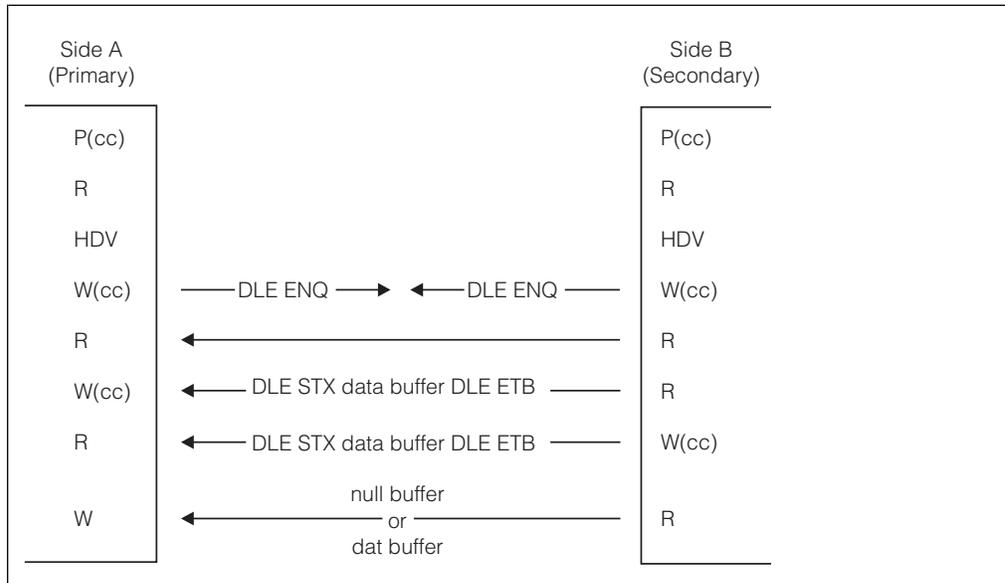
3. If one side goes down while I/O is suspended, the other side will detect it at resume time by a timeout as follows:

```
┌──────────────────────────────────────────────────────────────────────┐
│       Side A                             Side B                        │
│  ┌──────────────┐                   ┌──────────────┐                   │
│  │       P(cc)  │                   │  P(cc)       │                   │
│  │              │                   │              │                   │
│  │       R      │                   │  R           │                   │
│  │              │                   │  goes down   │                   │
│  │ Wake up      │                   │              │                   │
│  │ ──────▶ HDV  │  ─ DLE ENQ ▶      │              │                   │
│  │              │                   │              │                   │
│  │       W(cc)  │                   │              │                   │
│  │              │  ◀── Time out     │              │                   │
│  │       R      │                   │              │                   │
│  └──────────────┘                   └──────────────┘                   │
└──────────────────────────────────────────────────────────────────────┘
```
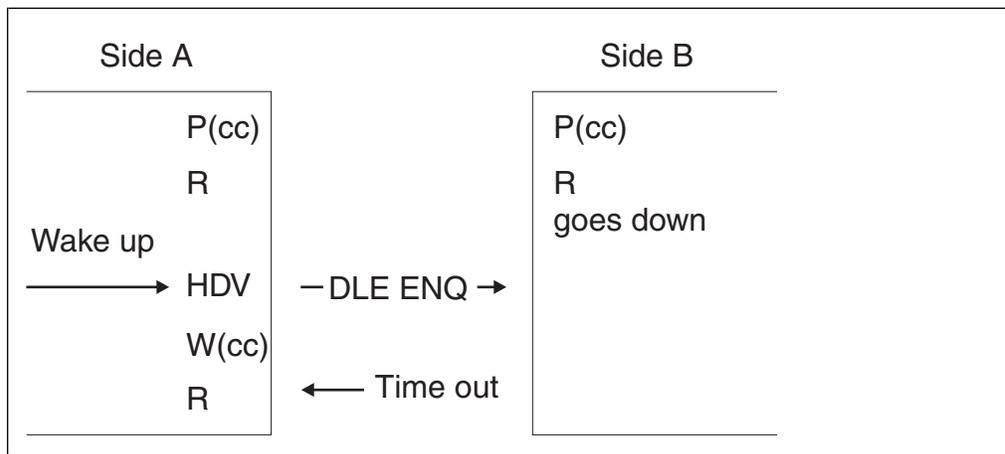
When the read times out in this case, normal time-out processing should be used. This includes sending a NAK.

4. If one side restarts while I/O is suspended, the following scenario will occur:

```
Case A
        Side A                                              Side B
   _____                                   _____
          P(cc)                                           P(cc)

                                                          R

                                                          Restarts
          R                ◄─────SOH  ENQ─────           W
      Detects
          SOH ENQ
          and goes
          to restart
          routine
   _____                                   _____

                          Normal Signon Processing
```
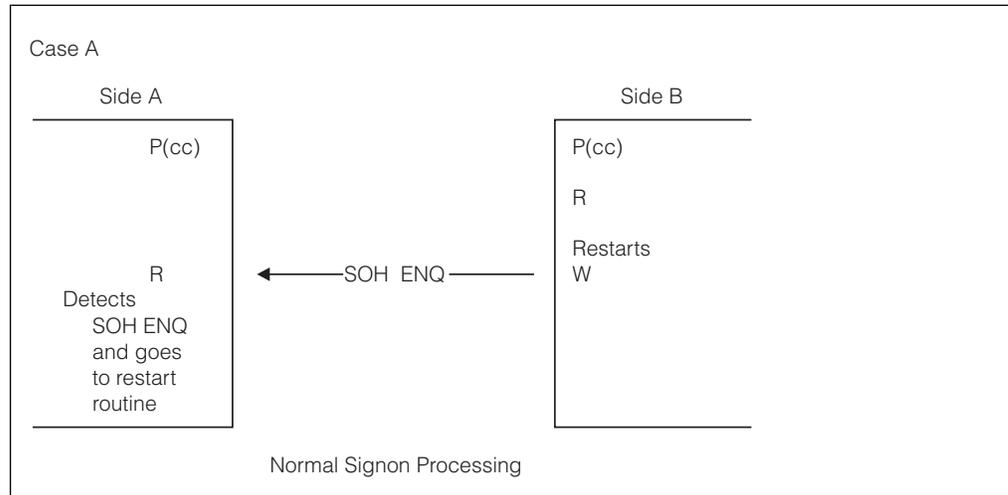
If the restart occurs at the same time as a resume, the following can happen:

```
Case B
        Side A                                              Side B
   _____                                   _____
          P(cc)

          R                                               Restarts

    Restarts  HDV        ◄─────SOH  ENQ─────           W

          W              ─────DLE  ENQ─────►           R    Note that the DLE ENQ
                                                            is not a valid response
                                                            So signon code tries again.

          R              ◄─────SOH  ENQ─────           W    So signon code tries
                                                            again
   _____                                   _____

                  Detects SOH ENQ and goes to restart routine.
```
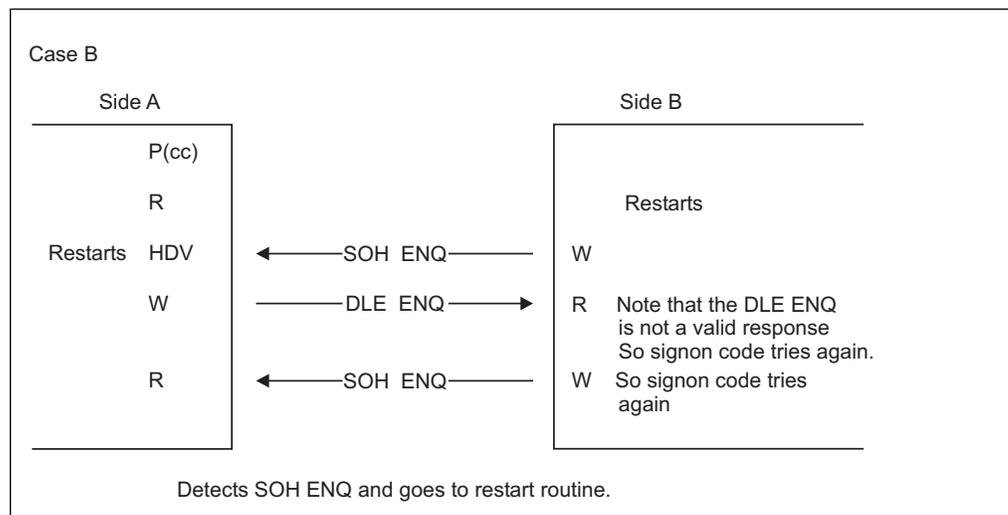
Note: If both sides try to restart while I/O is suspended, the signon contention protocol will resolve contention problems.

5. A system that has not agreed to use PREPARE at signon time should never receive DLE ENQ. However, if such a sequence is ever received unexpectedly, the proper procedure is to respond to it with a NAK.

```
        Side A                                    Side B
        _____                                  _____
              P(cc)

              R

Suspends  HDV
I/O
              W(cc)      ———— DLE  ENQ ————▶    R

              R          ◀———— NAK ————          W(cc)

              W(cc)      ———— null ————▶         R
                              buffer
              R

                          No data is lost
```
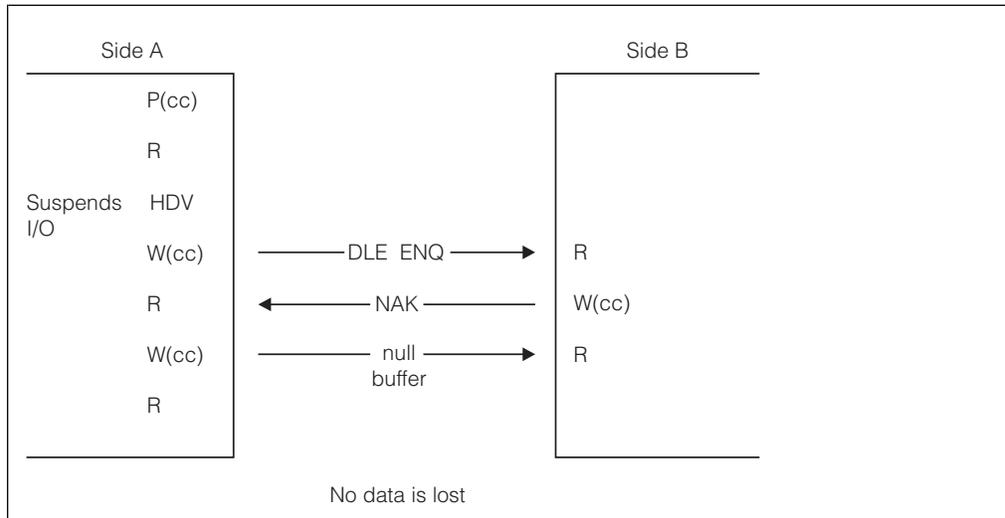
6. If the resuming side sends DLE ENQ and the sequence is lost (due to a
   hardware error), a timeout will occur and that side should send a NAK
   (following normal timeout recovery).

```
        Side A                                    Side B
        _____                                  _____
              P(cc)                               P(cc)

                                                  R

                                                  HDV       Resumes
                                                                I/O
Lost                     ———— DLE ENQ ————▶       W(cc)

                              Time Out  ————▶      R

              R          ◀———— NAK ————            W(cc)

              W(cc)      ———— null ————▶           R
                              buffer
```
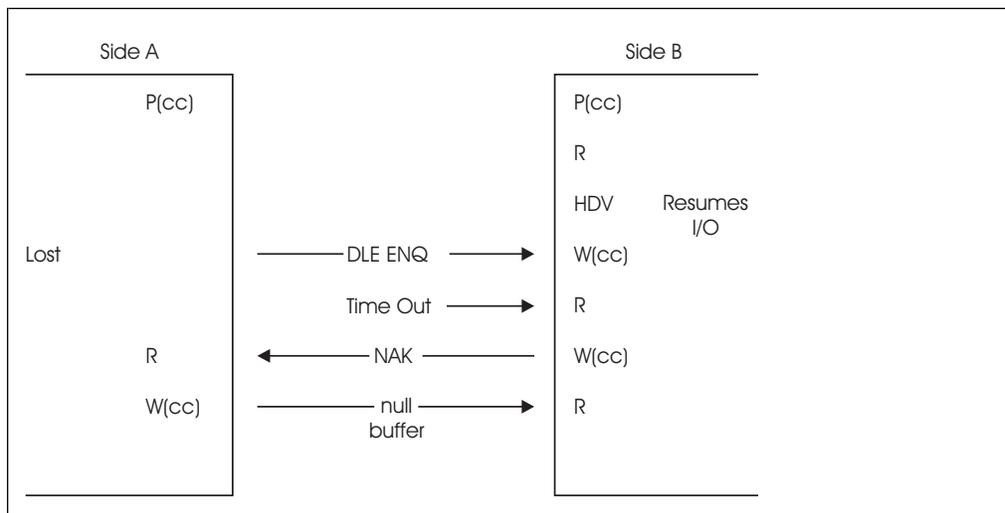
7. Note that the only valid sequences read by the READ chained to a PREPARE
   CCW are DLE ENQ, SOH ENQ, or NAK. All these cases have been covered in
   the preceding examples. The "resumer" never immediately sends a null or
   non-null data buffer, nor is there any way such a buffer can be received until
   normal sequences are used.
8. The following flow shows what happens when side A requests I/O suspension
   and side B does not want to enter this state because it has something to send.

```
       Side A                                      Side B

         R        ◄──────── data buffer ─────────   W(cc)

       W(cc)      ──── null buffer DLE ETX ───►      R

         R        ◄──── data buffer DLE ETB ────    W(cc)

       W(cc)      ────────── null buffer ──────►      R


                    Normal Sequencing Continues
```
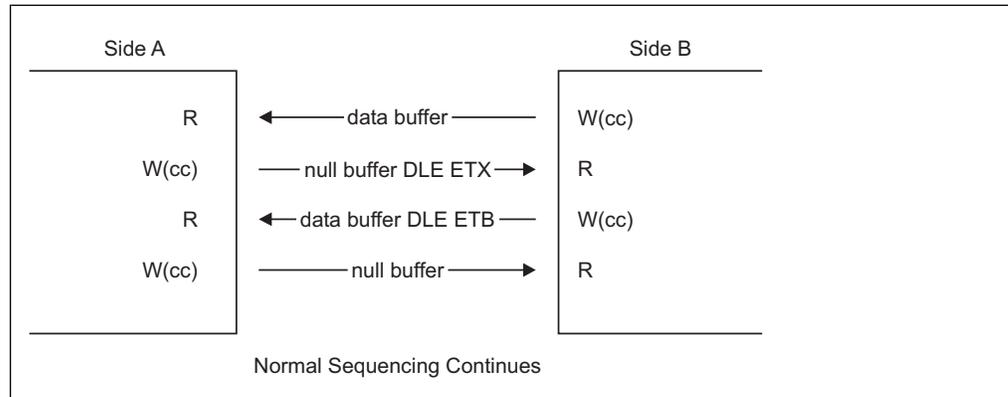
9. Either side may continue to request suspension, even if the other side does not agree to it.

```
            SIDE A                                          SIDE B

      /   W   ──── DLE STX ─null buffer─DLE ETB ───►    R
   (CC)
      \   R   ◄─── DLE STX ─data buffer─DLE ETB ────    W  \
                                                             (CC)
      /   W   ──── DLE STX ─null buffer─DLE ETX ───►    R  /
   (CC)
      \   R   ◄─── DLE STX ─null buffer─DLE ETB ────    W  \
                                                             (CC)
          W   ──── DLE STX ─null buffer─DLE ETX ───►    R  /

          R   ◄─── DLE STX ─null buffer─DLE ETX ────    W  \
    (CC)                                                      (CC)
  Suspend  P                                            P  /
    I/O
          R                                             R


Note: (CC) indicates command chaining
```
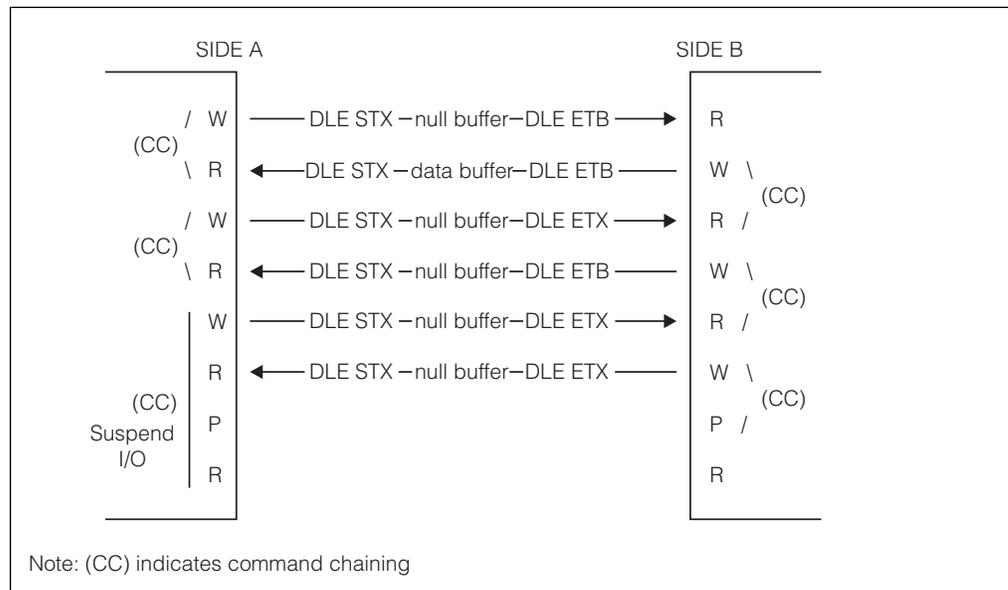
CTC Adapter Error Protocols

The symbols used in the following flows are:

- R = Read
- P = PREPARE
- W = Write
- S = Sense
- C = Control
- TO = Time Out
- cc = Command Chain

1. A NAK is never sent on a CTC adapter; if a NAK is ever received, the link should be taken down.
2. The hardware prevents any contention during I/O resumption.

```
  Side A                          Side B

    C(cc)  ───────────────────►
                                    C(cc)
                                    Busy
  Completes    ◄───────────────     S(cc)
  Control

        R      ◄───────────────     W(cc)

                                    C(cc)

                                    R
```
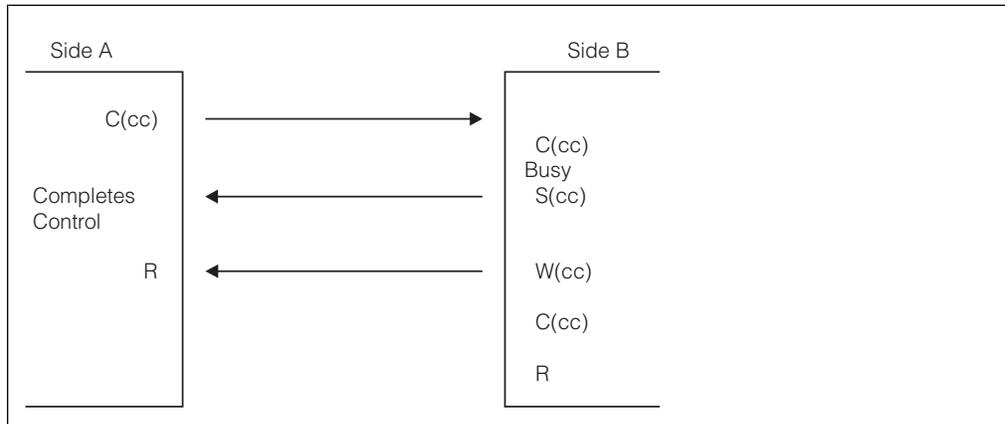
3. If one side goes down while I/O is suspended, the other must detect this during the resume as follows:

```
  Side A                          Side B

                   Both Waiting
                                    Goes Down

  Resumes

        C  ─────────────────────►

  Sets timer.
  If I/O has not completed after
  time expires, issues HDV.
```

4. Restart

```
             Side A              Side B
                 Both waiting without I/O


              Restarts
                 S      ◄──── X'00'
               C(cc)    ──────────────►   ATTN     Thinks this is
                                                   normal I/O
                                                   resumption.
              Completes   ◄────────────   S(cc)
              Control

  Detects the null    R   ◄── null buffer ── W(cc)
  buffer when it
  expects either     S(cc)  ◄───────────   C(cc)
  SYN NAK or SOH ENQ.
  However, it still    W   ─── SOH ENQ ──►   R
  can send SOH ENQ.
                                          ◄──── Detects SOH ENQ and
                                                performs its normal
                                                restart procedure.
```

5. If one side does not want to accept a request to suspend I/O, it responds normally to a null buffer ended by DLE ETX and I/O suspension is not entered

by either side.
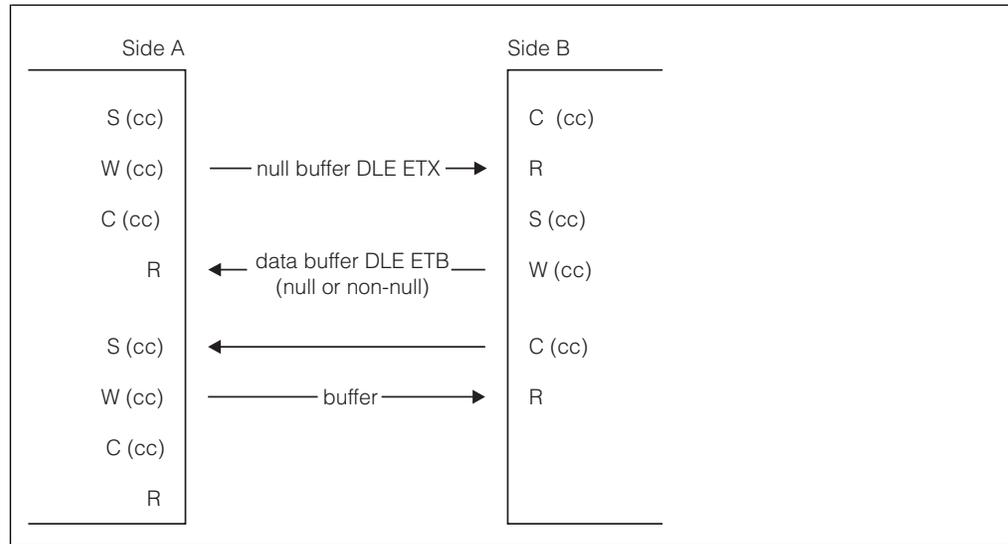
If side A still wants to enter I/O suspension after the other side has said no, it can again send a null buffer ended by DLE ETX. In the above example, the last write (buffer) would be replaced with:

```
W(cc) ---- null buffer DLE ETX --> R
```

If side B responds with null-buffer DLE ETX immediately, suspension will then take place. However, if side B responds with a normal buffer, normal I/O continues.

6. Side A may go on writing null buffer DLE ETX if it still wants to suspend the I/O. Suspension will not take place until side B responds immediately with the null buffer DLE ETX.
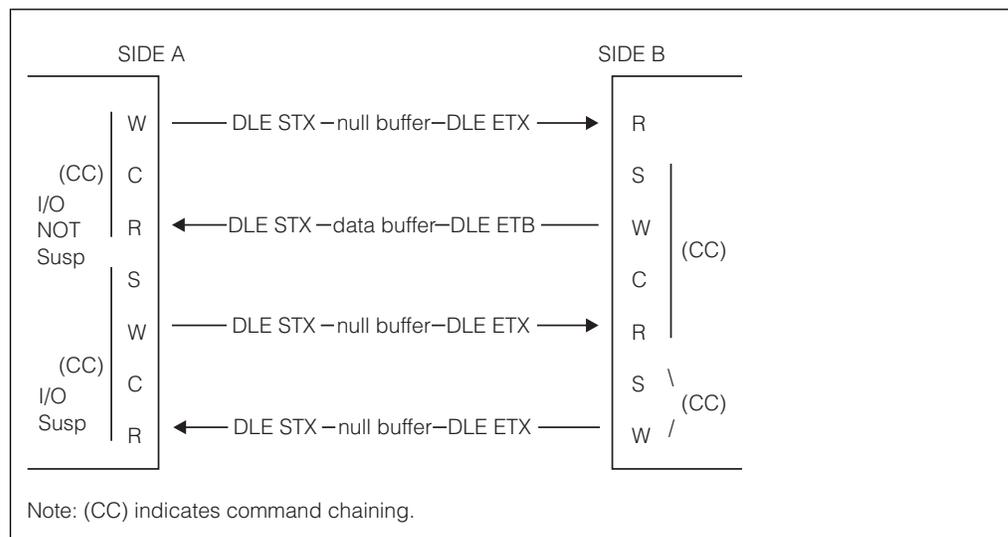
7. To prevent the possible loss of data in unexpected conditions, all reads should be for a full buffer. This is true even during I/O resumption when the first

transmission is expected to be a null buffer only. Normal BCB sequence checking and error handling apply during I/O resumption.

Wait-a-Bit and PREPARE

1. PREPARE mode can never be used during any wait-a-bit condition (by definition) because no files are active when a PREPARE is initiated.

2. PREPARE mode may not be requested with wait-a-bit all set because that would imply that the side wanting to suspend I/O could not receive data. This would make resumption difficult.

Systems Network Architecture - LU Type 0

Overview

NJE defines its own type of LU, LUTYPE 0, for all its application-to-application communication in an SNA environment. LUTYPE 0 is a full duplex protocol that is not architected in SNA. Bracket protocol is not necessary since no contention problems exist. Both ends of the SNA session are able to send and receive one or more streams. Compression is mandatory for LUTYPE 0, but compaction is optional. If compaction is desired, each node may specify the compaction table it will use for transmission to the corresponding receiving node. Each table applies for the duration of the session.

NJE (using SNA) allows for store-and-forward transmission directly through the network control program. Product implementation of SNA NJE is as follows:

- POWER, RSCS, and JES2 use SNA for NJE.
- JES3 uses MVS/BDT for SNA.

Session Initialization

In initiating an SNA NJE session, an application must issue an OPNDST OPTCD=(ACQ,SPEC) or SIMLOGON to obtain a connection to another application. The application initiating the session automatically becomes the primary logical unit (LU) for the life of that session. (This becomes important later in session termination.)

The session parameters are indicated in the BIND area; this BIND is associated with the OPNDST request. VTAM presents the BIND to the other application in its SCIP exit. This application then becomes the secondary LU. The secondary validates the BIND parameters and may choose to accept the session. To do so, the secondary LU issues an OPNSEC, which results in a positive response to the BIND.

After a positive response is returned to the primary LU, VTAM presents the SDT request in the secondary's SCIP exit. VTAM also returns the positive response to the SDT. At this point, communication between the two applications may begin.

NODE 1   VTAM   NODE 2

OPNDST   ⎯⎯ BIND ⎯⎯→   OPNSEC
   ←⎯⎯ (+)RSP ⎯⎯
   ⎯⎯ SDT ⎯⎯→
   ←⎯⎯ (+)RSP ⎯⎯

This side becomes the SNA primary LU   This side becomes the SNA secondary LU
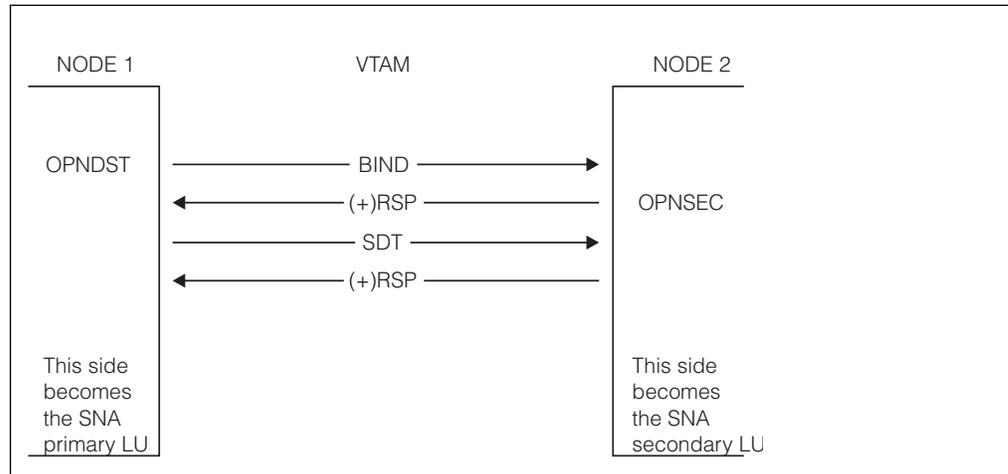
Figure 49. SNA Session Initialization

The secondary LU may reject the BIND request. In this case, a SESSIONC is issued to return a negative response to the primary LU.

NODE 1   VTAM   NODE 2

OPNDST   ⎯⎯ BIND ⎯⎯→
   ←⎯⎯ (-)RSP ⎯⎯   SESSIONC
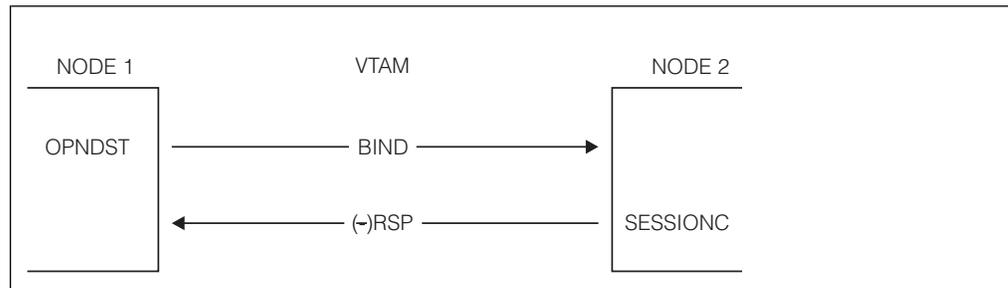
Figure 50. SNA Session Initialization Error Recovery

See SESSIONC Usage in Appendix A for JES2-specific parameters.

BIND RU Format

The BIND area is sent from the primary LU to the secondary LU to activate a SNA session. The BIND contains the parameters that will be in effect for the duration of that session. The BIND image is prefixed in the RU by the RU code for BIND, X'31'. The BIND RU mapping for LUTYPE 0 is as follows:

Table 65. BIND RU Mapping for LUTYPE 0

Byte	Bits	Value	Definition
0	0-7	X'31'	BIND code
1	0-3	B'0000'	LUTYPE 0
	4-7	B'0001'	Cold (non-negotiable)
2	0-7	B'00000011'	FM Profile 3
3	0-7	B'00000011'	TS Profile 3
4			FM usage - primary
	0	B'0'	Single RU chain
	1	B'1'	Delayed request mode
	2-3	B'11'	Definite or exception response
	4-5	B'00'	Reserved
	6	B'1'	Compression allowed
	7	B'0'	No brackets

Table 65. BIND RU Mapping for LUTYPE 0 (continued)

Byte	Bits	Value	Definition
5			FM usage - secondary
	0	B'0'	Single RU chain
	1	B'1'	Delayed request mode
	2-3	B'11'	Definite or exception response
	4-5	B'00'	Reserved
	6	B'1'	Compression allowed
	7	B'0'	No brackets
6			FM usage - common protocols
	0	B'0'	Reserved
	1	B'1'	FM headers allowed
	2	B'0'	Brackets not used
	3	B'0'	Brackets not used
	4	B'0'	Alternate code not sent
	5-7	B'000'	Reserved
7	0-1	B'00'	Full duplex
	2	B'1'	Symmetric responsibility for recovery
	3	B'0'	Reserved (no brackets)
	4-6	B'000'	Reserved
	7	B'0'	Reserved (no brackets)
8	0	B'0'	Pacing for secondary to primary occurs in one stage
		B'1'	Pacing for secondary to primary occurs in two stages
		B'bbbbbb'	
	1		Reserved
	2-7		Secondary Send pacing count
9	0-1	B'00'	Reserved
	2-7	B'bbbbbb'	Secondary Receive pacing count
10	0	B'0'	No maximum RU size for secondary
	1-7	B'0000000'	Secondary maximum RU size
11	0	B'0'	No maximum RU size for primary
	1-7	B'0000000'	Primary maximum RU size
12	0	B'0'	Pacing for primary to secondary occurs in one stages
		B'1'	Pacing for primary to secondary occurs in two stages
	1	B'0'	Reserved
	2-7	B'bbbbbb'	Primary Send pacing count
13	0-1	B'00'	Reserved
	2-7	B'bbbbbb'	Primary Receive pacing count
14			PS profile
	0	B'0'	Basic format
	1-7	B'0000000'	LUTYPE 0
15-25			No protocols specified
26	0-1	B'00'	Reserved-encryption not used by NJE
	2-7	B'000000'	No encryption by VTAM
	2-7	B'bbbbbb'	VTAM encryption
27	0-7	AL1'8'	Length of primary LU name
28-35		CL8' '	Primary LU name
36	0-7	B'00000000'	No user data

- See SNA Support in Appendix A for additional JES2 BIND information.
- See Data Flow Control in Appendix A for POWER data flow control.
- See Bind Parameters in Appendix A for additional RSCS BIND information.

Function Management Headers

Function management headers (FMHs) are used to control the data presentation for the session. FMHs are sent as only-in-chain elements, with a definite response required. They are not compressed or compacted, hence no SCBs are used.

Although several types of FM headers have been defined, only FMH4 and FMH3 are used in LUTYPE 0. These FMHs are exchanged immediately after the session has been initialized and before the initial signon and response signon path manager records are exchanged. FMH4s are exchanged first, indicating whether or not compacted data can be received. Each side then sends either signon, or FMH3 followed by signon, depending on whether the receiver supports compacted data or not *and* whether or not the sender elects to use compaction. In other words, both sides must send FMH4 indicating whether compacted data can be received or not, and may send FMH3 indicating whether compacted data is being sent or not, **completely independently** of each other, except that:

1. If side A indicates compaction is not supported as a receiver, then side B must not send an FMH3.

2. Similarly (and independently), if side B indicates compaction is not supported as a receiver, then side A must not send FMH3.

Only one FMH may be sent in an RU, hence NJE does not use the SNA FMH concatenation bit. This bit must not be set, and need not be checked. See Functional Management Headers in Appendix A for POWER-specific information; see Function Management Headers in Appendix A for RSCS-specific information.
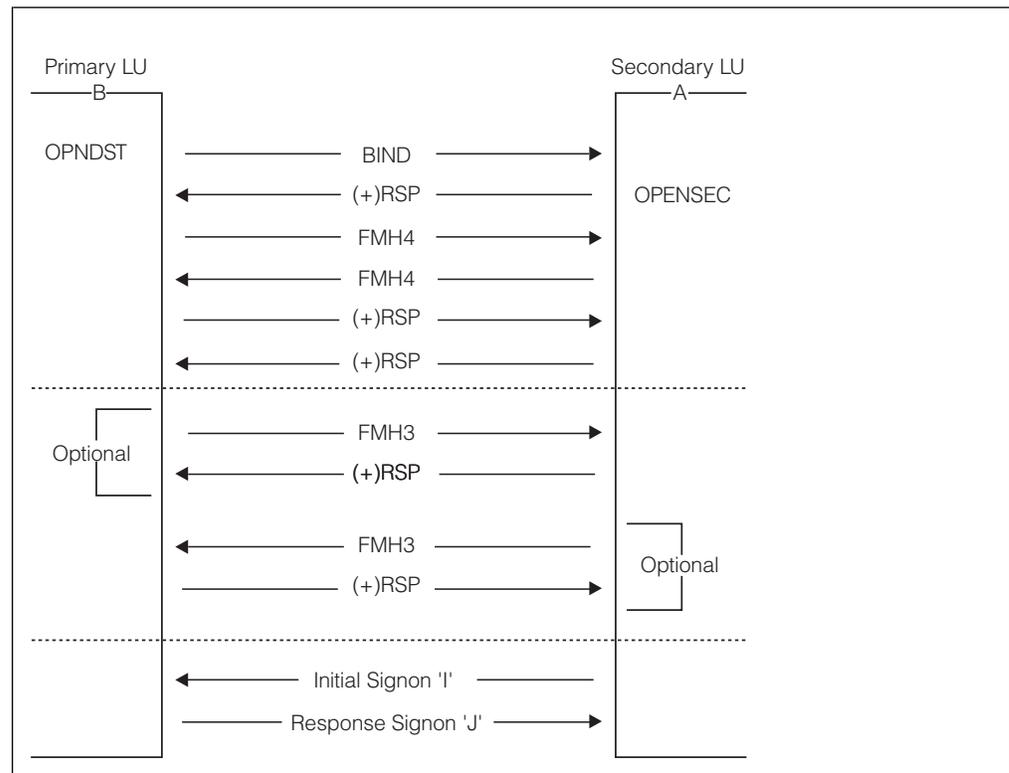
Figure 51. SNA FMH and Signon Flows

The figure above, SNA FMH and Signon Flows, shows the flows involved in the exchange of FMH4s and FMH3s prior to the signon sequence.

Note that the primary LU for the BIND flow is not necessarily the NJE primary node (the node having the higher node name and the sender of the signon I record). In fact, the figure shows an example where the roles are reversed.

FMH4

FMH4 is a non-architected SNA header used to exchange network characteristics between the two nodes of a session. Specifically, this header indicates the session RU size. If the RU size is different for the two nodes of this session, the smaller of the two sizes must be used for all data transmission.

Note: The direction of NJE is to eliminate this function of the FMH4. See Networking Connection Control Records (NCCR) for the preferred method of determining the transmission buffer size using a field in the signon record.

FMH4 indicates whether a compaction table and signon records can be received by the node that has sent the header. All NJE products must set the signon accepted bit (B'1'). Compaction may be optionally supported by setting or resetting the compaction supported bit accordingly. NJE products that receive an FMH4 with the compaction supported bit reset (B'0') must not send compacted data and must not send an FMH3. If an FMH3 is received by a sender after having indicated "compaction not supported", then the appropriate error action must be taken as follows.

The optimized fanout flag must be set (B'1'). This means that all NJE implementations must be able to receive data sets preceded by multiple data set headers and perform the necessary fanout. It is not mandatory for all products to create optimized fanout.

The FMH4 format is shown in Table 66.

Table 66. FMH4 Format

Byte	Bits	Value	Definition
0	0-7	X'08'	Length of header
1	0-1	B'00'	Reserved
	2-7	B'000100'	Header type 4
2-3		H'300' to H'65,535'	Buffer (RU) size
4-5		H'0'	Reserved
6			Features
	0	B'1'	Optimized fanout accepted
	1	B'1'	Signon accepted
	2	B'0'	Compaction not supported
		B'1'	Compaction supported
	3-7	B'00000'	Reserved
7	0	B'1'	RID format 1
	1-7	B'0000000'	Reserved

See Function Management Headers in Appendix A for JES2-specific information.

FMH3

FMH3 is specified by the sender and defines the compaction table that is to be used by its receiving partner on the session. If compaction is supported (as indicated in FMH4) by the receiver, the sender may send FMH3 prior to

signon. If the FMH3 is not sent, the sender has elected not to send compacted data (note that the sender may still optionally support receiving compacted data when the data traffic is reversed).

If compaction is not supported by the receiver (as indicated in FMH4), the sender must not send an FMH3. This means that a compaction table may only be specified on a session basis, not on individual SYSIN or SYSOUT streams. However, one compaction table may be used to receive data on one node while a different table may be used to receive data on another node.

The compaction table itself contains the master characters, followed by the non-master characters listed in reverse row-major order. (See "Compaction Table Format" for a description of compaction characters.) The format for FMH3 is shown in Table 67.

Table 67. FMH3 Format

Byte	Bits	Value	Definition
0	0-7	X'24'-X'FF'	Length of header
1	0-1	B'00'	Reserved
	2-7	B'000011'	Header type 3
2	0-7	B'00000010'	Compaction table follows
3	0-7	X'03'-X'10'	Number of master characters
4-x			Compaction table

See Functional Management Headers in Appendix A for JES2-specific information.

Error Handling Protocols

NJE does not require that all aspects of the FMHs be checked. Specifically, the FMH4 must be checked for the proper length and a valid RU size (greater than 299 bytes). All other checking is optional. A bit mask may be used to check the integrity of the required and reserved bit values. As the required bits are only for JES2 compatibility, NJE does not require the receiver to check them; however, the sender must set or reset them as specified.

FMH3s must be checked for the integrity of the compaction table and valid length. See the section "Compaction Table Format" for a description of the NJE compaction table structure. As with FMH4, checking for other bit settings is optional. Specific error situations and responses are as follows:

1. Broken FMH4 or FMH3 (length less than 300, bits set wrong, or compaction table format wrong, etc.)

   Action: Send negative RSP and UNBIND.

2. Receiving an FMH4 or FMH3 when not expected or not allowed by NJE.

   Action: Send negative RSP and UNBIND.

3. Multiple FMHs in buffer.

   Action: Product choice. Either ignore the extra data or send a negative RSP and UNBIND.

Compaction Table Format

Compaction is a technique that allows specification of two characters in one byte. Interpretation of compacted data is controlled by a compaction table that is transmitted between two networking nodes.

Note: Compaction is only done on SNA transmissions. JES2 is currently the only system that compacts data. (POWER can accept and decompact data, but does not send it.)

To interpret data that has been compacted, build a 16-by-16 matrix such as the sample matrix shown in Sample Compaction Table. Master characters are placed in the matrix beginning with position F0, F2, F3, etc. In the sample, there are 13 master characters: blank,'ADEGILNORSTU'.

When all of the master characters have been placed in the table, the non-master characters are filled in from left to right and from bottom to top. In the sample, the sequence would be:

```
X'15','.<(+|&',X'1E','$*);',X'0C','-/,',X'6C','_>'
'?\:#@'="BCFHJKMPQ',X'04','VWXYZ0123456789'
'abcdefghijklmnoprstuvwy'
```

Space in the upper left (m x m) of the matrix (where m is the number of master characters) is left blank. In the sample, m=13.

The limitations on the number of master characters are derived from recognizing that the maximum length of an FMH3 is 255 bytes and that there are 4 bytes of fixed overhead. If the number of master characters sent is m, then the number of non-master characters sent (for a 16x16 table) is 256 minus (m x (m + 1)). The smallest value of m for that works is m=3, which requires that the FMH3 total length = 251 bytes (4 overhead plus 3 master plus 244 non-master characters).

The largest value of m that works with this algorithm is 15. In this case, only 16 non-master characters are sent, yielding an FMH3 length of 36 bytes (4 overhead plus 15 master plus 16 non-master characters). Actually, up to 16 master characters may be sent, thus zero non-master characters. In this case, only sequences containing the master characters can be compacted and decompacted, and the 16 by 16 matrix need not be built. Each byte is interpreted as two 4-bit indices into the list of 16 master characters, thus yielding two bytes for each byte of compacted data.

The FM header type 3 that would be used to send the table shown in Sample Compaction Table is Table 68 shown below.

Table 68. FM Header Type 3

BYTE	VALUE	MEANING
0	X'5B'	Length of header: 4+256-(m x m), where m=13
1	X'03'	Header type 3
2	X'02'	Compaction table
3	X'0D'	Number of master characters
4-X'5A'		Master characters followed by non-master characters (max 252)
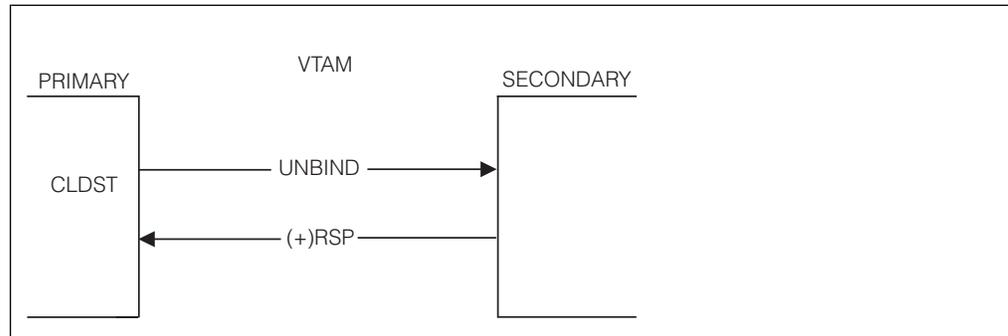
Figure 52. Sample Compaction Table. This figure shows how a sample master and non-master character set are placed in the compaction table.

The corresponding byte sequence for the FMH3 followed by the compaction table is:

```
Meaning     Len type CT  #M  |---- Master Characters ---
Graphic                           A   D   E   G   I   L
Data        5B  03   02  0D  40  C1  C4  C5  C7  C9  D3
Byte Number 0   1    2   3   4   5   6   7   8   9   10

Meaning     --Master (cont'd)----|  |-- Non-Master Characters -->
Graphic     N   O   R   S   T   U      .   <   (
Data        D5  D6  D9  E2  E3  E4  15  4B  4C  4D  ... etc
Byte Number 11  12  13  14  15  16  17  18  19  20
```

Sample Compacted Data: Following is a short data stream in its uncompacted form. The Sample Compacted Data figure shows how the data would appear in its compacted form and how the sample decompaction table (above) would be used:

```
    Sample Data Stream (between the quotes)

 'REQ/MODULE/MACRO/SOURCE NAME  -------------'  ...
```

```
Graphic                RE   Q   /  M OD UL   E   /  M
Data            55  93 CD EB DE 82 C6 F3 EB DE
Byte Number      0   1   2   3   4   5   6   7   8   9

Graphic          A   C RO   / SO UR  C  E_ NA  M
Data            F1  D9  98 EB A8 C9 D9 30 71 DE
Byte Number     11  12  13  14  15  16  17  18  19  20

Graphic          E_   _       -
Data            30  F0 CD 60 0B 0D  15  ...
Byte Number     11  12  13  14  15  16  17
```

Byte	Value	Meaning
1	55	Bits 0 and 1 indicate compacted data
2		Bits 2-7 ('010101'B) indicate 21 bytes
	93	This value is within the (mxm) portion of the matrix. To decode it, use row F. Hence, F9 is an 'R' and F3 is an 'E', so that 93 when decompacted becomes 'RE'.
3	CD	This value is not the (mxm) part of the matrix, hence the character 'Q' is substituted directly from the table.
4	EB	/
5	DE	M
6	82	OD
7-20	C6-DE	ULE/MACRO/SOURCE NAM
21	30	E blank
22	F0	blank
23	CD	SCB indicating repeat next 13 characters
24	60	Dash (-), the character to be repeated
25	0B	SCB indicating next 11 characters non-compressed and non-compacted
26	0D	Carriage return
27	15	Line feed

Figure 53. Sample Compacted Data. This figure shows a sample compacted data stream with a byte-by-byte illustration of the results of applying the table to decompress the data.

Session Termination

The following describes two types of termination:
- Normal termination with quiesce.
- Normal termination - immediate.

Normal termination can be effected by either the application or the VTAM operator. UNBIND flows in these cases, and the LOSTERM exit of the primary LU is driven.

On the other hand, when a link breaks, UNBIND does not flow from primary LU (PLU) to secondary LU (SLU). Rather, it is simulated to both the PLU and the SLU and the NS exits at both ends are driven.

Normal Termination with Quiesce

Normal session termination may be initiated by either the primary or secondary LU. After all session activity has quiesced, the primary LU starts session termination by issuing a CLSDST. From the CLSDST, VTAM presents an UNBIND request to the secondary LU SCIP exit. VTAM returns the positive responses to the

UNBIND to the primary LU. The session has then been terminated. No new sessions may be started until an outstanding CLSDST has completed.
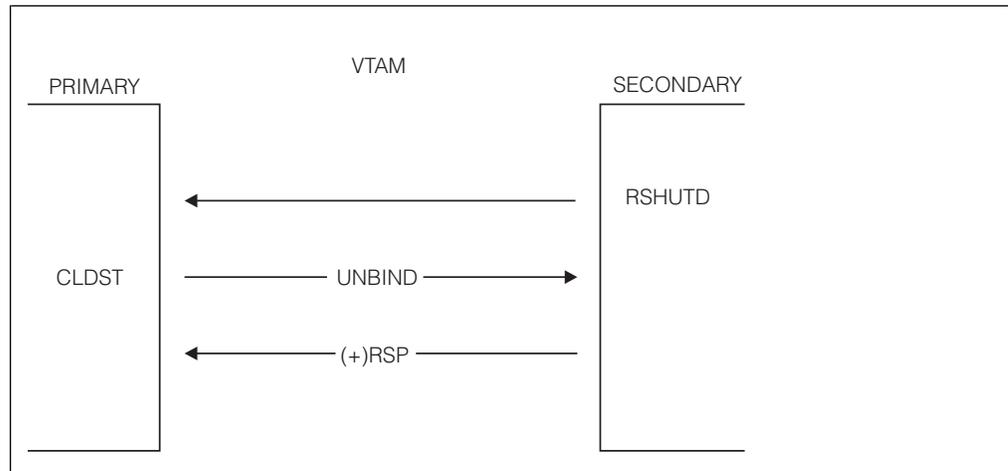
Figure 54. SNA Session Termination - Initiated by Primary Node

If the secondary LU wishes to terminate the session, it must issue RSHUTD. After all data activity has stopped, the primary LU will respond with a CLSDST request. This will result in an UNBIND, which will terminate the session. Note that CLSDST may only be issued by the primary LU.
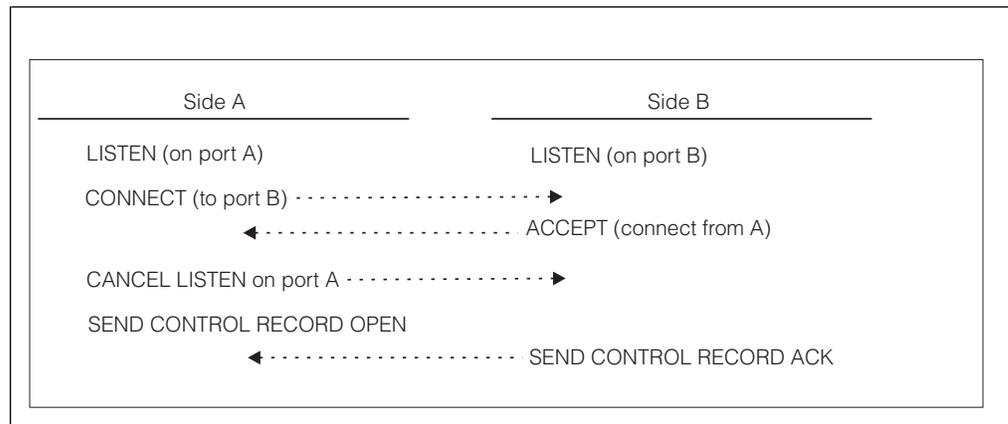
Figure 55. SNA Session Termination - Initiated by Secondary Node

Normal Termination - Immediate

Immediate termination may be initiated by either the primary or secondary LU. It may occur as a result of VTAM errors, a VTAM VARY NET,INACT issued for the application, or an NJE node termination.

If initiated by the primary, CLSDST is requested without waiting for current session activity to quiesce. Otherwise, processing is similar to when the primary LU initiates normal session termination.

```
      0 1 2 3 4 5 6 7 8 9 A B C D E F
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    TYPE    |      RHOST        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | RIP   |    OHOST    |   OIP    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | R |
     +-+
```
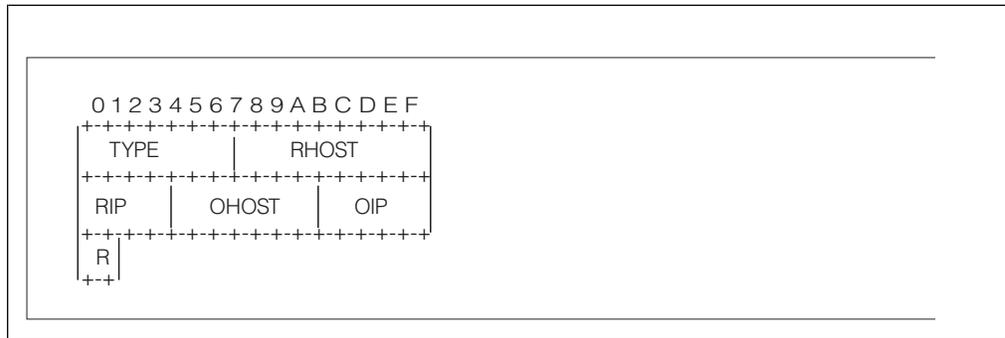
Figure 56. SNA Immediate Session Termination - Initiated by Primary Node

The secondary LU initiates abnormal termination with a TERMSESS request.
Through TERMSESS, VTAM presents the primary LU's LOSTERM exit with a
LOGOFF request. The primary LU then responds by aborting current session
activity and issuing a CLSDST request. As before, VTAM presents the UNBIND
requests to the secondary LU, thereby terminating the session.

```
    ---------------------------------TTB  TTR  data  TTR  data  TTREOB ---------------------------------
```
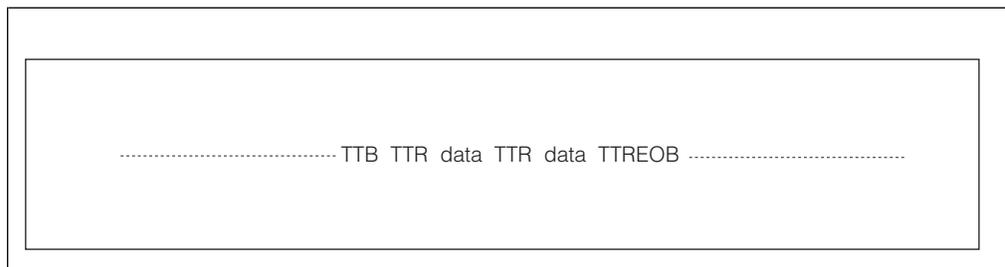
Figure 57. SNA Immediate Session Termination - Initiated by Secondary Node

See Termination in Appendix A for JES2 specific information.

Error Recovery

NJE (using SNA) data is sent as only-in-chain elements, with an exception
response required. If an exception response is received, all streams will be
terminated and the session will be closed using TERMSESS and CLSDST.

If an LUTYPE 0 application receives a data flow command that is not permitted
with the LUTYPE 0 architecture, the application will terminate the session.

FMHs are sent only-in-chain, with a definite response required. If an FMH is not
acceptable, the receiving application returns a negative response to the sender.

ACF/VTAM Considerations

Application exits and data flow commands are defined to VTAM and can be used
as described below.

Application Exits

Application exit routines are defined to VTAM during ACB OPEN. The following
exit routines could be used by the application for NJE using SNA.

SCIP
> The SCIP exit is scheduled by VTAM to handle BIND, UNBIND, and SDT
> requests.

LOSTERM

The LOSTERM exit is scheduled for normal session termination and for a session failure. The application should either immediately terminate or quiesce the session, based on the completion code in the RPL.

TPEND

This exit is scheduled when a HALT command has been entered or VTAM is abnormally terminating. If a normal HALT is requested, the session should be terminated after all activity has quiesced. For HALT NET,QUICK and VTAM termination, the session should be aborted by the application (using TERMSESS or CLSDST).

NSEXIT

VTAM calls this exit with a CLEANUP RU if the session has been lost.

Data Flow Commands

The following data flow commands are defined for VTAM. Only those marked "YES" may be sent or received by an LUTYPE 0 application.

Normal Flow Commands	Send	Receive
BID	NO	N/A
CANCEL	NO	NO
CHASE	NO	NO
LUS	NO	NO
QC	NO	NO
RTR	N/A	NO

Note:

Expedited Flow Commands	Send	Receive
QEC [6]	NO	N/A
RELQ [7]	NO	NO
RSHUTD	YES	YES
SHUTC	N/A	NO
SHUTD	NO	N/A
SIGNAL	N/A	NO

SESSIONC Commands	Send	Receive
CLEAR	NO	YES [8]
RQR [7]	N/A	NO
SDT	YES[8] [9]	YES
STSN [7]	NO	N/A

---

6. Not allowed in FM profile 3

7. Not allowed in TS profile 3

8. Ignored by the application if received

9. Sent automatically by VTAM after OPNDST

Chapter 12. TCP/IP Transmission

NJE uses TCP/IP protocol for its traffic over a TCP/IP network. The following figure describes the new protocol that allows CTC NJE data buffers transferred through TCP/IP.
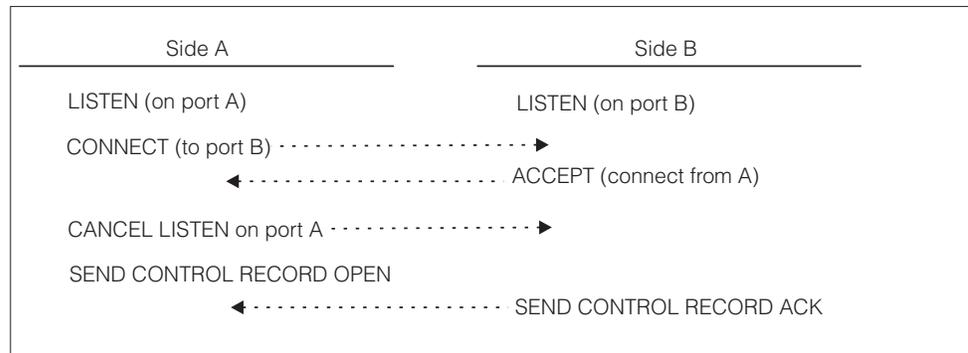
Figure 58. Description of new protocol allowing transfer of CTC NJE data

RSCS 3.2, VSE/POWER 6.5 (with VSE/ESA 2.5), JES2, JES3, and all their later releases support this protocol currently.

Initialization

To begin communication, a passive listen must be performed for a specific port (the default is 175). The listen allows another node to attempt a connection. Either side can attempt to connect to the other side's port. A connection attempts to connect to an issuing TCP connect socket call. Side B accepts the connection attempt with an ACCEPT socket call.

If a connection comes in that does not match an outstanding listen, you must ensure that the attempted connection is closed with a CLOSE socket call.

- If a control record is not valid, the socket must be closed. If it is valid, but not for the expected session, a NAK X'01' control record must be sent as a response.
- If the control record matches but the link is in a connected state, it is assumed that the connection is gone; however, there was no notification of this severance. In this case, the connect must be responded to with a NAK X'02' control record and termination of the driver with the option to restart.
- If the control record is an attempt to make an active connection, the socket must be closed and a response of NAK X'03' must be sent.
- If a temporary condition exists that prevents establishment of the NJE session, but allows subsequent restart attempts, the socket must be closed and a response of NAK X'04' must be sent.

The following are the actions for NAK responses to active connection attempts.

- NAK X'01' - Terminate the session without restart.
- NAK X'02', X'03', and X'04' - Close the socket and add a random number of seconds before the connection is attempted again.

The exchange of control records must take place as the first exchange of data in the TCP connection after the TCP connection is complete. The side that had the connect accepted, in the figure Side A, sends a control record OPEN. An OPEN is responded to with a control record ACK.

When the open completes and the control records are successfully exchanged and properly acknowledged, the NJE nodes exchange their signon sequences. Until the signon sequences (the exchange of type I and J signon records) are complete, the links remain in an ACTIVE mode, only going to CONNECT state when the NJE signon has successfully completed.

A control record has the following format:

```
      0 1 2 3 4 5 6 7 8 9 A B C D E F
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |    TYPE      |      RHOST      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |  RIP   |    OHOST    |   OIP   |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     | R |
     +-+-+
```
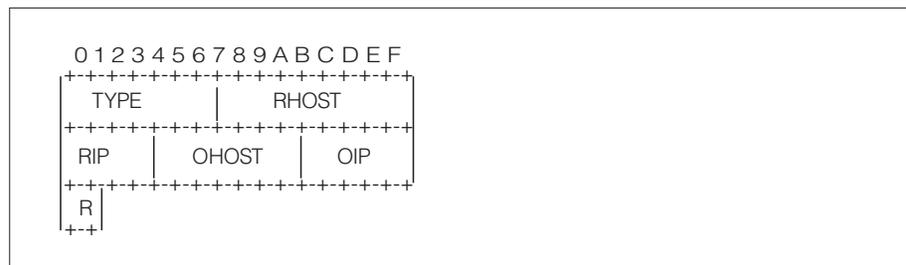
Figure 59. Format for the control record

TYPE Type of request in EBCDIC characters, left-aligned, and padded with blanks. Acceptable values are OPEN, ACK, and NAK.

RHOST
        Name of the host sending the control record and is the same value as the RSCS LOCAL associated with this link. This field is EBCDIC characters, left-aligned, and padded with blanks.

RIP   Hexadecimal value of the IP address sending the control record. For example, the IP address 128.112.14.1 has a value of X'80700E01'.

OHOST
        The name of the host expected to receive the control record. Same format as RHOST.

OIP   Hexadecimal value of the IP address expected to receive the control record. Same format as RIP.

R   If TYPE=NAK, reason code in binary, used to return additional information. Valid values are:
        • X'01' No such link can be found
        • X'02' Link found in active state and is reset
        • X'03' Link found attempting an active open.
        • X'04' A temporary condition exists that prevents the establishment of the NJE session.

The NJE node receiving the control record uses OHOSTNM and ONODENM to verify that the connection is made to the correct site and uses RHOSTNM to search for a definition of a link by that name. Several checks are made by the NJE node receiving the control record, such as:
• Is the link attempting an active open
• Is a link by that name defined

- Is the link currently connected

Currently, JES TCP/IP support does not issue SYN NAK, but a JES TCP/IP application can handle SYN NAKs from the server node.

The following figure summarizes how the process works:
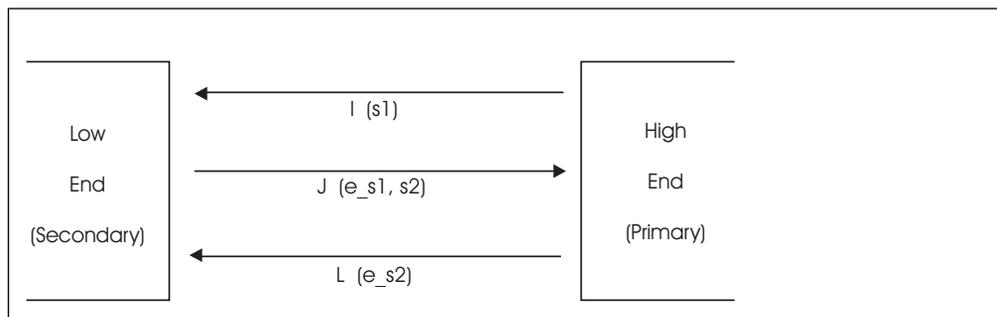
Figure 60. JES TCP/IP Application with SYN NAKs from the Server Node

Except for control records, each transmission block that is sent over TCP NJE links is preceded by a block header. A block header (TTB) is 8 bytes long and has the following format.
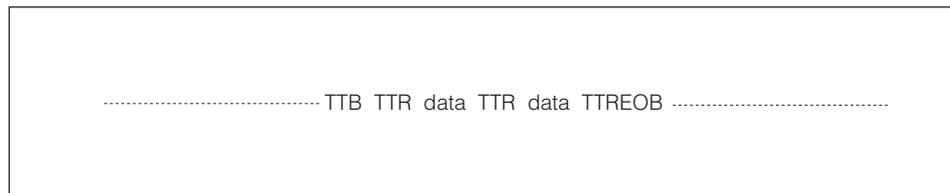
Figure 61. Format of TCP NJE Block

An IPV6 NJE control record has the following format:

TYPE   The 8 byte character type of request that is left-aligned and padded with blanks.

      Acceptable values are:
- OPENNJE
- ACKNJE
- NAKNJE

VERSION

      A 1 byte character version number of the control record.

      Acceptable values are:

Table 69. Acceptable Values for Version Number of the Control Record

Value	Version
0x00	IPV6 signon with host names
0x01	IPV6 signon for a TTLS connection
0xnn	Future versions

RNODENUM

The NJE node name of the host sending control record that is associated with the link. It is an 8 byte character that is left-aligned and padded with blanks.

RHOSTNM

The name of the host sending the control record that is associated with this link. It is a 255 byte character that is left-aligned and padded with blanks.

ONODENM

The NJE node name of the host receiving the control record that is associated with the link. It is an 8 byte character that is left-aligned and padded with blanks.

OHOSTNM

The Name of the host that is receiving the control record that is associated with this link. It is a 255-byte character that is left-aligned and padded with blanks.

R   A reason code in binary that is used to return additional information if the type is NAKNJE. The reason codes are similar to the existing reason codes for the IPV4 TCP/IP Signon.

```
0   8   17   271
+++++++++++++++++~+++
|TYPE | V | RNODENM|RHOSTNM|
+++++++++++++++~++~+++
|ONODENM|OHOSTNM|R|
 +++++++++++++~++++++
272   280   535
```

Figure 62. Format of an IPV6 NJE Control Record

If the state of the link on the node receiving the control record permits a connection, a control record of type ACKNJE is sent back to the sending node with the following values set:
- RHODENM and RHOSTNM values based on the sending node
- ONODENM and OHOSTNM values based on the receiving node

If the RHOSTNM value in the OPENNJE record is ::1 that is a loopback address (the remote and local host are the same), and the OHOSTNM value is set to the current host name.

Once the open has completed and the control records have been successfully exchanged and properly acknowledged, the NJE nodes then exchange their signon sequences (the exchange of type I and J signon records). Until the signon sequences are complete the links remain in an ACTIVE mode and only go into the CONNECT state when the NJE signon has successfully completed.

If the state of the link receiving the OPENNJE control record does not permit the connection a control record, with type set to NAKNJE, is sent to the sending node and if a NAKNJE control record is sent the TCP virtual circuit is closed allowing for another passive open to be issued.

An NAKNJE control record has a reason code field that indicates the reason for rejecting the open request. Reason codes that are currently used are the same reason codes as for the IPV4 TCP/IP Signon.

Outbound TCP/IP Signon (from client side): After receiving an ACK or ACKNJE record, the node initiating the connection (client node) can send either SOHENQ or SYN NAK records depending on whether it wants to become a primary node or not. An SOHENQ record from the node indicates that the node wants to be a primary node. An SYN NAK record from the node indicates that the node can either be a primary or secondary node.

The NJE primary node always sends the initial signon record during an NJE Signon. For an SOHENQ, the remote node (server node) can send a DLE ACK0, in which case the outbound node would become the primary node. However, if the server node wants to be a primary node, it can send a SOHENQ to the client node. If DLE ACK0 is not received for a SOHENQ the initiating node closes the connection. Currently JES TCP/IP support does not issue SYN NAK, but a JES TCP/IP application can handle SYN NAK from the server node.

If NodeA (outbound JES node) initiates the connection:

```
Node A Node B
Connect --------------->
OPEN(NJE)--------------->
<-------------- ACK(NJE)
```

Normal Sequence:

```
SOHENQ --------------->
<-------------- DLEACK0
```

Error Sequence:

```
SOHENQ --------------->
<-------------- Anything other than DLE ACK0
Connection broken by A
```

Inbound TCP/IP Sign-on (from server side): After the TCP/IP Signon the server node can receive a SOHENQ or SYN NAK from the client node (the node initiating the connection). When SOHENQ is received, a DLE ACK0 is issued that allows the client node become a primary node. The client then goes on to issue an I record during NJE Signon. When SYN NAK is received the JES TCP/IP support, on the server node, issues a SOHENQ to indicate to the client node that the server node can become a primary node. The client node responds with a DLE ACK0 and the server node is now set to send an I record. However, if the client node sends another SOH ENQ instead of a DLE ACK0, the connection is terminated by the server node.

If NodeB initiates the connection and JES node is NodeA:

```
NodeA Node B
<-------------- Connect
<-------------- OPEN(NJE)
ACK(NJE)-------------->
```

Normal Sequences:

Sequence 1:

```
<-------------- SOHENQ
DLE ACK0-------------->
<-------------- I record
```

Sequence 2:

```
<-------------- SYN NAK
SOH ENQ -------------->
<------------- DLE ACK0
I record-------------->
```

Error Sequences:

Sequence 1:

Sequence 1:

```
<-------------- Neither SOHENQ nor SYN NAK
Connection broken by A
```

Sequence 2:

```
<-------------- SYN NAK
SOH ENQ -------------->
<------------- Anything other than DLE ACK0
Connection broken by A
```

F Byte 0 (FLAG values currently not used).

U Byte 1 is unused.

LN Bytes 2 and 3 represent the length of the data block. Binary 16-bit value is the total length of the data block including the length of the TTB and end of buffer TTR.

UNUSED
Bytes 4-7 are reserved for future expansion.

A data block can contain one or more data records, and each data record is preceded by a data block record header (TTR). A TTR has the following format.


```
 0  1  2  3  4  5  6  7
+--+--+-----+--+--+--+--+
|F |U | LN  |U |NU|SE|D |
+--+--+-----+--+--+--+--+
```
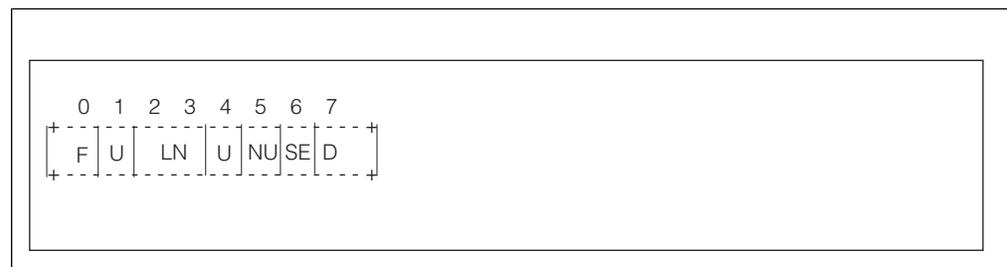
Figure 63. Format of Block Header (TTR)

The maximum buffer size for a TCP send/receive is 8192. However, all TCP READS are variable length, and as a result, processing might require multiple reads might have to be performed to determine the size of the DATA BLOCK (TTB) and its first record (TTR). Individual records might need to be read in the same fashion of multiple receives.

Secure sockets layer (SSL) feature

If a TCP/IP connection uses the SSL feature, control records are exchanged as for a non-SSL connection except the type of the first control record containing 'OPEN SSL' (with one blank between OPEN and SSL). According to the SSL feature, the encryption takes place only for NJE data records, which means the encryption is started after a control record of type ACK has been exchanged.

VSE/POWER, JES2, and JES3 support SSL-connections currently.

Generalized Control Records

The following figure shows the format of a generalized NJE control record:

```
            0  1  2  3
        +-----------------+
        |  F | U|   LN  |
        +-----------------+
```
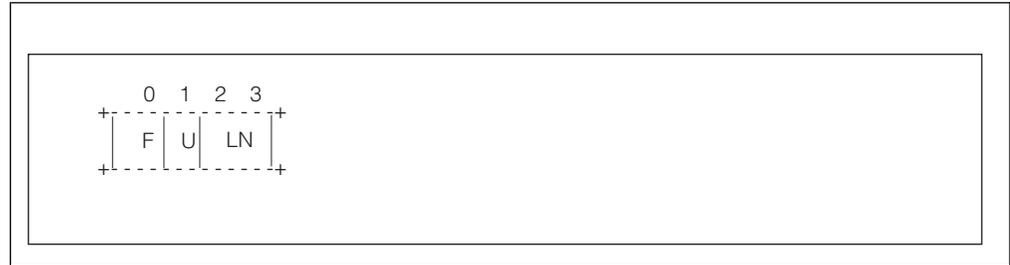
Figure 64. Format of a data block record header (TTR)

JES2 supports generalized control records.

Appendix A. System-Dependent Considerations

This section contains system-dependent considerations and deviations from the protocols for each of the networking facilities.

Support of Various NJE Features

The following table is provided as a summary chart showing which features are supported by the various products.

Table 70. Various NJE Features Supported by Networking Facilities

Feature	JES2	JES3	RSCS	VSE/POWER	BDT
TCP/IP communications	Yes	YES	Yes	Yes	No
BSC Communications	Yes	Yes	Yes	No	(J3)
SNA Communications	Yes	No(B)	Yes(*)	Yes	Yes
CTC Communications	Yes	Yes	Yes	No	No(J3)
Network Path Manager	Yes	No	No	No	No
Formatted Commands	Yes	(A)	(A)	No	N/A (J3)
Data Compaction	Yes	No	No	(A)	No
Output Fanout	Yes	(A)	Yes(*)	(A)	N/A (J3)
Spanned Headers	Yes	Yes	Yes	Yes	Yes
Multiple Streams	7	2(**)	Yes(*)	Yes	28
Wait-A-Bit All Streams	Yes	(A)	Yes	Yes	N/A
Null Buffers vs. DLE ACK0	(A)	(A)	Yes(*)	Yes	N/A
PREPARE Mode - Suspend I/O	No	No	Yes(*)	No	N/A
V.27 Contention Resolution	No	No	Yes(*)	No	N/A
Enhanced Termination - Quiesce	No	No	No	No	Yes
Signon Concurrence	(A)	(A)	Yes(*)	Yes	Yes

```
Key:
 YES  =  Supports the Feature
 NO   =  Does Not Support The Feature
 N/A  =  Not Applicable
 (A)  =  Accepts, but does not Send.
 (B)  =  Function Supported by BDT
 (J3) =  Function Supported by JES3
 (*)  =  Provided with RSCS Networking Version 2.
 (**) =  JES3 supports 1 SYSIN and 1 SYSOUT stream in parallel.
```

JES2

Network Control

Network Connection Control Records
JES2 supports I, J, K, L, M, N, and B (SRCB) records. The initial signon (I) and response signon (J) records must be the only records in their respective buffers. The other records may be sent in the same buffer. JES2 allows a user to specify his own function for SRCB types S-Z.

You can define a non-path manager connection to JES2 by entering NODE(n) PATHMGR=NO at initialization. JES2 sends only I, J, and B records to non-path manager nodes.

Network Addressing, Topology & Routing

Naming Conventions

JES2 supports up to 9999 remotes and 9999 nodes. Remotes and nodes are defined during JES2 initialization. JES2 allows the installation to specify a 1-8 alphameric name to symbolically define a node. The default node name is of the form Nnnnn. Leading zeros for node numbers are compressed. All members of a multi-access spool configuration must have the same node name in the network.

Parallel Lines

JES2 supports an unlimited number of parallel BSC lines connecting two nodes.

Commands and Messages (NMRs)

JES2 sends an SRCB of X'00' on NMRs.

Command Transmission

JES2 builds unformatted nodal message records (NMRs) as a result of $N operator commands.

Formatted (Global) Commands

JES2 builds formatted NMRs for "global" commands. The following global commands are supported by JES2:

Display job
Using the $G D operator command

Hold job
Using the $G H operator command

Release job
Using the $G A operator command

Cancel job
Using the $G C operator command

Route job (SYSIN or SYSOUT)
Using the $G R operator command

JES2 assumes that destinations for the global route command are syntax-checked at the receiving node. As a receiving node, JES2 may reject a global command if only the job name is specified and JES2 finds more than one job with that name. A message is issued to the local console if the global command is rejected.

Command Authorization

JES2 supports four levels of command authority, which may be specified on a node basis. The authority levels are as follows:

- Network - The node has the same authority as local consoles; this includes device, job and system authority.
- Device - The node has the authority to issue device-related commands.
- Job - The node has the authority to issue job-related commands.
- System - The node has the authority to issue certain system commands.

Message Transmission

Message NMRs are created by JES2 in three instances:

1. In response to a formatted or unformatted command sent across a network link.

2. Via a $DM operator command, which is used to send network messages.
3. For notification messages.

SYSIN (Job Input) Transmission

Store-and-Forward Transparency

As an intermediate node, JES2 will add the JES2 and job scheduling sections to the end of the job header if that header does not already contain those sections. JES2 also adds the accounting section to the job trailer for intermediate node processing.

/*XMIT or //XMIT

The XMIT statement allows the user to submit non-MVS jobs for network transmission. All JES2 syntax scanning stops after an XMIT statement is encountered within a job. Only the data after the XMIT statement (and before a specified delimiter) is transmitted across the network. Note that a valid MVS JOB statement must precede the XMIT statement.

The job name placed in the job header of an XMIT job is the name from the preceding MVS JOB statement. If this name is blanks, JES2 transmits a blank job name in the job header. All relevant job header information (such as priority and accounting information) is taken from the preceding JOB statement.

Job Header

The job header and trailer are created when a job is read into the JES2 system and stored in the same buffer as the job control table (JCT) for the job. Therefore, the total size of both the job header and trailer is limited by the spool buffer size (maximum and most common size is 4008) minus the JCT.

For Version 4.2, the job header was moved to its own buffer(s) on spool, chained out of the JCT. This job header's maximum size is 31637 bytes. JES2 segments control records in lengths of 253 bytes.

JES2 Section

The JES2 section of the job header contains two fields -- a flag byte and the originator's account number -- along with some fields used by the spool offload facility that are zeroed out for NJE transmissions. JES2 appends this section to the job header even at intermediate nodes.

Job Scheduling Section

The job scheduling section is sent in the job header. As an intermediate node, JES2 appends this section to the job header. Down-level JES2 nodes (those prior to SP Version 1.3.3) require no additional code to store-and-forward this section.

User Sections

If user sections are added, they must be placed before the spool offload section (even though the spool offload section is not sent on a NJE transmission).

Data Set Header

Record Characteristics Change Section

JES2 does not support the receipt of an RCCS prior to the receipt of a JOB statement. JES2 does not create or receive a segmented record characteristics change section.

Maximum SYSIN Data Record Length

The maximum length of an NJE SYSIN record is 252 bytes. Records are written

to spool in 256-byte maximum segments. The SYSIN record cannot be spanned; thus, the maximum is 252, with the extra bytes being used for record control.

Acceptable Job Streams

Multiple Jobs Between a Header and a Trailer
JES2 cannot accept more than one job between a job header and trailer.

Errors in JES2 JECL Statements
If an error is encountered while processing JECL statements, JES2 input processing skips the remaining records in the job and queues the job (with error messages) for output. Error messages are returned to the originating node and user.

Notification

When a job is first read in by JES2, the TSO/VM userid for notification is stored in the job header if NOTIFY was specified on the JOB statement or a JECL /*NOTIFY statement was included in the job. The originating node name is also stored in the job header.

If the job header contains a userid and the job is transmitted from its originating node for execution on another node, the job transmitter issues a message to the TSO/VM user indicating that the job was transmitted for execution.

When the job completes execution, a notification message is directed to the TSO/VM userid at the originating node. If the originating node and execution node are the same, this results in a TSO SEND command specifying the userid. Otherwise, the notification is sent in a nodal message record to the originating node, and that node issues the SEND command to the userid. There is also notification when each SYSOUT data set group reaches the destination node. (See "Notification" on page 224 for details.)

SYSIN Job Routing

Use of System Qualifier
There is no facility to route a job to a particular system in a JES2 MAS configuration. Either AFF= must be coded on the /*JOBPARM statement or a job class structure must be used to control which system executes the job.

Use of Userid
The use of userid on the /*XMIT or /*XEQ JECL statement is provided for routing SYSIN jobs to a VM userid. As the target execution node, JES2 ignores the NJHGXEQU field.

Operator Rerouting
JES2 operators may change the execution node (but not userid) for jobs on the JES2 queue, through the $R XEQ command.

Undefined Node
If JES2 receives a SYSIN job destined for an undefined (not unconnected) node, it will queue the job for local execution (with no error message).

Jobid Assignment

The JES2 job identifier (jobid) is a halfword binary number that is assigned when a job first enters the system. This number is unique within a JES2 system. The job header always contains the original (input system's) jobid.

When a job is transmitted from one system to another, the receiving system attempts to assign the original jobid (from the job header) to the job that is being received. If this number is currently in use on the receiving system, a jobid is

assigned as if the job were being read in locally; that is, the job counter is
incremented by one until an available number is found. A new jobid is assigned
even if a part of the original job is on a receiving system (as may occur for spin
data sets). The newly-assigned jobid is not transmitted in the job header.

SYSOUT (Job Output) Transmission

Store-and-Forward Transparency

All networking levels of JES2 transparently store-and-forward SYSOUT data sets.
As an intermediate node, JES2 will add the JES2 and job scheduling sections to the
end of the job header if that header does not already contain those sections. JES2
also adds the output processing section (also called the "data stream section" by
JES2) to the data set header but does not add the accounting section to the job
trailer.

JESNEWS

In certain situations, JES2 attempts to append the JESNEWS data set to the
front of the job log data set. This occurs when the job log data set is
transmitted from either the execution node or from an intermediate node.
JESNEWS is appended only if both of the following two conditions are met:

1. JESNEWS exists on the transmitting system.
2. The job log contains variable length records.

Trailing Blank Truncation

While JES2 truncates trailing blanks on spool, it keeps track of the original
record length and restores it upon re-transmission. JES2 also provides an
option to preserve trailing blanks on spool for specified SYSOUT classes.

Job Header

As an intermediate node, JES2 adds the JES2 and job scheduling sections to the
end of the job header if that header does not already contain those sections.

Job Copies

JES2 sets and uses the job copies field in the job header. This field is multiplied
by the copies field in the data set header to calculate the total number of
copies for a data set. The $T O operator command is used to update the
characteristics of data sets at a JES2 node. The 3800 characteristics is updated
only if the existing data set header contains a 3800 section. JES2 does not
update any data sets that have multiple "clone" copies (that is, /*JOBPARM
copies or those created by a $N operator command).

Job Trailer

Accounting Section

As an intermediate node, JES2 will not add the accounting section to the end
of the job trailer for SYSOUT.

Data Set Header

If this is the execution node, the data set header is created when the data set has
been selected for transmission. If this is an intermediate node, the original data set
header is transmitted. As an intermediate node, JES2 adds the output processing
section to the end of the data set header if the header does not already contain that
section.

The JES2 SYSOUT receiver spools data set headers as they are received. The data
set headers for a particular data set are stored contiguously in a spool buffer.
Additional buffers may be used to spool the headers. Data set headers may not

span buffers; therefore, the size of the data set header is limited by the spool buffer size (minus the size of the I/O block (IOB) that precedes the buffer).

3800 Section
> On the execution node, JES2 does not create a 3800 section if the data set has only the default 3800 characteristics specified.

Multiple Data Set Headers
> JES2 is able to send and receive multiple data set headers for a data set with more than one destination.

Notification

When any of the job's output data sets reaches the destination node, the destination SYSOUT receiver issues a message to the TSO/VM userid specified in the job header. This notification message indicates where the job's data sets were received and is sent to the job's originating node. See "Notification" on page 222 for a description of the end-of-execution message.

Job Output Routing

Default Output Routing
> Unless otherwise specified, JES2 routes job output back to the originating node and remote workstation unless specifically routed with a /*ROUTE or other JCL/JECL statement.

Undefined Node
> If JES2 receives a SYSOUT job destined for an undefined (not unconnected) node, it will queue the job for local processing (with no error message).

Operator Rerouting
> JES2 operators may change the destination node (but not userid) for output on the JES2 output queue, either by job or by output group. To reroute a node.userid that contains a dash in the userid, an operator must use parentheses as delimiters for the userid.
>
> **Note:** A JES2 userid is a 1-to 8-character symbol identifying a system user. Local node userids can contain A-Z, 0-9, and the characters #, $, and @. Non-local userids can contain all symbols except for periods, colons, dashes, and parentheses.

Interactive Data Transmission Facility

A file sent by the TSO/E TRANSMIT command has an external writer name that is identical to the remote/userid field in the data set header. Each transmitted data set is preceded by an internal header. The data records may be any record length, but TSO/E passes the records to JES2 as fixed-length 80-byte records without carriage control.

When this type of file is received at the destination node, JES2 provides an installation exit (13) to allow the installation to retain or delete the incoming file. This exit may also change the target userid. Based on a return code from the exit, JES2 issues the message "MAIL FROM (node/userid)" to the TSO userid.

If the userid is not valid for this system, the sender is notified with the message "MAIL TO (node/userid) DELETED, INVALID USERID".

Stream Support and Control

Multiple Streams

JES2 supports up to 7 job transmitters, 7 SYSOUT transmitters, 7 job receivers, and 7 SYSOUT receivers per network line. The total number of job and SYSOUT streams per line cannot exceed 8. This means that the sum of the job receivers and SYSOUT receivers on a line is less than or equal to 8. (Similar logic applies for transmitters.) The operator has the capability to start and stop each individual stream. A console stream is always defined for an NJE link.

Stream Initiation and Suspension

Receiver Cancel

JES2 responds to a "receiver cancel" with an transmitter cancel (SCB of X'40') or End of Transmission (EOT).

Operator Control (of Streams & Lines)

The operator commands that control NJE devices -- transmitters (e.g., Ln.JTn) and receivers (e.g., Ln.SRn) -- start and stop the streams on an individual basis.

SNA Support

Session Initiation

JES2 uses OPNDST OPTCD=(ACQ,SPEC) to connect to another SNA node. The node issuing the start networking command ($SN) becomes the primary LU node.

Bind Parameters

JES2 creates the BIND from three sources: MODTAB, JES2 initialization parameters, and an internal table that forces certain parameters such as TS and FM profile types. JES2 allows only the maximum primary and secondary RU sizes to be variable and requires the network topology and output dispersal flags to be on in FMH type 4.

SESSIONC Usage

JES2 sends SESSIONC for the following cases:

- Invalid application name in the JES2 application table.
- No logical SNA line available for this session.
- Invalid parameters in the BIND.
- This node is already in session or another OPNSEC is pending.

Function Management Headers

Compression

JES2 always sets the compression indicator on even if compression is not being used for that session. This is because JES2 will always receive a compression table even though it may not be sending one.

JES2 only sends FMH3 if compression for the receiving node is indicated during JES2 initialization and the receiving node has indicated compression is accepted via the flag bit in FMH4.

RU Composition

RU Size Determination

The NJE RU size is determined by the TPDEF BUFSIZE JES2 initialization parameter. It is set in three fields during NJE session establishment: in the BIND, the FMH type 4, and the initial and response signon records. However,

JES2 only checks the RU size when receiving the FMH type 4. If the received size from the other node is different from this node's buffer size, the smaller of the two RU sizes is used.

RU Multiplexing

Prior to Version 4.1, JES2 sends only 1 type of record in an RU, but is able to receive multiplexed records within one RU. For Version 4.1 and subsequent releases, JES2 can combine the job header, data set header, job trailer, and data records within the same RU if the signon feature flags show that the receiving node supports receiving multiple headers within the same buffer.

Termination

A JES2 NJE session using SNA is normally terminated by a $P LINE command. Normal session termination may be initiated by either the primary or secondary LU node. The purpose of abnormal termination is to clear the session as quickly as possible. A session is abnormally terminated when the line is restarted ($E LINE). TERMSESS is sent by JES2 for the following reasons:

1. CLEANUP RU received in NS exit
2. Logic error - no OPNDST AUTH=(ACQ).

BSC Line and CTC Adapter Support

CTC Adapter Initialization

JES2 issues the SYN NAK as specified in Initialization.

CTC Adapter Attention and Busy Status

JES2 performs one retry in this condition, which resets the attention busy. If the attention busy occurs during the retry, it is considered a hardware error and the line is drained. If it is a temporary condition, the one retry will always clear the condition.

Error Recovery

JES2 follows the actions in BSC Error Recovery and terminates after 10 errors.

Use of Null Buffers

JES2 uses DLE ACK0 instead of null buffers for positive acknowledgement when there is no data to send.

TCP/IP

Data is transferred over TCP in a way that allows full duplex capabilities. There should always be a receive waiting for data from TCP providing there is a receive buffer available to place the data in. Also, you need to ensure to send data to TCP as soon as the data has been completely packed in the send buffer. To prevent excessive idle overhead in unnecessary TCP sends and receives, the number of consecutive ACK and null buffers sends and receives will be monitored. After 10 consecutive idle transactions, a delay of 100 ms will be performed before issuing the next transmission. Each successive idle I/O after that increases the delay interval by 100 milliseconds up to a maximum of one second.

Accounting

Accounting Records

JES2 uses SMF type 26 (job purge) records to record all successful SYSIN job transmissions; for all successful SYSOUT transmissions, JES2 uses SMF type 57

records. Because multiple SYSOUT data sets may be transmitted within a job header and trailer, this record may represent multiple SYSOUT data sets.

None of these records contains the node name of the local node, which can present problems when combining records from multiple sites.

Type 26 Records
Execution node name (and other fields related to the execution node) are not recorded in the type 26 record when the job executes at the originating node.

Type 57 Records
The following standard job header information is missing from the type 57 record cut by JES2:
- Job name
- Time and date on the reader at original node
- User identification from the common exit parameter area.

NJE Network Management Records
JES2 records the following information reflecting network events:
- 55 Network signon
- 56 Network integrity (invalid password)
- 58 Network signoff.

Other Records
The following information is missing from the type 6 record cut by JES2 and other records (such as, types 4 and 5):
- Original job number
- Original node name.

Network Account Number

JES2 uses the following scheme to derive the network account number in the job header:

1. Obtains it from the /*NETACCT JECL statement, if specified by the user.
2. If not explicitly specified in the JECL, then it may be converted from a local account number to a network account number through a local-to-network account table set up by the NETACCT JES2 initialization statements.
3. Otherwise, defaults it to the local account number.

Miscellaneous Considerations

Installation Exits

JES2 has several NJE exits:
- Exit 46 - NJE header/trailer transmission exit for SNA and BSC NJE
- Exit 47 - NJE header/trailer reception exit for SNA and BSC NJE
- Exit 56 - NJE header/trailer transmission exit for TCP/IP NJE
- Exit 57 - NJE header/trailer reception exit for TCP/IP NJE.

You can use the following exits in JES2 input processing to screen jobs received by a JES2 node for execution:
- Exit 2 or Exit 52 (job statement)
- Exit 3 or Exit 52 (job statement accounting parameters)
- Exit 4 or Exit 54 (JCL and JECL)
- Exit 20 or Exit 50 (end-of-input processing).

These exits are always taken at the execution node. They are also taken at the submitting JES2 node with the following exception:

- Exit 4 or Exit 54 (and JES2 input processing) do not scan any JCL or JECL after the /*XMIT statement.

Spool Offload Considerations

JES2 uses the NJE interface for its spool offload feature. This allows jobs and data sets to be transferred in NJE format, using existing header protocols.

JES3

Network Control

Network Connection Control Records

Path Manager Records

JES3 does not include a path manager so any connection between JES2 and JES3 must be defined as a non-path manager connection on the JES2 node. JES3 supports only SRCB record types I, J, L, and B. JES3 supports SRCB record type L only for TCP/IP. (See "Network Connection Control Records" on page 219.)

Network Addressing, Topology, and Routing

Naming Conventions (Remote vs. Userid)

To the JES3 user and operator, there are only two levels of qualification for the specification of destinations and origins:

1. Node
2. Either the remote identifier or userid.

For example, to specify the destination of a data set, a user might use the JES3 FORMAT statement in a job as follows:

```
//*FORMAT PR,DDNAME=,DEST=NODEX.SECOND
```

The destination for SYSOUT data sets from the job is node NODEX and "secondary destination" SECOND. The secondary destination SECOND could be a VM userid or a remote workstation identifier. The networking code does not know what this secondary destination is, and usually considers it to be a remote identifier. If the secondary destination must be placed in a field which, by definition, could contain either a userid or a remote identifier, and a flag set to indicate which it is, the flag that is set will indicate a remote identifier.

Use of the System Qualifier in a JES3 Complex

A JES3 complex can consist one to eight processors. One of the processors is called the global and is responsible for complex-wide data set integrity, job scheduling for the complex, processing of SYSOUT, and other functions. Also, the networking code runs only on the global processor. The other processors are called locals and are mainly responsible for running jobs under MVS. The locals communicate with the global via CTC adapters.

TSO users may be attached to any system in the complex and may submit network jobs and receive status and notify messages. The system qualifier in the NJE header indicates to which system in the complex an individual user belongs. When a TSO user submits a job, a value is placed in the job header qualifier field NJHGORGQ. Other nodes may place this value in the NMRTOQUL field in the NMR for STATUS or NOTIFY messages.

The qualifier value (1-8) corresponds to the relative position of the main processor control (MPC) table in the MPC chain for the appropriate system (as determined by the sequence of MAINPROC statements in the initialization deck). For TSO NOTIFY messages, a value of 0 is invalid and the message will go to the default NJE message class and not the TSO user. For jobs that are not submitted from TSO, NJHGORGQ will be zero.

Parallel Links

Zero to three lines may be defined for each node in the network. One or two logical senders will be generated for each line (unless zero lines have been defined, in which case no logical senders will be generated). See "Stream Support and Control" on page 232 for a description of logical senders.

Dynamic Route Changes

A JES3 operator may modify the routes with a command. (There is no dynamic path manager.)

Commands and Messages (NMRs)

JES3 Console Service

JES3 supports MCS consoles in a JES3 complex. All JES3 consoles for remote workstations, and the NJE console, are Extended MVS consoles. Messages to JES3 consoles, and MCS consoles within the operation of JES3, are manipulated using console services with the eight byte console name. Messages destined for MCS consoles specify MCS routing and descriptor codes. (JES3 maps MCS routing codes to JES3 message classes.) JES3 networking supports only JES3 consoles as networking consoles. Networking messages, if not sent to a specific destination or console, are sent to the default NJE message class specified at initialization.

JES3 Use of the NMR

A field exists in the NMR which, for messages, designates the destination. This 8-byte field (NMROUT) is redefined several times to indicate a userid, a remote, or a console destination. The console destination usually has meaning only for command responses to a command that originated at a specific console.

Commands

Formatted (Global) Commands

JES3 supports most global (formatted) commands for input. The ROUTE command is ignored. These commands never originate at a JES3 node. Formatted commands are translated into equivalent JES3 commands if such exist.

NMR Command Length Restriction

A command sent to JES3 is placed into a buffer and inserted into the system with a JES3 INTERCOM macro. The INTERCOM buffer can be a maximum of 80 bytes. In addition, certain keywords are added to the commands, with the result that commands greater than 59 bytes long are rejected. To prevent needless command rejection, JES3 removes any trailing blanks from the command to reduce the length prior to checking for the 59-byte limitation.

Command Authorization

Only the following commands may be sent to a JES3 node for execution; all other commands are invalid:

- *I J={job name or job-number}

  display job status by job name or job number

- *I B

display statistics for number of jobs waiting to be processed

- *I Q[,N={xx|ALL}]

  display status of all jobs

- *F J=jobno,{H,R,C,CP,CO}

  hold, release, cancel, or cancel and print a job

The above inquiry commands will only display information about jobs submitted by the node issuing the command. The modify command will only modify jobs submitted by the node and userid/remote id issuing the command.

The installation can provide an exit (IATUX35) that will accept other commands, or place further restrictions on the commands listed above. This exit replaces JES3 standard validation/authorization of the command. (In addition, exit IATUX18 may be used for additional authorization checking.)

Note: For Version 3.1.3 and all subsequent releases, JES3 supports the use of all commands.

SYSIN (Job Input) Transmission

Multi-leaving Header Record Expansion

NJE control records (headers) are not padded with blanks when received by JES3. If a received header has had trailing blanks truncated by the sending system, the blanks are not restored and unpredictable results could occur.

Data Set Header (RCCS)

JES3 does not support receipt of the record characteristics change section (RCCS) that accompanies SYSIN data greater than 80 bytes. If received, this section is ignored. It will not be store-and-forwarded.

Acceptable Job Streams

Requirement For Two Job Statements

To transmit a job from a JES3 system for execution at another node, the following JCL must be submitted:

```
//abc    JOB  xxxxx        first job statement
//*NETACCT  ....           (optional)
//*ROUTE XEQ  nodename
//xyz    JOB  yyyyy        second job statement
  .
  .
  .
//
```

The first job statement and the NETACCT and ROUTE XEQ statements are stripped off at the submitting node; what is transmitted is the second job statement and whatever follows up to the next job statement or end-of-file. This places the following restrictions on the user who wants to submit jobs from JES3:

- Only one job can be submitted for execution elsewhere.
- Submitted jobs must begin with a statement that looks like a MVS JOB statement.
- The user must be familiar with the requirements of the execution node for job statements (accounting syntax, etc.).

- Users submitting jobs from TSO must remember to code NJB in place of JOB on the second job statement, or the second job statement will signal the beginning of a new job.

Acceptable Job Streams

JES3 does not support more than one job between a job header and a job trailer.

Notification

JES3 sends notification messages at the following times during SYSIN job transmission and execution:

- When a job has finished transmission at an intermediate node (this does not include the submitting node).
- When the job arrives at the execution node.

See "Notification" on page 232 for notification during SYSOUT processing.

SYSIN Job Routing

Use of System Qualifier

If the job is to be executed on a VSE/POWER system, then there is the capability to ensure that a certain system in a shared-spool complex executes the job. The system qualifier is specified on the * $$ JOB statement and is contained in the VSE/POWER section.

Use of the Userid

The userid field on the XDEST parameter can be used to route a job to a VM system for execution. If VSE/POWER is the target execution node, then the NJHGXEQU field is ignored.

Operator Rerouting

VSE/POWER operators may change the execution node and userid for any job on the VSE/POWER queue by use of the PALTER command.

Jobid Assignment

In VSE/POWER, the jobid is a combination of job name and job number. The VSE/POWER job number is a halfword binary number assigned by VSE/POWER when the job first enters the system. This number may not be unique within the VSE/POWER system. VSE/POWER uses the job name as the primary identifier for all system control, and the job number as a secondary identifier.

When a job is transmitted from one system to another, the receiving system will assign the original job name (from the job header) to the job that is being received. A new job number is always assigned to received jobs or SYSOUT. The new and original job numbers are displayed in the messages that are issued to acknowledge transmission of jobs or SYSOUT to the next node. The job header always contains the original (input system's) jobid. The jobid assignment procedure outlined above is followed by both job receivers and SYSOUT receivers.

If VSE/POWER is the execution node and receives a job which contains a * $$ JOB statement as its first record after the job header record the * $$ JOB statement is processed at receive time and influences the attributes of the receive job. This results in assigning a new job name, if the * $$ JOB statement contains a job name (specified using the JNM operand). If no job name is specified in the * $$ JOB statement, the received job has the original job name (specified job header control record). If the job contains more than one * $$ JOB statement, the received job is split into several VSE/POWER jobs, each with a different job name (if specified in the * $$ JOB statement) and a different job number.

If the SYSOUT has been segmented, the job name will remain the same as the job and each segment will retain the original job number.

SYSOUT (Job Output) Transmission

Job Output Routing

In SP 1.3.4, JES3 uses the submitting node as the default destination for output routing. Prior releases used the execution node as the default routing node.

Store-and-Forward Transparency

Spanned Record Support
JES3 supports spanned records.

Data Set Header

Multiple Data Set Headers
JES3 always splits jobs received with multiple data set headers into separate jobs. Multiple data set headers for one data set do not exist in jobs sent by JES3.

Notification

JES3 sends notification messages for job output:

- When output is queued for transmission at the execution node.
- When output has finished transmission at an intermediate node.
- When output arrives at the print (destination) node.

See "Notification" on page 231 for notification during SYSIN processing.

Job Output Routing

Use of System Qualifier
See "Network Addressing, Topology, and Routing" on page 228 for details.

Use of Userid
See "Network Control" on page 228.

Default Output Routing
JES3 sets the default output node to the submitting node.

Operator Re-routing
SYSOUT data sets on the transmission or output queues can be re-routed to another userid/node.

Stream Support and Control

Senders and Receivers
In JES3, the transmitters and receivers are not discussed in the same terms as elsewhere in this document. One to six logical senders per BSC node are generated at initialization time, which are analogous to transmitters. The number generated depends on other options, such as the number of lines for each node and whether or not multi-streaming (discussed below) is to be used. Each sender is capable of transmitting both jobs and SYSOUT (normal mode of operation) or only one of the two (multi-streaming). Only one receiving function exists, and it is not generally referred to as a receiver in JES3 publications. The receiver processes each transmission buffer as it is received, so it does not care what it is receiving (jobs or SYSOUT).

In JES3, senders and receivers for TCP/IP nodes are defined using the JOBTRANS, OUTTRANS, JOBRECV, and OUTRECV parameters on the NJERMT initialization statement. The sum of job and output receivers is limited to 8; similarly the sum of job and output transmitters is limited to 8. Each individual value can be an integer between 1 and 7.

Multiple Streams

JES3 supports one stream for job transmission and one stream for SYSOUT transmission. In each case, the stream supported is stream 1 (RCB X'98' or X'99'). However, with multi-streaming, one job stream and one SYSOUT stream can be active on a line simultaneously. To accomplish this, two senders are generated for each line - one to send jobs and one to send SYSOUT. The line manager alternates the two streams for transmission.

Operator Control of Lines

The JES3 operator can control individual transmitters (senders). Limited transmission control is also available in that the operator can place a remote node in hold status; all subsequent jobs scheduled for that node will be held. The operator may later release the node from hold status, then release the individual jobs.

Receiver-initiated Processing

JES3 uses an RCB of X'D0' to indicate to a remote node that this node's receiving function has been turned on. The scenario for this function is as follows:

1. The operator at NODE1 issues the command to stop data reception on the specified line: *S LINEX,NORCV.
2. NODE2 sends a request permission to initiate stream sequence (RCB X'90').
3. NODE1 responds with negative permission (RCB X'B0') due to the NORCV (no receive) in effect.
4. NODE2 varies the sender for the job offline and places the job in specialized reschedule status. (The job will wait for the associated device - sender, in this case - to become available.)
5. If multi-streaming is in effect, another job of the opposite type (job vs. SYSOUT) could ask for permission to start transmission, with the same result. This could cause more than one sender to be varied offline.
6. Some time later, the operator at NODE1 allows data reception to take place by specifying the command: *S LINEX,RCV.
7. NODE1 sends a receiver initiated sequence (RCB X'D0') to NODE2.
8. NODE2 then varies all the senders to NODE1 online. Jobs in specialized reschedule status that are waiting for the senders to become available will now be available for scheduling.
9. NODE2 can reissue the request permission to initiate a stream. If all else is well, the job will be transmitted.

JES3 always uses an SRCB of X'D7' to accompany the RCB of X'D0'. This is, in itself, meaningless, and is only present because an SRCB should always be associated with an RCB. Because JES3 only has the concept of a single receiving function to handle both jobs and SYSOUT, "receiver on" cannot be associated with either. Hence, it is impossible to place the RCB of the stream to be initiated in the SRCB that goes with the X'D0' RCB.

SNA Support

JES3 does not support SNA as a communication medium.

BSC Line and CTC Adapter Support

CTC Adapter Initialization

JES3 checks the data returned from the sense CCW and if it is X'00', a control-read sequence is issued; if it is X'01', a read-sense-write-control-read is issued; otherwise, a write-control-read is issued.

CTC Adapter Termination Deviations

For SP 1.3.4, JES3 does not send a SIGNOFF record to terminate a CTC adapter connection. Instead, a negative acknowledge (NAK) is transmitted and the connection is terminated. For SP 2.1.5, JES3 sends a signoff to terminate a CTC adapter connection.

CTC Adapter Attention and Busy Status

Some systems (RSCS, for example) place a TIC command between the standard CONTROL and READ commands for the CTC adapter. This can cause the remote node's channel program to terminate with ATTENTION plus BUSY status. If JES3 receives an ATTENTION plus BUSY status, the following occurs:

If the last I/O issued was a SENSE command, then it is reissued. If the last channel program terminated on the first CCW, and it was a SENSE command, then the channel program is retried. Otherwise, a SENSE command is issued. If the SENSE indicates that either a CONTROL or READ is pending from the remote node, the last channel program is restarted from the WRITE command. If none of the previous conditions is met, then the connection is terminated.

CTC Adapter Zero Byte Operation

It is possible for a READ or WRITE operation on a CTC adapter to terminate with successful ending status, but with a CSW residual byte count equal to the original byte count (i.e. a zero-byte operation). This can occur if a system reset is issued on a remote node. When JES3 detects this condition, the connection is terminated immediately without sending a negative acknowledge. (In JES3 1.3.4, NAK is not sent; in 2.1.5, a signoff is not sent.)

Error Recovery

BSC Initialization Error Recovery

JES3 differs from BSC Initialization Error Recovery in the following ways:

1. If data other than SOH ENQ, DLE ACK0, or NAK is received, the line will be canceled.
2. If command reject occurs on a write, the line will be terminated. Otherwise, the operation will be retried.
3. If intervention required occurs on a read, a retry will be performed.

JES3 will retry on Bus Out Check and Equipment Check before terminating.

BSC Error Recovery

JES3 follows the actions in BSC Error Recovery with the following exceptions:

1. A retry is attempted for unit exception on other than a read or write.
2. On command reject for a read, a NAK will be sent. On other than a read, retry is attempted.
3. On a unit check other than a command reject or intervention required, if the CCW is a read, a NAK is transmitted; otherwise a retry is performed.

JES3 terminates after 20 consecutive errors.

BCB Handling

JES3 has the following deviations from the correct BCB recovery procedures in BSC Error Protcols:

1. When in state 1 or 3 and a NAK is received (E3), if the line is a CTC adapter, it will be canceled. Also, a NAK counter is maintained (similar to the retry limit), such that after 10 consecutive NAKs are received, the line will be canceled.

2. When in state S2 and a duplicate BCB is received, that data block is ignored, but the last block is not re-transmitted.

3. When in state S2 and a missing BCB is detected, the line is canceled.

4. During signon, a duplicate BCB causes the line to be canceled.

Null Buffers

JES3 does not send null buffers as a response. DLE ACK0 is always sent as an acknowledgement.

Signon Deviations

JES3 does not negotiate the buffer size in the SIGNON records. If a mismatch occurs, the line is terminated.

Accounting

Use of Job Header and Trailer Fields

There are several fields in the NJE job header that relate to the recording of SMF information. For example, NJHGETIM, which is the estimated job execution time, and NJHGELIN, the estimated output print lines. The job trailer is also composed entirely of accounting type information. JES3 does not use most of the job trailer fields, and uses only some of the job header fields in recording information for SMF. Some of the accounting information that is recorded by JES3 is culled from other sources, such as internal control blocks.

Accounting Records (SMF)

System management facilities (SMF) is a function of MVS that allows the collection and recording of various types of system and job-related information. This information is recorded in the form of a number of different records, which are numbered. Installations can process the SMF records with any number of application programs to analyze the data, produce reports, etc.

Two SMF records (types 26 and 57) are recorded by JES3 that contain network-related information. The type 26 record is produced at job termination time and contains various job summary information. The type 57 record is produced for each transmitted job (or SYSOUT) after successful transmission and contains summary and resource usage information related to the network processing of the job.

There are no records produced in JES3 that record information of a network management nature.

Network Account Number

Users submitting jobs from JES3 can specify certain accounting information they wish to accompany their job. This is accomplished by using the //*NETACCT control statement in the jobs JCL stream. See "Acceptable Job Streams" on page 230 The information that can be supplied on this statement is as follows:

```
PNAME  - programmer's name        (1-20 characters)
ACCT   - network account number   (1-8 characters)
USERID - userid for origin or notify  (1-8 characters)
```

```
DEPT  - user's department        (1-8 characters)
BLDG  - user's building          (1-8 characters)
ROOM  - user's location          (1-8 characters)
```

This information is placed in the NJE job header built for the job.

Miscellaneous Considerations

Propagating Job Name

The JES3 procedure for submitting SYSIN to execute at another node (see "Acceptable Job Streams" on page 230) introduces two job names:

- The job name on the outer, or first, job statement.
- The job name on the inner, or second, job statement.

The submitting node uses the outer job name for all accounting and reporting information to the job stream. This outer job name sets NJHGJNAM before transmitting the SYSIN job. Once the execution node receives the SYSIN job, the inner job name becomes the source for all accounting and reporting information to the job stream. The inner job name remains the source as long as the SYSIN job or its output remain on the execution node.

If the output produced by the SYSIN job travels across the network, both the original job header and job trailer are reused. As a result, the outer job name is used for accounting and reporting on the destination node.

Use of Utility Jobs in JES3

JES3 uses "utility jobs" to process network traffic. A utility job is created when:

- An executing job produces output destined for another node.
- A job is submitted for execution on another node.
- A job or SYSOUT data is received for either local processing or store-and-forwarding.

A "job" in the JES3 sense is a set of well-defined, schedulable elements that define work to be done on behalf of some entity, be it a "real" job, or, in the networking case, a collection of network data. Utility jobs are created to contain specific scheduler elements to process the network data. This results, however, in a network job having more than one job number during its life on a JES3 node. For example, when a job is received for local execution, a utility job is created and assigned a job number. When the job actually executes, it is assigned another job number. If the job's output is to be transmitted, another utility job is created to do this and assigned another job number.

Installation Exits

JES3 networking code contains several installation exits. Some allow the installation to modify fields in the job or data set headers for received data, and some allow modification of the headers before transmit. A brief description of each of the exits follows.

- IATUX35 - allows the installation to perform special command validation/authorization for commands that execute locally.
- IATUX36 - allows the user the pull accounting information from the first segment of the job header for received jobs and SYSOUT.
- IATUX37 - allows the modification of the first segment of the data set header for received SYSOUT data sets.
- IATUX38 - allows the user to perform special processing of SYSOUT classes for received data sets.

- IATUX39 - provides access to the data set header constructed by this node for the SYSOUT from locally-executed jobs. The exit is called prior to the transmission of the SYSOUT.
- IATUX40 - allows modification of the job header that is built for locally-submitted jobs.
- IATUX43 - allows access to the entire job header for SYSOUT data sets from locally-executed jobs that are to be transmitted. The exit is called at intermediate nodes.

RSCS

Network Control

Network Path Management

RSCS does not support a path manager. Any connection between RSCS and JES2 must be pre-defined by a JES2 CONNECT statement. RSCS will not forward Network Path Manager (NPM) topology records. See "Network Connection Control Records" on page 219

Network Connection Control Records

RSCS supports only (SRCB) type I, J, and B records.

Network Addressing, Topology, and Routing

Remotes and Userids

VM makes extensive use of user identifications (userids) to identify virtual machines. Therefore, RSCS has record of the userids of originators and final receivers of network data. It also handles data for remote workstations. However, a "remote", as defined for JES2 and other NJE systems, is not used in exactly the same way in a VM environment.

The VM userid is not the same as a JES2 remote. RSCS uses the remote name as a link (or line) name. It does not replace a userid in a network address and RSCS uses it more like a node name than a userid. RSCS is able to handle the same remote identifiers for workstations that are used by other NJE systems with no restrictions on names. However, a remote and userid cannot have the same name on a single node. Also, because the networking control blocks (headers and NMRs) do not contain fields for both types of identifiers, RSCS often must use for userids, fields defined for remotes.

Parallel Links

RSCS supports links to an adjacent node only in Version 3.

Dynamic Route Changes

The RSCS operator may dynamically change links (direct connections to adjacent nodes) and routes (indirect paths to other nodes) through the use of operator commands.

Commands and Messages (NMRs)

Commands

Formatted Commands

RSCS networking does not support formatted commands that originate at its node. It will pass on formatted commands (received from another NJE node) correctly. It also will translate formatted commands into equivalent RSCS commands according to the setting of the NMRFOP field as follows:

1. NMRFOPD (display job) - Query file spool identifier
2. NMRFOPC (cancel job) - Purge spool identifier
3. NMRFOPA (release job) - Change spool identifier nohold
4. NMRFOPH (hold job) - Change spool identifier hold
5. NMRFOPR (route job) - Transfer spool identifier to local userid.

All bits are store-and-forwarded and the NMRFLAGT field is always turned on when a message contains a destination userid.

Command Authorization

Remote system operators may issue commands to RSCS to affect the link between RSCS and their system (without authorization). (The AUTH statement was not intended to be used to authorize system operators at remote systems.)

SYSIN (Job Input) Transmission

Store-and-Forward Transparency

Segmented Headers
  RSCS supports segmented job headers, data set headers (for both SYSIN and SYSOUT), and segmented job trailers.

Notification

RSCS sends a notification message (DMTxxx147I) when the file is successfully transmitted on a link. DMTAXM104I is sent when the file is spooled to the user.

Spool File Identifier Assignment

CP assigns a new spool file identifier when it receives a file. No attempt made to re-assign the same number it had on the originating node (in NJHGJID).

SYSOUT (Job Output) Transmission

Store-and-Forward Transparency

Segmented Headers
  RSCS supports segmented headers.

Print File Transparency
  RSCS modifies store-and-forward data in the following ways for SYSOUT print files only:
  1. For Version 2.2, RSCS converts the ASA carriage control to machine store-and-forwarded. The NDHGRCFM field in the data set header is changed accordingly. Records sent with machine carriage control character are store-and-forwarded without change.
  2. Records (with or without carriage control of any type) sent as spanned records are store-and-forwarded without change.
  3. For files sent to VM users--those not store-and-forwarded-- RSCS truncates any characters in print SYSOUT records that exceed the number allowed for the printer whose type is used to store the file on CP spool. (The method in which this printer type is determined is described in detail in section RSCS/CP Spool Interface Considerations.)

  At the destination node, records are truncated as described above.

Punch File Transparency
  RSCS handles incoming SYSOUT punch files as follows:

1. If NDHGRCFM indicates that a punch file contains ASA or machine carriage control, RSCS replaces the carriage control in each record with a machine punch operation code (opcode). For store-and-forwarded files, carriage control is forwarded properly.

2. If NDHGRCFM indicates no carriage control, the file is forwarded without change (as 80-byte records with no carriage control).

RSCS/CP Spool Interface Considerations

RSCS does not do its own spooling, but relies on the CP component of VM to create and manage spool files. These files may be stored on spool as either virtual print or virtual punch files. Virtual punch files can contain up to 80 bytes of data and, within the spool, they also contain a punch opcode of X'41'. Virtual print files that RSCS uses are 1403, 3211, and 3800. (Note that CP supports more virtual print types than RSCS.) Each print record is stored with a machine carriage control (or machine opcode) and is limited in its length by the maximum number of records a real device of that type can handle, based on the data set header.

RSCS truncates--only at the destination node--any records that conflict.

Table 71 is a table of virtual device types that are defined for different incoming NJE file types:

Table 71. RSCS Virtual Device Types

NJE data	CP device type
SYSIN SYSOUT:	Punch - limited to 80 characters
If NDHGF2PU is on	Punch - limited to 80 characters
If NDHGF2PR is on	Print - • 1403 if NDHGLREC is 133 or less (132 if NDHGRCFM indicates no carriage control) and file does not meet the 3800 criteria. • 3211 if NDHGLREC is greater than limits specificified for 1403 and • file does not meet the 3800 criteria. • 3800 if header contains a 3800 subsection withNDHAF1J flag on • (3800 file processing is discussed in section below in greater detail.)

Note: When a data set header contains an RSCS subsection, RSCS uses the originating device type in this section to determine the VM device type. Therefore, the SYSOUT part of this table only applies to files which do not originate on a VM system.

Once RSCS has determined the device type to use for a file at header processing time, the type will not be changed even if the data records themselves have different characteristics. Once the file is stored in CP spool, it will not be changed if it is forwarded to another non-VM NJE system. Hence, if RSCS receives a file with individual data records of 150 but the NDHGLREC field says 132, the file will be stored as a 1403 file and all characters after the 132nd character in each record will be truncated for printing on the RSCS node if an end node.

3800 Files

There are essentially two types of 3800 files:

• Regular print files which are meant to be printed on a 3800 printer (indicated by the fact that their data set header contains a 3800 section)

- Virtual 3800 files that are currently only understood by a VM system and thus are only originated by RSCS.

Virtual 3800 files are also sent with a 3800 section in their data set headers. For purposes of this discussion, virtual 3800 files may be distinguished from regular 3800 files by two characteristics: First they contain actual 3800 CCW opcodes as well as machine carriage control characters and, secondly, they are the *only* type of file with a 3800 subsection that are originated by RSCS.

When virtual 3800 files are created by an RSCS node, they are sent with both an RSCS subsection and a 3800 subsection in the data set header. RSCS sets the virtual device type in the RSCS subsection to be virtual 3800 so another RSCS system can process the file correctly. Thus, RSCS can always recognize a virtual 3800 file it has created.

RSCS creates a virtual 3800 file out of any 3800 print file that has the OPTCD=J flag on in the 3800 section of the data set header. This occurs for store-and-forward files and for files for which RSCS is acting as an end node. RSCS then removes the TRC byte from each record at the destination node and inserts select CCWs that cause the proper character arrangement table to be selected. If such a file is printed on a real 3800 printer on the receiving node or another VM node, the printout should be as the originator intended. If the same file is sent back to another non-VM NJE system, the TRC bytes are re-inserted and the select CCWs removed.

RSCS changes 3800 files that have ASA carriage control to machine carriage control in the same way as other types of print files. For Version 3, this restriction is removed. (see "Store-and-Forward Transparency" on page 238).

Spanned 3800 records are not unspanned when they pass through an RSCS node. Records are unspanned only when the files are printed on the node RSCS is running on.

Support of SRCB X'B0'

RSCS supports the SRCB execpt on those VM/XA SP1 and SP2.0 systems that do not have the AFP feature installed. In these cases, RSCS continues to throw these records away when it tries to print a file that contains them.

Data Set Header

Use of CP TAG Command

RSCS has no direct JCL equivalent to allows users to specify parameters for jobs or data sets. However, RSCS allows a file originator a limited method of specifying certain fields in the data set header RSCS sends with a file. The originator can use this method to either override default settings of fields used by RSCS or to specifically indicate values to be included for fields RSCS does not set. The CP TAG command, described in *RSCS Operation and Use Guide*, is used to specify these parameters. The TAG command can only be used for fields in the data set header.

Multiple Data Set Headers

The protocol allows multiple data set headers to appear within a job. These headers may indicate different routing for a single data set or many data sets may be grouped in the job because they are all output from its execution. In the second case, each data set is preceded by a data set header. RSCS handles these cases as follows:

1. When a single data set is received by RSCS with multiple destinations, the data set is store-and-forwarded as separate files (one for each destination). RSCS will only make separate files if the data sets involved are going out on different links.

2. When any data set within a job has different characteristics from another in the same job, each data set is store-and-forwarded as a separate file.

3. Spin data sets (flag NDHGF1SP on) are always stored as separate files.

Note: When data sets are store-and-forwarded as separate files, each file has the original job header and job trailer.

RSCS uses the LIST processor to generate multiple data set headers.

Notification
RSCS sends a notification message (DMTxxx147I) when the file is successfully transmitted on a link. DMTxxx147I is sent when the file is spooled to the user.

Special Processing

Print File Processing
If RSCS is the destination, it will default CCWs it does not understand.

Punch File Processing
Punch files may be sent either as 81-byte records with machine carriage control, or as fixed 80-byte records without carriage control. This is specified by the file originator on the TAG command using the PUNCC option.

Job Output Routing

Operator Rerouting
The TRANSFER command can be used by both the RSCS operator and the file originator to re-route files to other nodes. In Version 3, the file target user can use the TRANSFER command.

Stream Support and Control

Abnormal Termination Protocol

Version 2
When RSCS receives permission denied (when it asks to initiate a stream) it stops using that stream until either the RSCS operator resets that link, or until a receiver ready is received (X'D0') for that stream. Receiver cancel is still considered to be a file, rather than stream control record.

Whenever RSCS Version 2.1 initiates abnormal termination as either a receiver or a sender, an operator message is sent to the remote system operator. Additionally, the file originator is also informed.

Note: RSCS responds to the D0D7flow to restart all streams. In Version 3, RSCS sends the D0D7 flow.

SNA Support

Data Flow Control
The only data flow control (DFC) function supported by RSCS is RSHUTD. If any other function is received, RSCS will terminate the session with a CLSDST or a TERMSESS.

Bind Parameters
RSCS creates the bind from the bind image VTAM returns in the CINITRU, which is passed to the RSCS LOGON exit after a SIMLOGON has been issued.

VTAM gets this bind image from the logon mode tables associated with the remote system's VTAM. Through use of RSCS START command parameters, it is possible to specify which logon mode table entry is returned to RSCS, but the logon mode table to be searched must be specified when defining the remote system to its local VTAM.

Function Management Headers

Compression
RSCS does not support compression. If an FMH3 is received, RSCS will terminate the session.

RU Composition

RU Size Determination
The transmission block size is defined in RSCS START parameters for each link. This value is used in both the signon record, and FMH type 4. On a received FMH4, the transmission block size is checked to verify that it is 300 or greater. The actual transmission block size is the smaller of the two values specified in the signon records.

BSC Line and CTC Adapter Support

Wait-a-bit Processing
RSCS always uses null buffers. RSCS accepts and processes receives buffers that have wait-a-bit set. It responds with a null buffer without any response time difference (no special delay).

Error Handling

Initialization Error Recovery
RSCS does not follow the procedure outlined in BSC Initialization Error Recovery. It retries any type of error 15 times and only terminates after this many errors or if there is no I/O device defined for the line. All retries involve rewriting the SOH ENQ.

Normal (Non-initialization) Error Recovery
RSCS follows the actions in BSC Error Recovery except for data check, data overrun, or lost data. If these errors happen on a write, the data is re-transmitted. If they happen on a read, a Negative Acknowledge is sent.

Use of Null Buffers

Use of the PREPARE Extension
RSCS uses the PREPARE extension whenever it is indicated on the SIGNON record. You can use the PREPARE=parameter to suppress this extension.

Accounting

Accounting Records
For information about RSCS accounting records, see *RSCS Exit Customization* mentioned on /RSCSB/.

Miscellaneous Considerations

Installation Exits

There are installation exits in RSCS to examine headers and trailers and to modify or add user sections from files that originate at RSCS. See *RSCS Exit Customization* on /RSCSB/ for more information.

VSE/POWER

Network Control

Network Connection Control Records

VSE/POWER does not have a network path manager, so any connection between VSE/POWER and JES2 must be pre-defined as a non-path manager connection on the JES2 node. See "Network Connection Control Records" on page 219

VSE/POWER supports only SRCB types I, J, and B.

Network Addressing, Topology, and Routing

Naming Conventions

VSE/POWER supports up to 250 remotes (RJE workstations) since Version 2.2; all versions support an unlimited number of nodes. Remotes are defined during VSE/POWER generation and are loaded during VSE/POWER initialization. Nodes are also pre-assembled and are loaded during the initialization phase. Remote workstations are referenced by Rnnn, where nnn must be three numeric characters.

All members of a shared spool configuration must have the same node name in the network. Members are distinguished by the node name qualifier.

Parallel Links

Multi-trunk is NOT supported.

Dynamic Route Changes

A new network definition table (NDT) can be loaded by the PLOAD operator command and used to dynamically alter the network routing and topology.

Network Routing

VSE/POWER does not support the JES2 Network Path Manager. All routing information must be pre-defined by the user with the aid of the PNODE macro. A route table is loaded into storage when VSE/POWER is initialized. This table may be dynamically replaced while VSE/POWER is active by means of the PLOAD PNET operator command.

If a job or OUTPUT is received from the network and the destination is not found in the network definition table, then the job/OUTPUT is put into HOLD status in the transmission queue and VSE/POWER tries to send a message to the originating node and userid.

Alternate routing is supported by VSE/POWER. If the primary route is not active, but the secondary route is active, then the secondary route will be chosen. Both routes cannot be used together. As soon as the primary route is available, all new transmissions will be sent over the primary route and the secondary route will cease to be used as soon as the current transmission finishes.

Transmission Recovery Techniques

Jobs and OUTPUT data are handled on a store-and-forward basis. Responsibility for a unit of work does not pass from a transmitting node until it receives positive acknowledgment to the end of file that follows the job trailer record. If transmission is broken off before that point, the receiver discards any data it may have received so far and the transmitter queues its work for re-transmission from the beginning.

Transmission may be discontinued voluntarily or involuntarily:

- Voluntarily, by either the transmitter or receiver.
- Involuntarily, in the case of a line disconnection.

In either case, the receiving system discards the job or SYSOUT data that it has partially received. In the case of voluntary termination of transmission, the transmitting system will always requeue the job or SYSOUT data for re-transmission.

Commands and Messages (NMRs)

Commands

VSE/POWER builds an unformatted nodal message record (NMR) as a result of the PXMIT operator command. The syntax of the command is not checked by the transmitting system. The destination of the command is checked by the transmitting system and the command is rejected if the destination is not known.

Store-and-Forward Transparency

Commands and messages are transmitted to their final destination only if there is a path available. If the primary route is not available, then any alternate route may be chosen. If no route is available, then commands are thrown away but a message is sent to the originator regarding the inability to deliver the command. If the route is not available, messages are discarded without informing the originator.

Formatted (Global) Commands

Formatted commands are not supported by VSE/POWER.

Command Authorization

VSE/POWER supports three types of command authorization, which must be specified on a node basis in the node definition table:

- Network - The node has the authority to control many functions that the local operator can control. Transmitters/receivers can be started remotely and transmission flushed. No control is possible for local I/O devices.
- Job - The node can only manipulate jobs or output that either originated at or is destined for that node.
- Nojob - The node is not allowed to manipulate anything within the network. Users can still display queues on other nodes.

Message Transmission

Messages are created by VSE/POWER in the following instances:

- In response to a unformatted command.
- Via a PBRDCST command issued either by the operator or by any interactive user having authority.
- For notification at end of execution of a job or on receipt of jobs or output.

SYSIN (Job Input) Transmission

Store-and-Forward Transparency
All sections in the job header are forwarded without change. As an intermediate node, no sections are added to the job header. The operator at the store-and-forward node can change the execution node of the job and this will be updated in the job header.

Job Header
The job header is created as soon as the job enters the VSE/POWER system. The header is used to store job-specific data (such as execution destination, default print and punch routes, etc.). Segmented job headers are supported up to the maximum allowed by the architecture.

VSE/POWER Section
The VSE/POWER section contains information specific for VSE/POWER's usage. For example, the due date information which specifies the date and time a job should be processed.

Job Trailer
The job trailer is built as soon as the job enters the VSE/POWER system. Because VSE/POWER doesn't use a lot of the counts that JES2 and JES3 use for scheduling purposes, the majority of the fields in this record are not used and are set to defaults.

Data Set Header
Record Characteristics Change Section. This section is put into the job stream as soon as VSE/POWER determines that the stream may contain records greater than 80 bytes. This means that when a SYSIN stream is read from a diskette with a header that says 128 bytes, the record characteristics change section is sent directly after the job header. There is also a VSE/POWER section appended to this section to allow the transmission of the diskette address used when reading the file.

Notification
When a job is first read in by VSE/POWER, the node and userid for notification are stored in the job header record if "NTFY=" was specified on the JCL JOB statement.

If the job header contains a notify userid and the job is transmitted from its originating node for execution on another node, the job transmitter of each intermediate node will issue a message to the user indicating that the job has been transmitted.

When the job completes execution, the execution processor on the execution node will issue a notification message to the user specified by the notify userid field and by the originating node field.

SYSIN Job Routing
Use of the System Qualifier

If the job is entered on a VSE/POWER system, then there is the capability to ensure that a certain system in a shared-spool complex executes the job. The system qualifier is specified on the * $$ JOB statement and is contained in the VSE/POWER section.

Use of the Userid: The userid field on the XDEST parameter can be used to route a job to a VM system for execution. If VSE/POWER is the target execution node, then the NJHGXEQU field is ignored.

Operator Rerouting: VSE/POWER operators may change the execution node and userid for any job on the VSE/POWER queue by use of the PALTER command.

Jobid Assignment

In VSE/POWER, the jobid is a combination of job name and job number. The VSE/POWER job number is a halfword binary number assigned by VSE/POWER when the job first enters the system. This number may not be unique within the VSE/POWER system. VSE/POWER uses the job name as the primary identifier for all system control, and the job number as a secondary identifier.

When a job is transmitted from one system to another, the receiving system will assign the original job name (from the job header) to the job that is being received. A new job number is always assigned to received jobs or SYSOUT. The new and original job numbers are displayed in the messages that are issued to acknowledge transmission of jobs or SYSOUT to the next node. The job header always contains the original (input system's) jobid. The jobid assignment procedure outlined above is followed by both job receivers and SYSOUT receivers.

If VSE/POWER is the execution node and receives a job which contains a * $$ JOB statement as its first record after the job header, the * $$ JOB statement is processed at receive time and influences the attributes of the received job. This results in assigning a new job name (specified in JNM operand). If no job name is specified in the * $$ JOB statement, the received job has the original job name (specified in the job header control record). If the job contains more than one * $$ JOB statement, the received job is split into several VSE/POWER jobs, each with a different job name (if specified in the * $$ JOB statement) and a different job number.

If the SYSOUT has been segmented, the job name will remain the same as the job and each segment will retain the original job number.

SYSOUT (Job Output) Transmission

NJE Unit of Transmission

Within NJE, a unit of transmission is defined as being everything between a job header and a job trailer record. It may consist of several data sets of different characteristics.

Within VSE/POWER, spooling is necessary to differentiate between print and punch output because there are separate queues for each type. This means that if output is received containing mixed SYSOUT types (print and punch data), it is split.

Store-and-Forward Transparency

All SYSOUT data sets are forwarded through VSE/POWER transparently. A maximum length of spanned record as defined in the architecture is supported. No sections are added to the record by VSE/POWER, although the user may add a user section.

Spanned Records with Carriage Control: VSE/POWER supports spanned records with carriage control, both as intermediate node and as destination node.

Trailing Blank Truncation: Trailing blanks are truncated from all data, both on spool data and on transmission. However, the original record length is retained. The blank truncation can be suppressed and all blanks are written to the spool file and transmitted by using a BTRNC operand.

Print File Transparency: VSE/POWER Version 2.1 converts all ASA carriage control to machine carriage control.

Job Header

The job copy count field of the job header is not used by VSE/POWER. This means that a user should use other means of specifying that they want to print multiple copies of the output. The default print and punch destination and user ID fields in the job header are not used for SYSOUT because their information has already been copied to the data set headers.

Job Trailer

The majority of the fields in the job trailer are not used by VSE/POWER and are set to default values.

Data Set Header

The data set header contains the information necessary to handle SYSOUT data sets on the receiving system. It contains three types of information:

- Identification (data set number)
- Routing control (destination node name and remote)
- Data set characteristics (output class, copy count, FCB name).

Data set headers exist as soon as any job has executed on a VSE/POWER node or SYSOUT is received from another node. They are built from information contained in the job header and in the VSE/POWER queue record, which, in turn, is built from JECL statements, if present. The data set header is always present on the VSE/POWER spool, even if the data set is not intended for transmission. The data set header record is flagged as an internal record, so it is ignored by local print/punch processors and external writers at the destination node. Information is retrieved from the data set header record by the print/punch processor when the data set is finally printed/punched.

VSE/POWER does not mix SYSOUT data types in one transmission stream. If there is print and punch output to be sent, then this will be transmitted as two distinct transmissions, each having its own job header and job trailer.

3800 section

At execution time, VSE/POWER creates a 3800 section for all output for which a 3800 device had been specified. The values in the section are taken from the values specified in the * $$ LST statement or from the SETPRT defaults.

In VSE/POWER, there are two sets of defaults for the 3800 -- the hardware defaults and the software defaults. A value of X'00' denotes the use of hardware defaults and a value of X'40' denotes the use of software defaults for VSE/POWER; in JES2 and JES3, a value of X'00' denotes the use of software defaults (hardware defaults are unknown). This means that when VSE/POWER is the end destination, it must determine if the file originated at a VSE/POWER node. When that is not the case, the defaults are converted.

Output Processing Section

This section is built by VSE/POWER (starting with Version 2.3) when a user-defined output parameter is coded on the * $$ LST/PRT statement or when an OPTB structure is passed via the SAS interface at the beginning of output spooling.

VSE/POWER supports OPTBs up to a maximum length of 32,670. No additional code is required to have down-level VSE/POWER systems (prior to Version 2.3) store-and-forward this section.

Multiple Data Set Headers

VSE/POWER does not create multiple destination data sets. The receipt of multiple headers with no SYSOUT data in between indicates to the SYSOUT receiver that the data set has multiple destinations; the receiver builds multiple queue entries accordingly. When the data set is transmitted again, it will be transmitted as multiple copies of headers and data. VSE/POWER always splits jobs when they are received and, therefore, never generates multiple data set headers.

Notification

When a job's system output (or any part of it) reaches its ultimate destination, the SYSOUT receiver at the destination node examines the userid in the job header. If "NTFY=" was specified on the JCL JOB statement, the SYSOUT receiver issues a message to the specified user indicating that the job's output was received.

Job Output Routing

Default Output Routing

Unless otherwise specified on control statements, the default print and punch nodes are set to the originating node. The information is taken from the job header record.

Operator Rerouting

The operator can reroute any SYSOUT that is currently resident in his queue. Both the node and the userid can be changed.

Fanout (Optimized) Support

VSE/POWER does not support fanout. Any input stream containing multiple data set headers is broken down into streams consisting of single data sets using the following criteria:

VSE/POWER keeps all data sets belonging to a transmission together, provided the following important criteria do not change:

- Target node and userid
- Type of output, list or punch
- FCB name
- Priority
- Output class
- Copy count
- Form name
- 3800 characteristics (copy group, burst, copy index).

If any of these characteristics changes within one unit of transmission, then a new spool entry is built together with the job header and data set header. This is necessary because the operator can change the destination of any job or SYSOUT in the spool file and make its destination the local node. This means that all entries on spool are capable of being processed locally.

Stream Support and Control

VSE/POWER supports up to 7 job transmitters, 7 SYSOUT transmitters, 7 job receivers, and 7 SYSOUT receivers per network line/session. The total number of job and SYSOUT streams concurrently active on any line/session cannot exceed 8. The operator has the capability to start and stop each individual stream. A console stream is always defined for every line/session and cannot be stopped by the operator.

Stream Initiation and Suspension

Operator Stream and Line Control

The operator can start and stop any stream on any line/session. If the operator has network authority, the transmitters and receivers on another connected node can also be manipulated. The PACT and PDRAIN commands are used for this purpose.

NJE Tasks

When the line is initialized, all receivers are set in active status, but only job transmitter 1 and SYSOUT transmitter 1 are set in active status. If the VSE/POWER operator wants to activate more transmitters, the operator might do so with the aid of the PACT command. The operator must enter:

- PACT TRn,{JOB|OUT}

  where n is a number from 1-7 and corresponds to transmitter 1-7. The specification of JOB or OUT is necessary to set the correct RCB and FCS bits and inhibit the starting of invalid combinations of JOB and SYSOUT transmitters (for example, JOB transmitter 5 and SYSOUT transmitter 4).

In VSE/POWER, receiver tasks live only for the time span of one transmission (for everything between the job header and job trailer). After the transmission is complete (the end-of-file record has been received and acknowledged), the receiver task detaches. The task is created again when another RIF for the same RCB is received. The detaching of tasks saves resources and can be critical in small systems.

Transmitter tasks live while there is anything in the transmit queue for the node they are serving. As soon as there is nothing eligible in the queue, the task detaches and is created again only when a job or SYSOUT is placed in the transmission queue for this node.

SNA Support

Data Flow Control

RSHUTD is the only data flow control (DFC) function supported by VSE/POWER. If any other function is received, VSE/POWER will terminate the session with a CLSDST or a TERMSESS.

Bind Parameters

VSE/POWER does not negotiate any bind parameters. They are "hard-coded" and most of the parameters received by the other node from VSE/POWER are ignored.

Functional Management Headers

The session will be terminated if the FMH received by VSE/POWER is not acceptable.

Compression

VSE/POWER accepts and handles an FMH3 and decompressed inbound data. VSE/POWER does not compress outbound data.

RU Composition

RU Size Determination

The transmission block size is defined in the network definition table. This value is used in the signon record exchanged with the communicating node. The value that is received from the partner node is compared with the defined value and the smaller of the two values is taken. For SNA, the maximum size allowed by VSE/POWER is 32,000.

RU Multiplexing
> VSE/POWER sends only 1 type of record in an RU and is not able to receive RUs containing mixed record types.

BSC Line and CTC Adapter Support

BSC

Buffer Size
> For BSC, the maximum transmission buffer size allowed by VSE/POWER Version 2.2 is 1800.

Error Recovery

Initialization Error Recovery
> VSE/POWER will retry on Bus Out Check and Equipment Check before terminating.

CTC Adapter

The channel-to-channel adapter is supported by all versions of VSE/POWER Version 2.3 under control of z/VM (RSCS).

TCP/IP

Idle Considerations

VSE/POWER can be an application to the TCP/IP product distributed with z/VSE. When VSE/POWER has no data to send, the TCP/IP product itself decides to send some 'keep-alive' data now and then to avoid that the connection is dropped. Because other TCP/IP products or some network controller do not understand the meaning of 'keep-alive' data (although it is well defined according to the TCP/IP architecture), a connection may nevertheless be dropped in some network configuration. To avoid this dropping, VSE/POWER sends an 'empty' buffer every two minutes. An empty buffer consists of some control bytes indicating a length of zero userdata.

Accounting

For accounting information, see *Networking*, SC33-6573.

Miscellaneous Considerations

Installation Exits

For information about installation exits, see *Networking*, SC33-6573.

Appendix B. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the "Contact us" web page for z/OS (http://www.ibm.com/systems/z/os/zos/webqs.html) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS V2R2 ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM® Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out

251

punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element
The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element
The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the

default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used. However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

* indicates an optional syntax element that is repeatable
The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.

2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.

3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included
The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

This information was developed for products and services offered in the U.S.A. or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. 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 IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY   10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

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

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:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information might contain sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (http://www.ibm.com/software/support/systemsz/lifecycle/)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming Interface Information

This publication documents information that is NOT intended to be used as programming Interfaces of JES2.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at Copyright and Trademark information (http://www.ibm.com/legal/copytrade.shtml).

Index

IBM®

Product Number: 5650-ZOS

Printed in USA