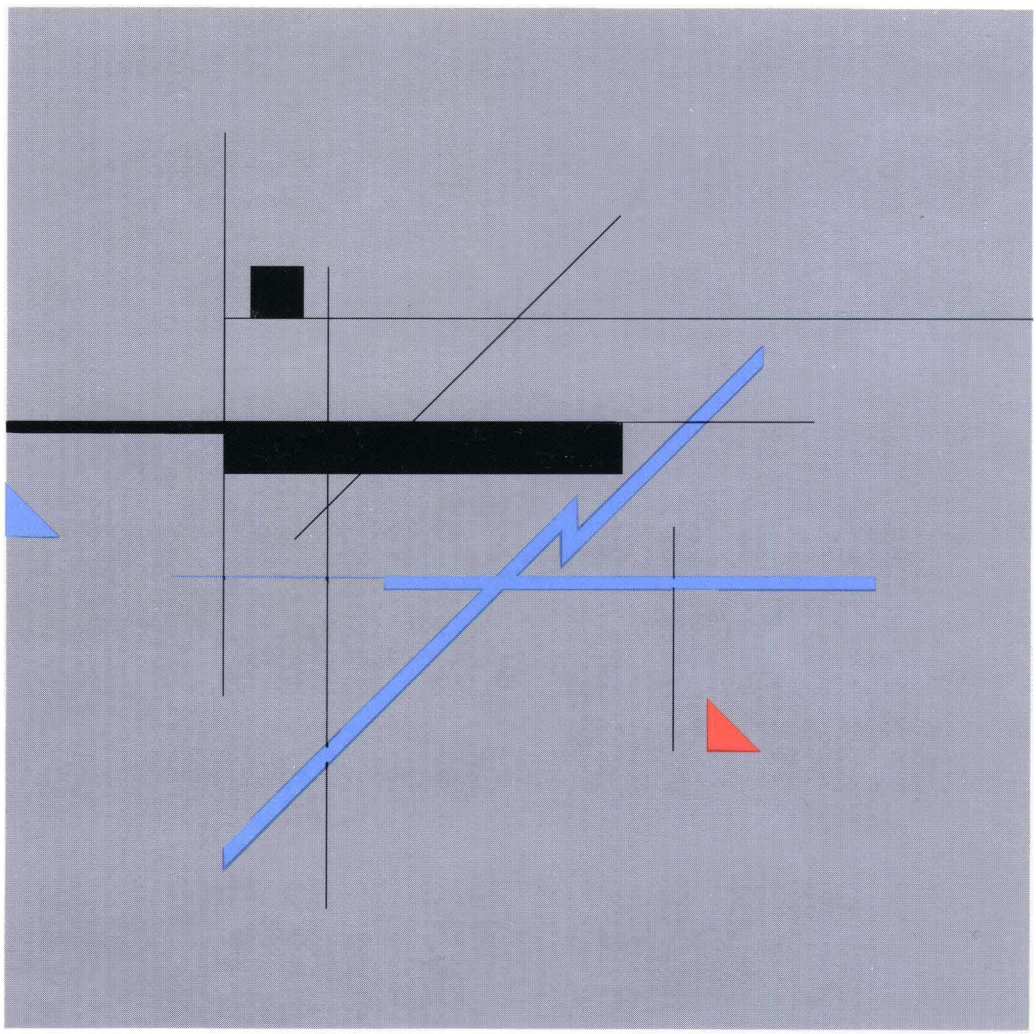
Systems Network Architecture

GA27-3136-12

Formats



Systems Network Architecture

GA27-3136-12

Formats

Note!

See "Notices" on page iii.

Thirteenth Edition (January 1992)

This is a revision of GA27-3136-11, which is now obsolete.

Publications are not stocked at the address given below; the licensed documents are available to customers that have a product license.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Department C40, P. O. Box 12195, Research Triangle Park, North Carolina U.S.A. 27709. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Notices

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates.

Any reference to an IBM licensed program or other IBM product in this publication is not intended to state or imply that only IBM's program or other product may be used.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to use these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, New York 10577.

This book is furnished as is. IBM assumes no responsibility for the use of the functions described in this book in any manner.

This publication may include references to microcode. Some IBM products contain microcode classified as Licensed Internal Code. Licensed Internal Code is provided under terms and conditions set forth in the IBM Customer Agreement.

Preface

ABOUT THE BOOK

This book describes the Systems Network Architecture (SNA) formats used between subarea nodes and peripheral nodes, and between type 2.1 nodes. Readers who need the formats used between subarea nodes should consult *Systems Network Architecture Network Product Formats*, which includes those formats.

HOW THIS BOOK IS ORGANIZED

This book identifies the formats and meanings of the bytes that a basic link unit (BLU) contains. A BLU is the basic unit of transmission at the data link and link station level.

Chapter 1 identifies the formats and meanings of the bytes in a link header and a link trailer.

Chapter 2 identifies the formats and meanings of the information-field bytes in a data link control (DLC) Exchange Identification (XID) command and response.

Chapter 3 identifies the formats and meanings of the bytes in a transmission header.

Chapter 4 identifies the formats and meanings of the bytes in a request or response header.

Chapter 5 identifies the formats and meanings of the bytes in request units and response units.

Chapter 6 explains the transmission services and function management profiles that SNA defines to describe session characteristics.

Chapter 7 identifies the formats and meanings of the bytes in user-structured subfields that appear in a request or response unit.

Chapter 8 identifies the formats and meanings of the control vectors and session keys that appear in a request or response unit.

Chapter 9 identifies the formats and meanings of the management services vectors that appear in a request or response unit.

Chapter 10 explains the meanings of the sense data values defined by System Network Architecture (SNA) that appear, for example, in negative response units.

Chapter 11 presents the descriptions and formats of the different function management headers.

Chapter 12 identifies the formats and meanings of the bytes in a presentation services header.

Chapter 13 identifies the formats and meanings of the general data stream (GDS) variables that are specific to SNA service transaction programs.

Chapter 14 identifies the formats and meanings of the FS1 message units that SNA/Distribution Services transaction programs use.

Chapter 15 identifies the formats and meanings of the FS2 message units that SNA/Distribution Services transaction programs use.

Chapter 16 identifies the general data stream (GDS) variables that are for general use.

Chapter 17 identifies the formats and meanings of the message units that SNA/File Services transaction programs use.

Appendix A provides a summary of SNA character sets and symbol-string types.

Appendix B provides a summary of general data stream identifier (GDS ID) value assignments.

Appendix C lists the common structures for SNA condition reports.

Appendix D lists the abbreviations and symbols that are used in this book.

RELATED PUBLICATIONS

Related publications, providing overview and protocol information, are:

- *Systems Network Architecture Concepts and Products* (GC30-3072)
- *Systems Network Architecture Technical Overview* (GC30-3073)
- *IBM Synchronous Data Link Control Concepts* (GA27-3093)
- *Systems Network Architecture Format and Protocol Reference Manual: Architectural Logic* (SC30-3112)
- *Systems Network Architecture Network Product Formats* (LY43-0081)
- *Systems Network Architecture Type 2.1 Node Reference* (SC30-3422)
- *Systems Network Architecture: Sessions Between Logical Units* (GC20-1868)
- *Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2* (GC30-3084)
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* (SC30-3269)
- *Systems Network Architecture LU 6.2 Reference: Peer Protocols* (SC31-6808)
- *Systems Network Architecture/Distribution Services Reference* (SC30-3098)
- *Systems Network Architecture/File Services Reference* (SC31-6807)
- *Systems Network Architecture/Management Services Reference* (SC30-3346)
- *Token-Ring Network Architecture Reference* (SC30-3374)

- *Document Interchange Architecture: Technical Reference* (SC23-0781)
- *IBM Implementation of X.21 Interface General Information Manual* (GA27-3287)

Summary of Changes

Major additions for GA27-3136-12:

The new information in this edition includes:

- Formats for APPN network nodes – primarily additions to GDS variables and control vectors.
- Formats for length-checked compression – see the discussion in the RH chapter.
- Additional formats for SNA/MS. New formats have been defined for problem management, change management, operations management, and multiple-domain support.
- Additional formats for SNA/FS. New formats have been defined for the SNA/FS Server Request and the global name.

Contents

Figures

Chapter 1. DLC Links

Two data link controls are described in this chapter: "Synchronous Data Link Control (SDLC)," beginning on this page, and the "Token-Ring Network DLC" on page 1-7.

Synchronous Data Link Control (SDLC)

All transmissions on an SDLC link are organized in a specific format called a frame:

```
Frame = BLU = LH [,I-field], LT

where:        BLU  = Basic Link Unit
               LH  = Link Header
          I-field  = Information field
               LT  = Link Trailer
```

Link headers and link trailers contain data link control information for synchronous data link control (SDLC) links. An SDLC frame begins with the link header (LH), which has three fields: the Flag, Address, and Control fields. The link trailer (LT) follows the Information field and is three bytes long. The first two bytes make up the Frame Check Sequence field; the last byte, the closing Flag field. The following pages identify the formats and meanings of the bytes in a link header and a link trailer.

Link Header (Flag)

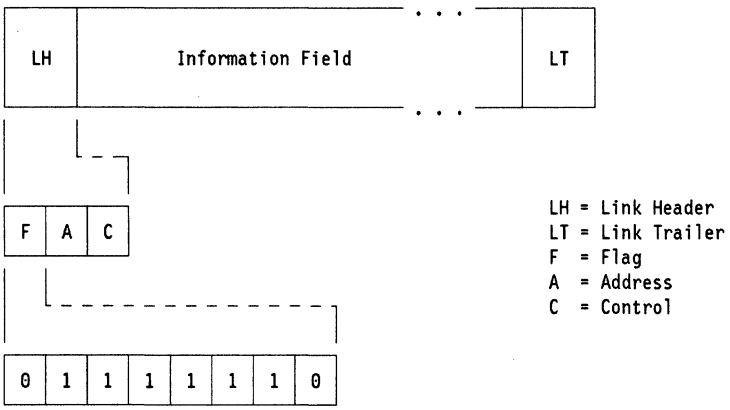


Figure 1-1. Flag Field of Link Header. Always X'7E' (01111110)

All frames begin with a Flag field. The configuration of the flag is always 01111110 (X'7E'). Because frames also *end* with flags (see link trailer), the trailing flag of one frame may serve as the leading flag of the next frame. When receiving, the last 0 in the trailing flag may also be the first 0 in the next leading flag, as Figure 1-2 on page 1-2 illustrates.

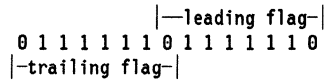


Figure 1-2. Shared Trailing/Leading 0 in SDLC Flags

Note: Zero-bit insertion between the beginning and ending flags prevents a flag pattern from occurring anywhere else in the frame.

Link Header (Address)

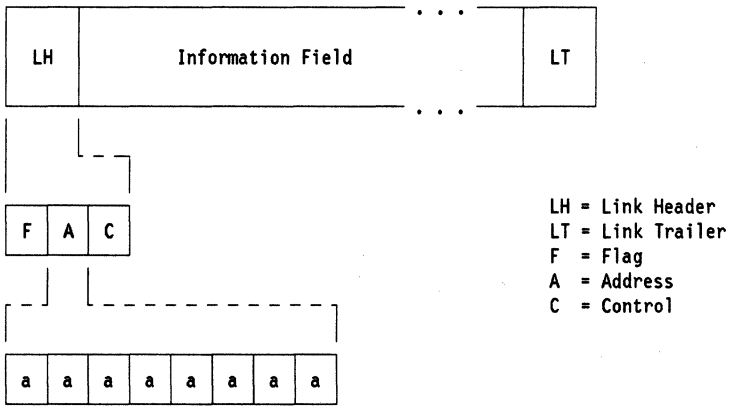


Figure 1-3. Address Field of Link Header. (aaaaaaaa)

The second byte of the link header is the Address field. This address can be:

- a specific link station address — to only one link station
- a group address — to one or more link stations
- a broadcast address X'FF' (or 11111111) — to all link stations
- a "no stations" address X'00'.

The "no stations" address is reserved and should not be used for any link station or group of link stations.

Note: The specific link station address of the secondary is used when the transmission is going from primary to secondary or from secondary to primary.

Link Header (Control)

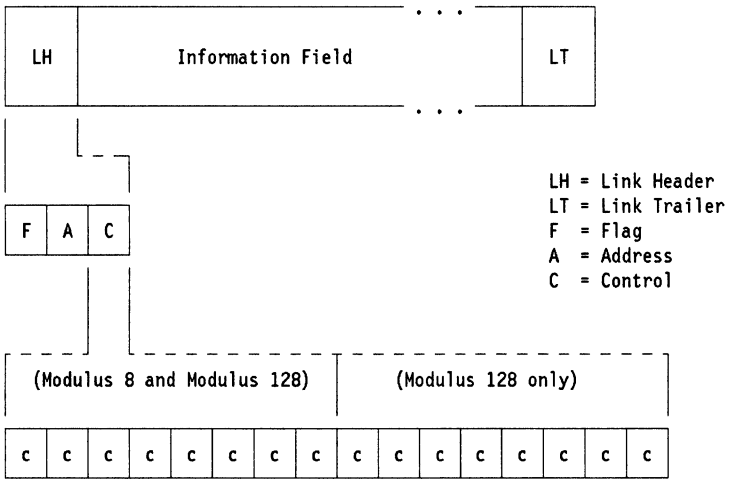


Figure 1-4. Control Field of Link Header. (Eight bits for modulus 8; sixteen bits for modulus 128)

The third byte (or third and fourth bytes) of the link header is the Control field. The Control field contains either an SDLC command or a response. All frames transmitted by a primary station are commands, while frames transmitted by a secondary station are responses. There are three categories of SDLC commands and responses:

- Unnumbered Format
- Supervisory Format
- Information Format

Unnumbered Format: These commands and responses have a poll/final (P/F) bit that is set to 1 to solicit a response (P bit) or when it is the last SDLC frame of a transmission (F bit). This bit is a poll bit for commands and a final bit for responses. Each of the Unnumbered Format commands and responses have two possible hex values: one value for when the poll/final bit is 0 and another value for when the poll/final bit is 1.

Supervisory Format: These commands and responses have a varying number of possible hex values. The number of possible hex values corresponds to the receive sequence numbers assigned to this frame and the setting of the P/F bit. To increase the sequence number modulus from 8 to 128, a two-byte extended Control field is used.

Information Format: These commands and responses also vary in the number of possible hex values. The number of possible hex values correspond to the send and receive sequence numbers assigned to this frame and the setting of the P/F bit. To increase the sequence number modulus from 8 to 128, a two-byte extended Control field is used.

The Information Format is identified by a 0 in the low-order bit of the first or only byte of the Control field. In an Information Format SDLC command or response,

the Information field contains a PIU (path information unit). The remaining chapters of this book, with the exception of Chapter 2, discuss the contents of the PIU.

Figure 1-5 lists the SDLC commands and responses for modulus 8 (one-byte) Control fields; Figure 1-6 lists them for modulus 128 (two-byte) Control fields.

Figure 1-7 describes the Information field of the Frame Reject (FRMR) response frame, which is one of the unnumbered formats listed in Figure 1-5.

FORMAT	BINARY CONFIGURATION	HEX EQUIVALENT P/F off,P/F on	COMMAND NAME	ACRO-NYM
Unnumbered Format	000 P/F 0011	X'03', X'13'	Unnumbered Information	UI
	000 F 0111	X'07', X'17'	Request Initialization Mode	RIM
	000 P 0111	X'07', X'17'	Set Initialization Mode	SIM
	000 F 1111	X'0F', X'1F'	Disconnect Mode	DM
	001 P 0011	X'23', X'33'	Unnumbered Poll	UP
	010 F 0011	X'43', X'53'	Request Disconnect	RD
	010 P 0011	X'43', X'53'	Disconnect	DISC
	011 F 0011	X'63', X'73'	Unnumbered Acknowledgment	UA
	100 P 0011	X'83', X'93'	Set Normal Response Mode	SNRM
	100 F 0111	X'87', X'97'	Frame Reject	FRMR
	101 P/F 1111	X'AF', X'BF'	Exchange Identification	XID
	110 P/F 0111	X'C7', X'D7'	Configure	CFGR
	110 P 1111	X'CF', X'DF'	Set Normal Response Mode Extended	SNRME
	111 P/F 0011	X'E3', X'F3'	Test	TEST
	111 F 1111	X'EF', X'FF'	Beacon	BCN
Supervisory Format	RRR P/F 0001	X'*1', X'*1'	Receive Ready	RR
	RRR P/F 0101	X'*5', X'*5'	Receive Not Ready	RNR
	RRR P/F 1001	X'*9', X'*9'	Reject	REJ
Information Format	RRR P/F SSS0	X'**', X'**'	Numbered Information Present	

Notes: P = Poll bit (sent to secondary station)
F = Final bit (sent to primary station)
RRR = Nr (receive count)
SSS = Ns (send count)
* = Any value

Figure 1-5. Control Fields for SDLC Commands and Responses—Modulus 8

FORMAT	BINARY CONFIGURATION	HEX EQUIVALENT	COMMAND NAME	ACRO-NYM
Unnumbered Format	same as modulus 8 (one-byte), as in Figure 1-5.			
Supervisory Format	0000 0001 RRRR RRR P/F	X'01**'	Receive Ready	RR
	0000 0101 RRRR RRR P/F	X'05**'	Receive Not Ready	RNR
	0000 1001 RRRR RRR P/F	X'09**'	Reject	REJ
Information Format	SSSS SSS0 RRRR RRR P/F	X'****'	Numbered Information Present	

Notes: P = Poll bit (sent to secondary station)
F = Final bit (sent to primary station)
RRR = Nr (receive count)
SSS = Ns (send count)
* = Any value

Figure 1-6. Control Fields for SDLC Commands and Responses—Modulus 128

Link Trailer (Frame Check Sequence)

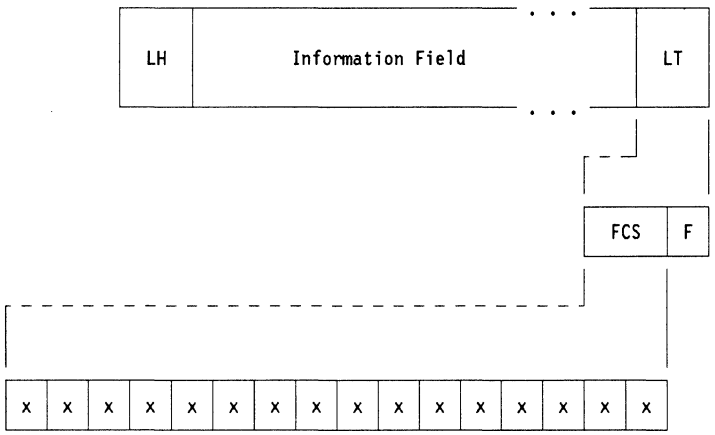
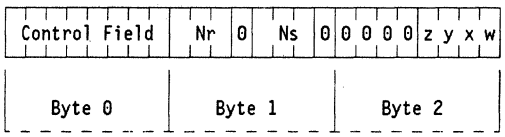


Figure 1-8. Frame Check Sequence Field of Link Trailer

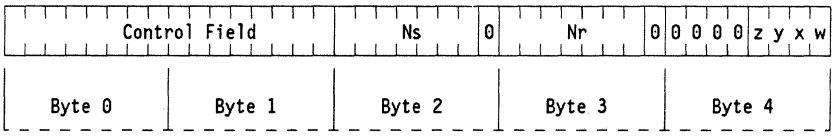
The Frame Check Sequence field carries information that the receiver uses to check the received frame for errors that may have been introduced by the communication channel. This field contains a 16-bit check sequence that is the result of a computation on the contents of both the LH (with the exception of the flag) and the Information field at the transmitter. Cyclic redundancy checking (CRC) is used to perform this calculation. The receiver performs a similar computation and checks its results.

Information Field of the FRMR Response Frame

Modulus 8:



Modulus 128:



Note: For modulus 128, if control field causing FRMR is an unnumbered format (one-byte), it is placed in byte 0 and byte 1 is set to all 0's.

Field	Description	Explanation/Usage
C	Control Field	Control field of the rejected command, as received
Nr	Receive Count	This station's present receiver frame count (the existing count prior to FRMR)
Ns	Send Count	This station's present transmitter frame count (the existing count prior to FRMR)
	Rejection Indicators:	
z	Count	0 = no error 1 = Received Nr disagrees with transmitted Ns
y	Buffer	0 = no error 1 = Buffer overrun (I-field is too long)
x	I-field	0 = no error 1 = Prohibited I-field received
w	Command	0 = no error 1 = Invalid or nonimplemented command received

Figure 1-7. Information Field of the FRMR Response Frame – Modulus 8 and Modulus 28. In each byte, the low-order bit is sent first and the high-order bit is sent last.

Link Trailer (Flag)

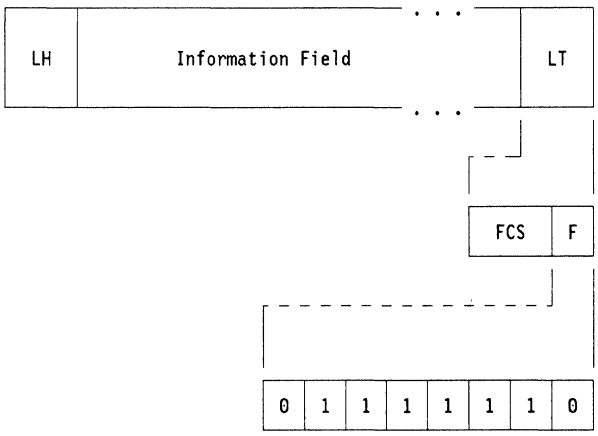


Figure 1-9. Flag Field of Link Trailer. Always X'7E' (01111110)

All frames end with a Flag field. The configuration of the ending (trailing) flag is the same as that of the beginning (leading) flag that is present in the link header: X'7E' (01111110).

Token-Ring Network DLC

The token-ring network DLC consists of two sublayers: the medium access control and the logical link control. The medium access control (MAC) sublayer controls the routing of information between the physical layer and the logical link control sublayer. It provides the following functions: address recognition, frame copying, frame delimiting, and 32-bit frame check sequence generation and verification. The logical link control (LLC) sublayer provides sequential, connection-oriented data transfer.

The following commands and responses, a subset of those shown in Figure 1-6, are used by the LLC sublayer in the token-ring network:

Format	Command/Response Name
Unnumbered Format	DM Response
	DISC Command
	UA Response
	SABME Command
	FRMR Response
	XID Command or Response
	Test Command or Response
Supervisory Format	Receive Ready
	Receive Not Ready
	Reject
Information Format	Numbered Information Present

Figure 1-10. LLC Commands and Responses

The code points associated with these commands and responses are the same as those shown in Figure 1-6.

The token-ring network DLC, in contrast to SDLC, transmits the high-order bit first and the low-order bit last within each byte. Also, zero-bit insertion is required on the token-ring network, since the differential Manchester encoding technique is used.

Additional information about the token-ring network DLC architecture is contained in the *Token-Ring Network Architecture Reference.*

Chapter 2. Exchange Identification (XID) Information Fields

This chapter describes the formats of the information field of the DLC XID command and response.

Throughout this book, *reserved* is used as follows:

- Reserved bits or fields are currently set to 0's (unless explicitly stated otherwise)
- Reserved values are those that currently are invalid

Correct usage of reserved fields is enforced by the sender; no receive checks are made on these fields.

DLC XID Information-Field Formats

DLC XID Information Field

Byte	Bit	Content
0	0−3	Format of XID I-field: X'0' fixed format: only bytes 0−5 are included X'1' variable format (for T1|2.0 to T4|5 node exchanges): bytes 0−p are included X'2' reserved X'3' variable format (for T2.1 to T2.1|4|5 node exchanges): bytes 0−p are included X'8'−X'F' defined for external standards organizations
	4−7	Type of the XID-sending node: X'1' T1 X'2' T2 X'3' reserved X'4' subarea node (T4 or T5)
1		Length, in binary, of variable-format XID I-field (bytes 0−p); reserved for fixed-format XID I-field
2−5|7		Node Identification

DLC XID Information Field

Byte	Bit	Content
2–5	0–11	Block number: an IBM product specific number; see the individual product specifications for the specific values used *Note:* The values all 0's and all 1's indicate that bytes 2–5 do not contain a unique node identifier.
	12–31	ID number: a binary value that, together with the block number, identifies a specific station uniquely within a customer network installation; the ID number can be assigned in various ways, depending on the product; see the individual product specifications for details *Note 1:* When the Block Number field does not contain all 0's or all 1's, a value of all 0's in the ID number indicates that no ID number has been assigned. *Note 2:* For XID format 3, the contents of bytes 2–5 of the Node Identification field are used in some instances as a role-negotiation-value to resolve contention in protocol roles of nodes, e.g., primary/secondary DLC roles or the ODAI value to be appended to the (OAF', DAF') values assigned at a node. When a role-negotiation value is needed and the node does not supply a unique node identification value, it supplies a random value in the ID Number field.
End of Format 0		
6–p		Format 1 Continuation
6–7		Reserved
8		Link Station and Connection Protocol Flags
8	0–1	Reserved
	2	Link-station role of XID sender: 0 sender is a secondary link station (nonnegotiable) 1 sender is a primary link station (nonnegotiable)
	3	Reserved.
	4–7	Link-station transmit-receive capability: X'0' two-way alternating X'1' two-way simultaneous
9		Characteristics of the node of the XID sender:
	0–1	Reserved
	2–3	Segment assembly capability of the path control element of the node: 00 the Mapping field is ignored and PIUs are forwarded unchanged 01 segments are assembled on a link-station basis 10 segments are assembled on a session basis 11 only whole BIUs are allowed
	4–7	Reserved
10–11		Maximum I-field length that the XID sender can receive:
	0	Format flag: 0 bits 1–15 contain the maximum I-field length (only value defined)
	1–15	Maximum I-field length, in binary

DLC XID Information Field

Byte	Bit	Content
12	0–3	Reserved
	4–7	SDLC command/response profile: X'0' SNA link profile (only value defined) *Note:* These profiles refer to the mandatory command/response support on an SDLC link, as follows: • For an SDLC link in normal response mode (NRM/NRME), having a point-to-point or multipoint configuration (determined from system definition), the support required is: **Commands** / **Responses** I-frames / I-frames RR / RR RNR / RNR Test / Test XID / XID SNRM / UA SNRME / UA Disconnect / DM - / RD - / Frame Reject Reject / Reject *Note 1:* The RD response is sent by the secondary station if and only if CS has decided to deactivate the link. *Note 2:* Reject is required only if both sender and receiver have two-way simultaneous transmit-receive capability. • For an SDLC link in normal response mode (NRM), having a loop configuration (determined from system definition), the support required is: **Commands** / **Responses** I-frames / I-frames RR / RR RNR / RNR Test / Test XID / XID SNRM / UA Disconnect / DM UP / - - / Frame Reject Configure / Configure - / Beacon - / RD *Note:* The RD response is sent by the secondary station if and only if CS has decided to deactivate the link.

DLC XID Information Field

Byte	Bit	Content
13	0–1	Reserved
	2	SDLC initialization mode options: 0 SIM and RIM not supported 1 SIM and RIM supported
	3–7	Reserved
14–15		Reserved
16	0	Reserved
	1–7	Maximum number of I-frames that can be received by the XID sender before an acknowledgment is sent, with an implied modulus for the send and receive sequence counts—less than 8 implies a modulus of 8, 8 or greater implies a modulus of 128
17		Reserved
For byte 9, bit 7 = 0 (short-hold mode not supported)		
18–p		SDLC Address Assignment Field
18		Length (p minus 18), in binary, of the SDLC address to be assigned
19–p		Secondary station address to be assigned
For byte 9, bit 7 = 1 (short-hold mode supported)		
18–p		Short-Hold Mode Dependent Parameters
18		Reserved
19–n		Dial Digits of XID Sender
19		Number, in binary, of dial digits
20–n		Dial digits: a string of digits, each having the form X'Fn' ($0 \le n \le 9$)
n+1–p		Dial digits of an available short-hold mode port *Note:* This field is included only in an XID from a T4 or T5 node and only for an incoming call on an already logically busy (byte 9, bit 6 = 1) short-hold mode port. If this field is not included, then p = n.
n+1		Number, in binary, of dial digits of an available short-hold mode port, if one exists
n+2–p		Dial digits of an available short-hold mode port: a string of digits, each having the form X'Fn' ($0 \le n \le 9$) *Note:* Byte n+1 is set to the value X'00' and the n+2-p field is not included if no free alternate port is found. In this case, the station may retry later on the same port used for the current XID.
End of Format 1		
6–p		Format 3 Continuation
6–7		Reserved
8–9		Characteristics of the node of the XID sender:

DLC XID Information Field

Byte	Bit	Content
	0	INIT-SELF support: 0 INIT-SELF may be sent to the XID sender *Note:* If the XID sender does not contain an SSCP, it forwards any INIT-SELF received to the proper node for processing, which returns the response to the originator of the request. 1 INIT-SELF (and character-coded logon) cannot be sent to the XID sender *Note:* For bits 0−1, the value 11 is reserved.
	1	Stand-alone BIND support: 0 BIND may be sent to the XID sender without a prior INITIATE sequence (i.e., XID sender supports independent-PLU session partners) 1 BIND may not be sent to the XID sender (i.e., the XID sender does not support independent-PLU session partners) *Note:* For bits 0−1, the value 11 is reserved.
	2	Whole-BIND-PIUs generated indicator: 0 this node can generate BIND PIU segments 1 this node does not generate BIND PIU segments
	3	Whole-BIND-PIUs required indicator: 0 this node can receive BIND PIU segments 1 this node cannot receive BIND PIU segments *Note:* The value 10 for bits 2−3 is reserved.
	4−7	Reserved
	8	ACTPU suppression indicator: 0 ACTPU for an SSCP-PU session requested 1 ACTPU for an SSCP-PU session not requested
	9	Networking capabilities indicator: 0 the sender is not a network node 1 the sender is a network node
	10	Control point services (reserved when bytes 8−9, bit 11 is 0): 0 CP services not requested or provided 1 CP services requested or provided: when network services are not provided on this TG by the XID sender (bit 9 = 0), CP services are requested; when network services are provided on this TG by the XID sender (bit 9 = 1), CP services are provided.
	11	CP-CP session support: 0 CP-CP sessions not supported on this TG 1 CP-CP sessions supported on this TG *Note:* The following combinations of bits 9, 10, and 11 are not valid: 010, 101, 110.
	12−13	XID exchange state indicators: 00 exchange state indicators not supported (set only by implementations not at the current level of SNA) 01 negotiation-proceeding exchange 10 prenegotiation exchange 11 nonactivation exchange
	14	Nonactivation exchange secondary-initiated capability: 0 nonactivation exchange initiated by secondary station not supported 1 nonactivation exchange initiated by secondary station supported

DLC XID Information Field

Byte	Bit	Content
	15	CP name change support indicator: 0 the sender will fulfill nonactivation XID exchange protocols but, except for the Exchange State indicators, is not able to process fields in the received XID3 that differ from those sent in during the previous XID3 exchange. 1 the sender can process nonactivation XID3s that contain a CP name or TG number that differs from that received by the sending node during the last XID exchange.
10	0–1	BIND pacing support over the TG
	0	Adaptive BIND pacing support as a BIND sender: 0 adaptive BIND pacing as a BIND sender not supported 1 adaptive BIND pacing as a BIND sender supported
	1	Adaptive BIND pacing support as a BIND receiver: 0 adaptive BIND pacing as a BIND receiver not supported 1 adaptive BIND pacing as a BIND receiver supported *Note:* The combinations of bits 0 and 1 have the following meanings: 00 means adaptive BIND pacing is not supported; 01 means one-way adaptive BIND pacing is supported; 11 means adaptive BIND pacing is fully supported.
	2	Quiesce TG request indicator: 0 the sender requests that the receiving node generate a topology update stating that the TG from the receiver to the sender is operative. 1 the sender is requesting that the receiving node generate a topology update stating that the TG from the receiver to the sender is quiesced. *Note:* The requested topology update is generated only if the value sent in this field differs from that sent in the previous XID exchange.
	3	PU capabilities support: 0 does not support receipt of ACTPU containing a PU Capabilities (X'80') control vector 1 supports receipt of ACTPU containing a PU Capabilities (X'80') control vector
	4	APPN border node support: 0 APPN border node function not supported 1 APPN border node function supported
	5	Reserved
	6–7	Qualifier for Adaptive BIND pacing support: 00 Adaptive BIND pacing support applies to BINDs for BOTH independent and dependent LUs, and is nonnegotiable. 01 Adaptive BIND pacing support applies to BINDs for BOTH independent and dependent LUs, unless overridden by the partner node. 10 reserved 11 (Retired) Adaptive BIND pacing support applies to BINDs for only independent LUs. **Note:** Nodes using this setting cannot be connected to those using the 00 setting.
11–14		Reserved

DLC XID Information Field

Byte	Bit	Content
15	0	Parallel TG support indicator: 0 parallel TGs not supported, only a single TG between the sender and the receiver is permitted 1 parallel TGs are supported, more than one TG between the sender and the receiver may be activated
	1–7	Reserved
16		Transmission group number: a binary value in the range 0 to 255
17		DLC type: X'01' SDLC X'02' System/370 channel to controller DLC
18–n		DLC Dependent Section
18		Length, in binary, of the DLC Dependent Section field
For SDLC		
19		Link Station and Connection Protocol flags
19	0	Reserved
	1	ABM support indicator: 0 XID sender cannot be an ABM combined station 1 XID sender can be an ABM combined station
	2–3	Link-station role of XID sender: 00 sender is a secondary link station (nonnegotiable) 01 sender is a primary link station (nonnegotiable) 10 reserved 11 sender is a negotiable link station (primary or secondary capability) *Note:* For ABM stations, the value of bits 2–3 is used only for the purposes of OAF′-DAF′ assignment and deciding which node sends the Set Mode command.
	4–5	Reserved
	6–7	Link-station transmit-receive capability: 00 two-way alternating 01 two-way simultaneous
20	0	ABM nonactivation XID exchange initiator indicator: 0 XID sender is not the initiator of a nonactivation XID exchange on an ABM TG 1 XID sender is the initiator of a nonactivation XID exchange on an ABM TG XID command *Note:* Support for the ABM Nonactivation XID Initiator indicator is required for all ABM link stations that also support secondary-initiated nonactivation XID exchanges.
	1–7	Reserved
21–22		Maximum BTU length that the XID sender can receive:
	0	Format flag: 0 bits 1–15 contain the maximum BTU length (only value defined)
	1–15	Maximum BTU length, in binary
23	0–3	Reserved

DLC XID Information Field

Byte	Bit	Content
	4–7	SDLC command/response profile: X'0' SNA link profile (only value defined) *Note:* These profiles refer to the mandatory command/response support on an SDLC link, as follows: • For an SDLC link in normal response mode (NRM/NRME), having a point-to-point or multipoint configuration (determined from system definition), the support required is:

Commands	Responses
I-frames	I-frames
RR	RR
RNR	RNR
Test	Test
XID	XID
SNRM/SNRME	UA
Disconnect	DM
-	RD
-	Frame Reject
Reject	Reject

Note 1: The RD response is sent by the secondary station if and only if CS has decided to activate the link.
Note 2: Reject is required only if both sender and receiver have two-way simultaneous transmit-receive capability.

• For an SDLC link in normal response mode (NRM), having a loop configuration (determined from system definition), the support required is:

Commands	Responses
I-frames	I-frames
RR	RR
RNR	RNR
Test	Test
XID	XID
SNRM	UA
Disconnect	DM
UP	-
-	Frame Reject
Configure	Configure
-	Beacon
-	RD

Note: The RD response is sent by the secondary station if and only if CS has decided to activate the link.

DLC XID Information Field

Byte	Bit	Content
		• For an SDLC link in asynchronous balanced mode (ABM) (determined from the Link-Station Role of XID Sender field), having a point-to-point configuration, the support required is: **Commands** / **Responses** I-frames / - RR / RR RNR / RNR Reject / Reject SABME / UA Disconnect / DM Test / Test XID / XID - / Frame Reject *Note 1:* All commands and responses are transmitted and received in two-octet format (extended control field). *Note 2:* Frame Reject is not required to be transmitted; receive capability is required.
24	0–1	Reserved
	2	SDLC initialization mode options: 0 SIM and RIM not supported 1 SIM and RIM supported
	3–7	Reserved
25–26		Reserved
27	0	Reserved
	1–7	Maximum number of I-frames that can be received by the XID sender before an acknowledgment is sent (with an implied modulus on NRM connections for the send and receive sequence counts—less than 8 implies a modulus of 8, 8 or greater implies a modulus of 128) The value received in this field is the maximum number of I-frames that the XID receiver may send and have unacknowledged at any given time.
28(=n)		Reserved
For Channel DLC (System/370 Channel to Controller)		*Note:* The System/370 node always contains the primary link station for channel data link control (CDLC); the controller always contains the secondary station.
19–20		Indicators:
	0	Change CDLC parameters; may be set by the primary on a nonactivation XID; echoed by the secondary; reserved for both primary and secondary for other XID exchange types: 0 do not change CDLC parameters 1 change CDLC parameters to the values in this XID; the parameters that may be changed are buffer pre-fetch, number of read commands, buffer size, blocking delay, attention timeout, and time units

DLC XID Information Field

Byte	Bit	Content
	1	Attention timeout support; set by the secondary; reserved for the primary: 0 not supported 1 supported
	2	Reserved
	3	Change CDLC parameters support; specifies whether the XID sender supports changing CDLC parameters by means of a nonactivation XID exchange (see bytes 19−20, bit 0): 0 not supported 1 supported
	4−15	Reserved
21−22		Length, in binary, of the maximum link PIU (LPIU) that the XID sender can receive
23		Buffer pre-fetch: number of buffers suggested for the secondary to preallocate each time the secondary reads LPIUs from the primary
24−25		Number of Read commands: number of Read CCWs the primary must include in every read channel program used to read LPIUs
26−27		Buffer size: for the primary, the size of the input area associated with each Read CCW in channel programs used to read LPIUs; for the secondary, the approximate number of bytes available for LPIU storage in each buffer used for accepting LPIUs from the primary
28−29		Blocking delay: maximum interval that the secondary delays between the time it has an LPIU to send to the primary and the time it presents an Attention to the primary
30−31		Attention timeout: maximum interval that a secondary awaits a read channel program after presenting an Attention to the primary; if the timeout expires, a secondary-detected inoperative station condition is declared. This timeout value is also used for idle detection (1/2 Attention timeout [ATO] is used), second-chance Attention (1/2 ATO is used), and primary-detected inoperative station (3/2 ATO is used)
32−33		Previous number of Read commands: set by the secondary in an XID sent in reply to a change-CDLC-parameters nonactivation XID; otherwise, reserved. The field contains the value of the number-of-Read-commands parameter that was active prior to the change.
34−35		Previous primary buffer size: set by the secondary in an XID sent in reply to a change-CDLC-parameters nonactivation XID; otherwise, reserved. The field contains the value of the primary-buffer-size parameter that was active prior to the change.
36(=n)		Time units: specifies the time units used for Attention timeout and blocking delay X'00' 100-millisecond time units X'01' 1-millisecond time units

DLC XID Information Field

Byte	Bit	Content
n+1−p		Control vectors, as described in "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). X'0E' Network Name control vector: type X'F1', PU name (present only from a T4|5 node XID sender) X'0E' Network Name control vector: type X'F4', network-qualified CP name (always present; the network identifier is always used; i.e., valid lengths of the network-qualified CP name are 3 to 17 bytes with an imbedded period separating the network identifier and CP name parts of the field) X'0E' Network Name control vector: type X'F7', local name of the ALS at the XID sender (present when the sending node provides a nonnegotiated representation of the link in addition to the negotiated TG number) X'10' Product Set ID control vector (always present when the Exchange State indicators are supported) *Note:* When included in XID, the Product Set ID control vector is limited to a maximum of 60 bytes. X'22' XID Negotiation Error control vector (present when an error during XID negotiation is detected; more than one may be present) X'46' TG Descriptor control vector (present when the sending node is activating a TG through a virtual routing node, or when a border node is activating an intersubnetwork TG)

Chapter 3. Transmission Headers (THs)

Introduction

A transmission header (TH) is the leading, or only, field of every PIU. The first half-byte of any TH is the Format Identifier (FID) field. FID2 corresponds to hexadecimal value 2 in the FID field. The FID2 TH is described below.

FID2 Layout

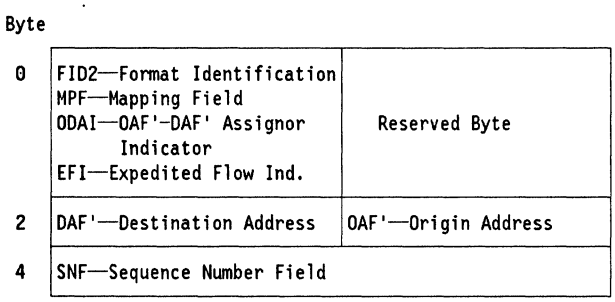


Figure 3-1. Transmission Header for FID Type 2

FID2 Field Descriptions

FID2 is the format used between a T4 or T5 node and an adjacent T2 (i.e., T2.0 or T2.1) node, or between adjacent T2.1 nodes.

FID2 Field Descriptions

Byte	Bit	Content
0	0–3	FID2—Format Identification: 0010
	4–5	MPF—Mapping Field. The MPF consists of bit 4, the Begin-BIU (BBIU) bit, and bit 5, the End-BIU (EBIU) bit. It specifies whether the information field associated with the TH is a complete or partial BIU, and, if a partial BIU, whether it is the first, a middle, or the last segment. 10 first segment of a BIU (BBIU, ¬EBIU) 00 middle segment of a BIU (¬BBIU, ¬EBIU) 01 last segment of a BIU (¬BBIU, EBIU) 11 whole BIU (BBIU, EBIU) *Note:* For all responses (RRI field of the RH is set to 1) and expedited requests (EFI is set to 1), with the exception of BIND and RSP(BIND), the MPF is set to 11; i.e., no segmenting of responses and expedited requests is performed.

FID2 Field Descriptions

Byte	Bit	Content
	6	ODAI—OAF′-DAF′ Assignor Indicator (used for T2.1 - T2.1|BF flows; otherwise, reserved). The ODAI indicates which node assigned (at session-activation time) the OAF′-DAF′ values carried in the TH (see *SNA Type 2.1 Node Reference* for details). Together with the DAF′ and OAF′ values, the ODAI value forms a 17-bit local-form session identifier (LFSID); the DAF′ and OAF′ values used in the TH in one direction are reversed in the other direction. *Note:* See "ISOLATED PACING MESSAGE (IPM)" on page 4-13 for the discussion of the adaptive BIND pacing IPM, which makes exceptional use of these fields.
	7	EFI—Expedited Flow Indicator. The EFI designates whether the PIU belongs to the normal or expedited flow. Normal-flow PIUs are kept in order on a session basis by PC; so are expedited-flow PIUs. Expedited-flow PIUs can pass normal-flow PIUs flowing in the same direction at queuing points in TC within half-sessions and boundary function and APPN session connectors. It has the following meaning: 0 normal flow 1 expedited flow
1		Reserved
2		DAF′—Destination Address Field. See discussion above for ODAI.
3		OAF′—Origin Address Field. See discussion above for ODAI. *Note:* The PU T2.0 is always assigned the local address value of 0. Therefore, BIUs *to* the physical unit always have the associated DAF′ = 0; BIUs *from* the physical unit always have the associated OAF′ = 0. The OAF′ is also 0 for BIUs *from* the SSCP, and DAF′ is 0 for BIUs *to* the SSCP. For T2.1 nodes, an OAF′ or DAF′ can also be set to 0 for independent LU-LU sessions (see *SNA Type 2.1 Node Reference* for details).
4 – 5		SNF—Sequence Number Field. The Sequence Number Field contains a numerical identifier for the associated BIU; path control, when segmenting, puts the same SNF value in each segment derived from the same BIU. The numerical identifier used depends on a number of factors. If the TS profile indicates sequence numbers are not used, the SNF value is a 16-bit identifier that distinguishes a request being sent or responded to from any other outstanding request on the same flow. If the TS profile indicates sequence numbers are used, the flow is a factor. Expedited-flow requests (other than SIG for LU 6.2) carry 16-bit identifiers; expedited-flow responses echo the SNF values of their corresponding requests. Normal-flow requests, other than between LU 6.2s, carry 16-bit numerical values ranging in value from 1 – 65,535 (incremented by 1 for each request) and wrapping through 0 thereafter; the corresponding responses echo their SNF values. The table below defines the SIG and normal-flow SNF usage between LU 6.2s.

	Request	Response
(FMD|LUSTAT) with BB	A	C
(FMD|LUSTAT) with ¬BB	A	B
BIS	A	D
RTR	A	E
SIG	B	E

FID2 Field Descriptions

Byte	Bit	Content
		A: A 16-bit number (1 − 65,535) incremented by 1 for each request and wrapping through 0 thereafter B: Low-order 15 bits of the SNF in the request that carried the last successful BB; the high-order bit identifies the half-session that started the bracket (0 = secondary, 1 = primary); in the case of the first bracket of a session, where the BB is implied, not sent, the low-order 15 bits are 0 and the high-order bit is 1. C: Low-order 15 bits of the SNF in the BB request being responded to; the high-order bit identifies the sender of the BB request (0 = secondary, 1 = primary). D: The half-session does not respond to BIS. E: Same value as the corresponding request. *Note:* For additional details of LU 6.2 processing, see *SNA LU 6.2 Reference: Peer Protocols.*

FID2

Chapter 4. Request/Response Headers (RHs)

Introduction

This chapter identifies the formats and meanings of the request and response headers (RH); "Descriptions of Request Units" on page 5-4 and "Positive Response Units with Extended Formats" on page 5-61 describe the request and response units (RU).

To distinguish between a request and a response, examine bit 0 in byte 0 of the RH:

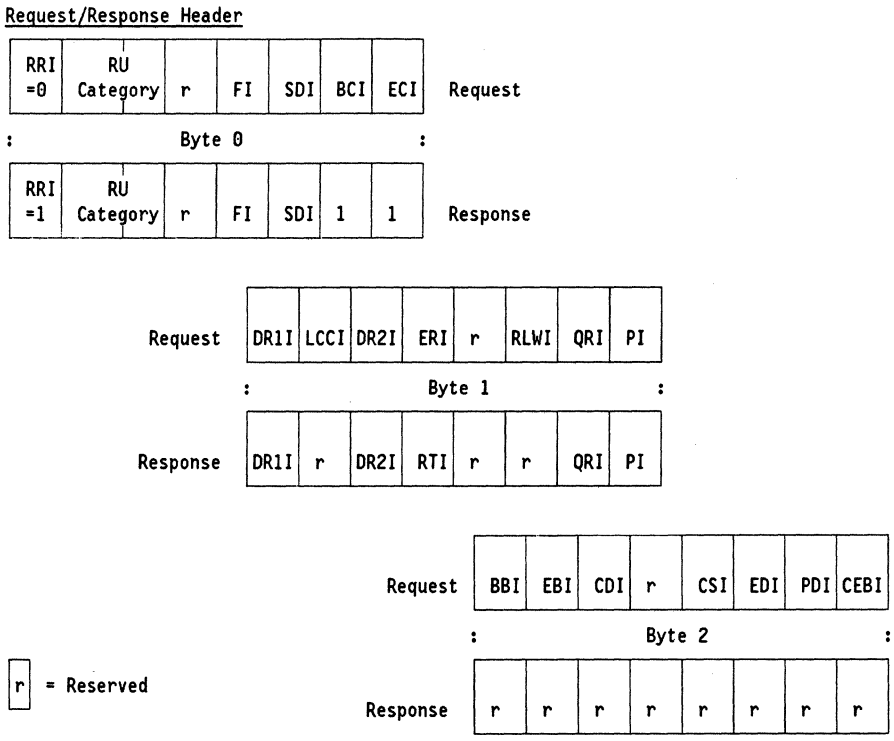
If bit 0 = 0: the RH is a request header and the associated RU is a request unit.

If bit 0 = 1: the RH is a response header and any associated RU is a response unit.

Figure 4-1 on page 4-2 provides a summary of the bytes and field names in the RH.

Length-checked compression (LCC) and the algorithms and additional formats supporting LCC are described in the section "Length-Checked Compression" on page 4-9. The lengths of request units are changed by LCC.

Three message units—IPR, IPM, and EXR—which make use of the RH for special purposes, are described at the end of this chapter.



Field	Description	Explanation/Usage
RRI	Request/Response indicator	0 = request (RQ); 1 = response (RSP)
RU Category	Request/Response Unit Category	00 = FM data (FMD) 01 = network control (NC) 10 = data flow control (DFC) 11 = session control (SC)
FI	Format indicator	0 = no FM header (¬FMH), for LU-LU sessions; or character-coded without an NS header (¬NSH), for network services (NS) 1 = FM header (FMH) follows, for LU-LU sessions; or field-formatted with an NS header (NSH), for NS
SDI	Sense Data Included indicator	0 = not included (¬SD); 1 = included (SD)
BCI	Begin Chain indicator	0 = not first in chain (¬BC); 1 = first in chain (BC)

Figure 4-1 (Part 1 of 2). RH Formats

Field	Description	Explanation/Usage
ECI	End Chain indicator	0 = not last in chain (¬EC); 1 = last in chain (EC)
DR1I	Definite Response 1 indicator	0 = ¬DR1; 1 = DR1
LCCI	Length-Checked Compression indicator	0 = RU is not compressed (¬LCC); 1 = RU is compressed (LCC)
DR2I	Definite Response 2 indicator	0 = ¬DR2; 1 = DR2
ERI	Exception Response indicator	Used in conjunction with DR1I and DR2I to indicate, in a request, the form of response requested. Values and meanings of DR1II, DR2I, ERI are: 000 = no-response requested 100|010|110 = definite-response requested 101|011|111 = exception-response requested
RTI	Response Type indicator	0 = positive (+); 1 = negative (-)
RLWI	Request Larger Window indicator	0 = larger pacing window not requested (¬RLW); 1 = larger pacing window requested (RLW)
QRI	Queued Response indicator	0 = response bypasses TC queues (¬QR); 1 = enqueue response in TC queues (QR)
PI	Pacing indicator	0 = ¬PAC; 1 = PAC
BBI	Begin Bracket indicator	0 = ¬BB; 1 = BB
EBI	End Bracket indicator	0 = ¬EB; 1 = EB (reserved for LU type 6.2)
CDI	Change Direction indicator	0 = do not change direction (¬CD); 1 = change direction (CD)
CSI	Code Selection indicator	0 = code 0; 1 = code 1
EDI	Enciphered Data indicator	0 = RU is not enciphered (¬ED); 1 = RU is enciphered (ED)
PDI	Padded Data indicator	0 = RU is not padded (¬PD); 1 = RU is padded (PD)
CEBI	Conditional End Bracket indicator	0 = not conditional end bracket (¬CEB); 1 = conditional end bracket (CEB) (used for LU type 6.2; else, reserved)

Figure 4-1 (Part 2 of 2). RH Formats

RH Formats

The request/response header (RH) is a 3-byte field; it may be a request header or a response header. The RH control fields shown in Figure 4-1 on page 4-2 are described below.

Request/Response Indicator (RRI): Denotes whether this is a request or a response.

RU Category: Denotes to which of four categories the BIU belongs: session control (SC), network control (NC), data flow control (DFC), or function management data (FMD). (The NC category is not supported by T2.1 nodes.)

Format Indicator: Indicates which of two formats (denoted Format 1 and Format 0) is used within the associated RU (but not including the sense data field, if any; see Sense Data Included indicator, below).

For SC, NC, and DFC RUs, this indicator is always set to Format 1.

On FMD requests for SSCP-SSCP, SSCP-PU, and SSCP-LU sessions, Format 1 indicates that the request RU includes a network services (NS) header and is field-formatted (with various encodings, such as binary data or bit-significant data, in the individual fields). Format 0 indicates that no NS header is contained in the request RU and the RU is character-coded. The Format indicator value on a response is the same as on the corresponding request.

For LU-LU sessions that support FM headers on FMD requests, Format 1 indicates that an FM header begins in the RU (see Chapter 11, "Function Management (FM) Headers"); Format 0 indicates this is not the case. The Format indicator is always set to 0 on positive responses; negative responses are implementation dependent.

For LU-LU sessions that do not support FM headers, the meaning of this indicator on requests, positive responses, and negative responses is implementation dependent. (A BIND session parameter indicates whether FM headers are supported by the session. For further information, see "BIND (BIND SESSION)" on page 5-6 for details on BIND.)

Sense Data Included Indicator (SDI): Indicates that a 4-byte sense data field is included in the associated RU. The sense data field (when present) always immediately follows the RH and has the format and meaning described in Chapter 10, "Sense Data" on page 10-1. Any other data contained in the RU follows the sense data field. Sense data is included on negative responses and on EXRs, where it indicates the type of condition causing the exception.

(The Format indicator does not describe or affect the sense data, which is always in the 4-byte format shown in Chapter 10, "Sense Data" on page 10-1.)

Chaining Control: Indicates that a sequence of contiguous transmitted requests is being grouped in a chain. Two indicators, Begin Chain indicator (BCI) and End Chain indicator (ECI), together denote the relative position of the associated RU

within a chain. The 1 values of these indicators (BCI = 1 and ECI = 1) are referred to as BC and EC, respectively.

```
(BC, ¬EC)  = first RU in chain
(¬BC, ¬EC) = middle RU in chain
(¬BC, EC)  = last RU in chain
(BC, EC)   = only RU in chain
```

Responses are always marked "only RU in chain."

Length-Checked Compression Indicator (LCCI): Indicates that the request unit contains compressed data. Only normal-flow FMD request units can be compressed. When LCCI is set to LCC, the RU consists of a compression header (see "Length-Checked Compression" on page 4-9) followed by compressed data.

For SC, NC, and DFC RUs, this bit is reserved.

Form of Response Requested: In a request header, defines the response protocol to be executed by the request receiver.

Three bits in a request header specify the form of response that is desired. They are: Definite Response 1 indicator (DR1I), Definite Response 2 indicator (DR2I), and the Exception Response indicator (ERI). They can be coded to request:

1. No-response, which means that a response will not be issued by the half-session receiving the request. (DR1I,DR2I) = (0,0) = (¬DR1,¬DR2) and ERI = 0 is the only coding possible; the abbreviation RQN refers to a request with this coding. (Two special responses, ISOLATED PACING RESPONSE [IPR] and ISOLATED PACING MESSAGE [IPM], set [DR1I,DR2I,ERI] = [0,0,0], but they are used independently of the other responses listed. For both IPR and IPM, the sequence number in its associated TH does not correlate it to any given request.)
2. Exception response, which means that a negative response will be issued by the half-session receiving the request only in the event of a detected exception (a positive response will not be issued). (DR1I, DR2I) = (1,0)|(0,1)|(1,1) and ERI = 1 are the possible codings; RQE1, RQE2, and RQE3 are the abbreviations, respectively; the abbreviation RQE or RQE* refers to a request with any of these codings.
3. Definite response, which means that a response will always be issued by the half-session receiving the request, whether the response is positive or negative. (DR1I, DR2I) = (1,0)|(0,1)|(1,1) and ERI = 0 are the possible codings; RQD1, RQD2, and RQD3 are the abbreviations, respectively; the abbreviation RQD or RQD* refers to a request with any of these codings.

A request that asks for an exception response or a definite response has one or both of the DR1I and DR2I bits set to 1 (three combinations); a response to a request returns the same (DR1I, DR2I) bit combination (see Figure 4-2 on page 4-6).

The setting of the DR1I, DR2I, and ERI bits varies by RU category. In the case of LU-LU sessions (e.g., LU 6.2), BIND parameters specify the form of response to be requested during the session; Figure 4-2 on page 4-6 shows the values in tabular form.

For sessions that use sync point protocols with TS profile 4 (LU 6.1), RQD2 or RQE2 asks for the commitment of a unit of work that is to be shared between the session partners; RQD1 is used to request a response when the current unit of work is not to be committed. The table for this set of values is given in Figure 4-3 on page 4-7.

For *nonzero*, non-LU 6.2, LU types that do not use sync point protocols, the specific meanings of the DR1I and DR2I bits are defined in *SNA: Sessions Between Logical Units*; for LU type 0, the interpretations of the DR1I and DR2I bits (and distinctions among the three settings) are implementation-dependent.

The (DR1I, DR2I, ERI) = (0, 0, 1) combination is reserved.

REQUEST	VALID RESPONSE	MEANING OF RESPONSE
RQD1=(1,0,0) (Used by DFC)	+RSP1=(1,0,0) -RSP1=(1,0,1)	positive response negative response
RQE1=(1,0,1) (Used by DFC and PS)	implied +RSP1 -RSP1=(1,0,1)	reply received with no intervening response negative response
RQD2=(0,1,0) RQE2=(0,1,1) (Used by PS)	+RSP2=(0,1,0) -RSP2=(0,1,1) implied +RSP2 -RSP2=(0,1,1)	CONFIRMED verb issued SEND_ERROR verb issued reply received with no intervening response no CONFIRMED verb issued
RQD3=(1,1,0) RQE3=(1,1,1) (Used by PS)	+RSP3=(1,1,0) -RSP3=(1,1,1) implied +RSP3 -RSP3=(0,1,1)	CONFIRMED verb issued SEND_ERROR verb issued reply received with no intervening response no CONFIRMED verb issued

Notes:

1. Values displayed in this table are in the order (DR1I,DR2I,ERI) for requests and (DR1I,DR2I,RTI) for responses.
2. All ¬EC requests are sent as RQE1.
3. RQN = (0,0,0) is not used.

Figure 4-2. FMD Request/Response Combinations for Sessions between Two LU 6.2s

Queued Response Indicator (QRI): In a response header for a normal-flow RU, the Queued Response indicator denotes whether the response is to be enqueued in TC queues (QRI = QR), or whether it is to bypass these queues (QRI = ¬QR). In a

request header for a normal-flow RU, it indicates what the setting of the QRI should be on the response, if any, to this request (i.e., the values on the request and response are the same).

For expedited-flow RUs, this bit is reserved.

The setting of the QRI bit is the same for all RUs in a chain.

Response Type: In a response header, two basic response types can be indicated: positive response or negative response. For negative responses, the RH is always immediately followed by four bytes of sense data in the RU. Thus, RTI = NEG and RTI = POS occur jointly with SDI = SD and SDI = ¬SD, respectively.

REQUEST	VALID RESPONSE	MEANING OF RESPONSE
RQD1=(1,0,0)	+RSP1=(1,0,0) -RSP1=(1,0,1)	positive response negative response
RQE1=(1,0,1)	-RSP1=(1,0,1)	negative response
RQD2=(0,1,0)	+RSP2=(0,1,0) -RSP2=(0,1,1)	positive sync point response negative sync point response
RQE2=(0,1,1)	-RSP2=(0,1,1)	negative sync point response
RQD3=(1,1,0)	+RSP3=(1,1,0) -RSP3=(1,1,1)	positive sync point response negative sync point response
RQE3=(1,1,1)	-RSP3=(1,1,1)	negative sync point response

Notes:

1. Values displayed in this table are in the order (DR1I,DR2I,ERI) for requests and (DR1I,DR2I,RTI) for responses.
2. Each definite- or exception-response chain has the same setting of (DR1I,DR2I)—either (1,0) or (0,1)—on all requests with ECI = ¬EC. When DR1I = 1 on these requests, the End-Chain request can carry (DR1I,DR2I) = (1,0)|(1,1). When DR2I = 1 on these requests, the End-Chain request can carry only (DR1I,DR2I) = (0,1). ERI is 0 only for definite-response chains and when ECI = EC.
3. RQN = (0,0,0) is not used.

Figure 4-3. Request/Response Combinations For TS Profile 4 Sync Points

Three kinds of positive and negative responses correspond to the three valid (DR1I, DR2I) combinations allowed on requests. The settings of the DR1I and DR2I bits in a response always equal the settings of the DR1I and DR2I bits of the form-of-response-requested field of the corresponding request header.

Pacing: In a request header, the Pacing Request indicator denotes that the sender can accept a Pacing Response indicator.

The Pacing Response indicator in a response header is used to indicate to the receiver that additional requests may be sent on the normal flow. In the case of nonadaptive session-level pacing, the Pacing Response indicator may be *on* in an RH that is attached to a response RU on the normal flow; or, if desired, a separate, or isolated, response header may be used, to which no RU is attached. This latter RH signals only the pacing response; it is called an ISOLATED PACING RESPONSE (IPR); isolated and non-isolated pacing responses are functionally equivalent. In the case of adaptive session-level pacing or adaptive BIND pacing, only an ISOLATED PACING MESSAGE (IPM) is used as a pacing response; it is similar to an IPR, but carries additional information. IPR and IPM are discussed further in a later section of this chapter.

Bracket Control: Used to indicate the beginning or end of a group of exchanged requests and responses called a bracket. Bracket protocols are used only on LU-LU sessions. When used, BB appears on the first request in the first chain of a bracket and denotes the beginning of the bracket; the end of the bracket is indicated in one of two ways, depending on LU type.

- For LU 6.2, CEB appears on the last request of the last chain of a bracket. (When bracket usage is specified in BIND, the BIND request carries an implied BB.) The bracket indicators are set only on LUSTAT and FMD requests, and are thus sent normal-flow.
- For other LU types, the end of bracket is delimited by setting EBI to EB in the first request of the last chain in the bracket.

Change Direction Indicator (CDI): Used when there is half-duplex (HDX) control of the normal flows within a session (not to be confused with link-level HDX protocols). It permits a sending half-session to direct the receiving half-session to send. The HDX protocol is useful to half-sessions with limited input/output capabilities that cannot simultaneously send and receive user data. When used, CD appears only on the last request in a chain; it is set only on LUSTAT and FMD requests.

Code Selection Indicator (CSI): Specifies the encoding used for the associated FMD RU. When a session is activated, the half-sessions can choose to allow use of two codes in their FMD RUs (e.g., EBCDIC and ASCII), which they designate as Code 0 and Code 1. FM headers and request and response codes are not affected by the Code Selection indicator.

For SC, NC, and DFC RUs, this bit is reserved.

Enciphered Data Indicator (EDI): Indicates that information in the associated RU is enciphered under session-level cryptography protocols.

For SC, NC, and DFC RUs, this bit is reserved.

Padded Data Indicator (PDI): Indicates that the RU was padded at the end, before encipherment, to the next integral multiple of 8 bytes in length; the last byte of such padding is the count of pad bytes added, the count being a number (1 – 7 inclusive) in unsigned 8-bit binary representation.

For SC, NC, and DFC RUs, this bit is reserved.

Request Larger Window Indicator (RLWI): For a request with PI = PAC, indicates, for adaptive pacing, that the receiver should increase its window size (as specified in the most recently returned IPM) if it is possible to do so; otherwise, the bit is reserved. Typically, the sender sets RLWI to RLW if its residual pacing count is 0 when it receives a solicited IPM and its send pacing queue is not empty, indicating that it could make use of a larger window size; otherwise, it sets RLWI to ¬RLW.

Length-Checked Compression

Two forms of compression are used in SNA: the older form is distinguished by FM headers and is known as *FMH-1 SCB compression*; the newer form, distinguished by the usage of an indicator bit in the RH, is called *length-checked compression (LCC)*. While FMH-1 SCB compression uses only a run-length encoding (RLE) algorithm, LCC can use more powerful algorithms, as well as RLE. FMH-1 SCB compression is not addressed in this section; for details, see the discussion of FM header 1 in Chapter 11, "Function Management (FM) Headers" on page 11-1 and in *SNA Sessions between Logical Units*.

For LCC, the length-checked compression indicator (LCCI) in the RH is set to 1 (LCC). When the LCCI is set to LCC, the first three bytes of the RU form the compression header and the remainder of the RU carries compressed data. The first byte of the compression header gives information about the size of the input symbols for the raw data, the compression algorithm or algorithms used, and the number of bytes (currently always 3) in the compression header. The remaining bytes in the header indicate the raw data length (length of the original RU).

The availability of compression and compression algorithms is implementation-dependent. The use of compression and choice of compression algorithms is negotiated at BIND time. The levels and even usage of compression can be different for the PLU-to-SLU and the SLU-to-PLU traffic.

Only normal-flow FMD RUs are compressed. An RU whose uncompressed length is greater than can be expressed in the compression header will not be compressed. Compression is performed before encryption and decompression is performed after decryption. Sessions that have negotiated compression do not need to compress all RUs; an RU may be compressed with algorithms different from those used for a later RU on that same session.

Currently, two LCC algorithms exist: run-length encoding (RLE) and Lempel-Ziv (LZ). Their usage is specified in the compression header, which has the format shown below.

Compression Header

Byte	Bit	Content
0	0–3	Compression algorithm: 0001 RLE, if uncompressed data type (indicated in bits 4–7) is 0001 0010 LZ, if uncompressed data type is 0001 0011 LZ-compression after RLE, if uncompressed data type is 0001

Compression Header

Byte	Bit	Content
	4–7	Uncompressed data type and compression header size: 0001 8-bit text data; compression header size is 3 bytes
1–2		Length, in binary, of uncompressed RU

After decompression, the decompressed RU's length is compared with the length given in the compression header. If a mismatch exists, the session is terminated with an UNBIND, accompanied by the appropriate sense data.

Run-Length Encoding

Run-length encoding (RLE) eliminates strings of repeated bytes. With the RLE algorithm, the first byte after the compression header is a control byte, known as an SCB. The SCB has the format shown below.

SCB Format

Bit	Content
0–1	SCB type: 00 Raw data: the following bytes are uncompressed. The Count field (bits 2–7) indicates the number of uncompressed bytes. If the RU is not exhausted, another SCB follows n + 1 bytes after this SCB. 01 Reserved 10 Master-character: the Count field indicates the number of space (X'40') characters compressed. If the RU is not exhausted, another SCB follows this master-character SCB. 11 Duplicate-character: the character (called the *duplicated character*, or DC) that follows this SCB appears in the raw data in an n-byte run; the n-byte run is compressed to this SCB-DC pair. If the RU is not exhausted, another SCB follows this SCB-DC pair.
2–7	Count: indicates the number (n), in binary, of uncompressed bytes that follow (in the case of SCB type 00) or that should be generated upon decompression of this SCB sequence.

SCBs cannot span RUs. In short:

- If the last SCB in an RU is a raw-data SCB, then all of the raw data to which it refers must be in that RU.
- The master-character SCB is allowed to be the final byte in an RU.
- If the duplicate-character SCB is the last SCB in the RU, then that SCB is always the next-to-last byte in the RU, the last byte being the DC.

Lempel-Ziv Compression

The Lempel-Ziv (LZ) algorithm is a dynamic compression algorithm that compresses previously seen strings (in the current or preceding RUs) to 9-, 10-, or 12-bit codes. The choice of code lengths is negotiated at BIND time. Each code, with the exception of 256, represents a zero-origin index of an entry in the compression/decompression table. (The value 0 represents the first entry; the value 1 represents the second, and so forth.) The table entries store previously seen strings. The table entry 256 is not used; the code 256 is used to indicate an LZ control sequence (see "LZ Control Sequence").

Generally, LZ compresses better than RLE, but at a higher cost in terms of storage and processor cycles. Both the LZ compressor and the LZ decompressor have a table in which strings are stored. The compressor and decompressor tables are synchronized and are updated as new strings are seen. Least recently used strings are deleted from the tables when table capacity is reached in order to add new strings.

12-bit LZ is called *large-table LZ*; 10-bit LZ is called *medium-table LZ*; and 9-bit LZ is called *small-table LZ*. In general, the longer the bit code, the better compressed the data. While each of the three requires about the same amount of processing, large-table compresses better than medium-table, which in turn compresses better than small-table. Large-table requires more storage than medium-table, while small-table makes the least demand on storage. LZ compression can be done alone or after RLE.

The majority of the processor support needed for LZ is for updates to the compression tables. In certain situations, it can be advantageous to "freeze" the compression tables and allow only lookups into the tables. If hardware table-lookup is available, compression can be done very quickly. Even without hardware support, a frozen compression table can speed up the compression step. While the tables are frozen, compression ratios can remain favorable. After a while, it may be necessary to unfreeze the tables and allow updates to the tables again. The choice of when to freeze or unfreeze is implementation-dependent. The LZ-compressor signals the freeze or unfreeze condition to the LZ-decompressor by sending an *LZ control sequence*, explained below.

The LZ-compressor may also reset its tables to their initial condition. In this event, the LZ-decompressor must reset its tables also. The compressor signals this condition by sending an LZ control sequence.

Note: The tables are reset whenever a CLEAR or DEALLOCATE(ABEND) is sent. In either event, the tables are immediately set to their initial condition (i.e., their condition at session-activation time); no Reset LZ control sequence is necessary.

LZ Control Sequence

The LZ control sequence is a two-byte format that may appear only directly after a compression header that also indicates the RU is LZ-compressed. The control sequence consists of the LZ control code and the LZ command. The control code is the 9-, 10-, or 12-bit encoding (depending on table size) of 256. (The code 256 is reserved for this purpose; the compression tables do not use this code.) The format of the 2-byte control sequence is shown below.

LZ Control Sequence Format

Bit	Content
0 – i(= 9|10|12)	A right-justified binary 256 control code indicating the following field carries an LZ command.
i + 1 – 15	A right-justified hex value representing an LZ command: X'1' Reset the table to its initial state. X'2' Freeze the table in its current state; do not update it for new strings. X'3' Unfreeeze the table; update it for new strings.

BIND Negotiation for Compression

The BIND negotiation process selects for each direction of communication in the session, the nonusage or usage of compression, and, if compression is to be used, the level of compression in each direction.

Two ways exist to negotiate for compression:

- Using the extended BIND and RSP(BIND), the Length-Checked Compression (X'66') control vector carries the compression capabilities of the nodes along the session path.
- For LU types other than 4 and 7, the nonextended BIND uses two bits (byte 25, bits 6 and 7) to negotiate compression.

Negotiation using the latter method is restricted to small-table LZ or RLE for PLU-to-SLU traffic, and RLE for SLU-to-PLU traffic.

IPR, IPM, and EXR

Three special message units exist in SNA: ISOLATED PACING RESPONSE (IPR), ISOLATED PACING MESSAGE (IPM), and EXCEPTION REQUEST (EXR). These are explained below.

ISOLATED PACING RESPONSE (IPR)

An IPR is used on a session if BIND specifies nonadaptive session-level pacing is used; it indicates a pacing response, and can be used even when operating under no-response protocols.

The following fields of the TH and RH are set for an IPR:

TH: Either the normal or expedited flow may be indicated. The sequence number is undefined (it may be set to any value, and is not checked by the receiver).

RH: An IPR is coded all 0's except for the Request/Response indicator, the Pacing indicator, and the Chain indicators, which are set to 1's; thus, the IPR RH is coded X'830100' by the sender; the receiver identifies an IPR by detecting that (RRI, DR1I, DR2I, PI) = (1, 0, 0, 1) and ignoring the remaining bits.

ISOLATED PACING MESSAGE (IPM)

An IPM is used on a session if BIND and RSP(BIND) specify adaptive session-level pacing is used. Three types of IPM exist: *solicited* IPMs, *unsolicited* IPMs, and *reset acknowledgment* IPMs.

A receiver of paced requests sends a solicited IPM to a sender of paced requests to grant the sender permission to send a group (or *window*) of paced requests; the solicited IPM explicitly specifies the number of requests in the window as the *next-window size*. A receiver of paced requests sends a solicited IPM either (1) after receiving a pacing request, or (2) after sending an unsolicited IPM with a next-window size of 0 and receiving a reset acknowledgment IPM.

A receiver of paced requests sends an unsolicited IPM to a sender of paced requests to withdraw from the sender previously granted permission to send paced requests, typically because of congestion detected by the receiver of paced requests. Upon receiving an unsolicited IPM, a sender of paced requests (1) resets previously granted windows so that any queued requests are sent as part of a subsequent window, and (2) sends a reset acknowledgment IPM to the receiver of paced requests to delimit the end of the current truncated window. The unsolicited IPM also specifies a next-window size that grants a new window; the next-window size may be any value, including 0 (no new window). After sending an unsolicited IPM, a receiver of paced requests ignores any Pacing Request indicator it receives until it receives a reset acknowledgment IPM.

Besides its use for session-level pacing, an IPM is also used on a link basis between a T2.1 node and an adjacent boundary node or T2.1 node for adaptive BIND pacing if the XID3 exchange on the link so allows. This use of IPM is the same as for adaptive session-level pacing, except the pacing window applies only to BINDs flowing over the link.

The following fields are set for an IPM.

TH: Expedited flow is indicated except for a reset acknowledgment IPM, which is always sent normal-flow (because it delimits the current window). The sequence number is undefined (may be set to any value, and is not checked by the receiver). For an adaptive BIND pacing IPM, ODAI is always set to 0, and OAF′ and DAF′ are set according to the sender's normal setting of ODAI in BIND: a node that sets ODAI to 0 in BIND sets OAF′ to X'01' and DAF′ to X'00' in the BIND pacing IPM, while a node that sets ODAI to 1 in BIND sets OAF′ to X'00' and DAF′ to X'01' in the BIND pacing IPM.

IPM

The IPM consists of the RH and a 3-byte extension shown below.

IPM (ISOLATED PACING MESSAGE)

Byte	Bit	Content
0–2		RH: X'830100' (same as for an IPR, with the same receiver-checking mentioned above)
3–5		IPM Extension
3	0–1	Type: 00 solicited: sent in response to a pacing request, or after receiving a reset acknowledgment IPM acknowledging an unsolicited IPM that carried a *zero* next-window size (so paced requests can resume flowing) 01 unsolicited: can be sent at any time, except when a previous unsolicited IPM is still outstanding (no reset acknowledgment yet received) 10 reset acknowledgment: sent to acknowledge receipt of an unsolicited IPM 11 reserved
	2	Reset current-window residual-count indicator: 0 do not reset the residual count 1 reset the residual count to 0 (i.e., terminate the current window) *Note:* Currently, this bit is set to 1 in an unsolicited IPM, and 0 otherwise.
	3–7	Reserved
4–5		Next-window information:
	0	Format: 0 (only value defined)
	1–15	Next-window size: a binary value in the range 1–32,767 in solicited IPMs, and 0–32,767 in unsolicited IPMs; echoed from unsolicited IPMs in reset acknowledgment IPMs (the echoed value is not checked when received)

EXCEPTION REQUEST (EXR)

Two EXR types are defined: those replacing requests, and those replacing too-long path information units (PIUs) received by transmission group control (TGC) from an upper layer (e.g., ERC in an intermediate routing node).

EXRs replacing requests are generated by some component between the origin and intended destination of a request found to be in error. The following fields are set in the TH, RH, and RU.

TH: The sequence number remains the same as in the request being replaced. The data count is altered to properly record the new BIU size. The Mapping field is set to (BBIU, EBIU); an EXR replaces a complete BIU, not just one segment of a segmented BIU. All other fields are left as received.

RH: The Sense Data Included bit is set to 1. All other fields are unchanged.

RU: Bytes 0–3 contain sense data defining the last error detected, and in the same format as returned in negative responses. The sense data is followed by the original RU, truncated to no more than three bytes, as described for negative responses.

EXRs replacing too-long PIUs are formatted as follows.

TH: Like EXRs replacing requests, EXRs replacing too-long PIUs change only the Mapping field (to 1's) and the data count (to 10 in this case).

RH: If the PIU is a request, the SDI field is set to indicate sense data is included; the remainder of the RH is unchanged. If the PIU is a middle or last segment of a multi-segment BIU, an RH is supplied and set to X'07B000'.

RU: Bytes 0 – 3 always contain the sense data, X'800A0000'. If the PIU contained a request, bytes 4 – 6 contain up to the first three bytes of the original RU.

Note: A too-long PIU may be found to be a response. In the case of a positive response, the first three bytes are retained and a sense data value of X'800A0000' is inserted ahead of them; the RH is changed to indicate SD and negative response. In the case of a negative response, the existing sense data value is changed to X'800A0000' and the following three bytes of the RU are retained; the RH is unchanged. In both cases, the TH is set to indicate BBIU, EBIU, and DCF = 10.

Chapter 5. Request/Response Units (RUs)

Introduction to Request Units

This section contains detailed formats of the request units, arranged in alphabetical order. Each format description begins with the following heading:

"ABBREVIATED RU NAME (RU NAME)
Origin-NAU → Destination-NAU, Normal (Norm) or Expedited (Exp) Flow; RU Category"

Notes:

1. "RU Category" is abbreviated as follows:

   DFC data flow control

   SC session control

   NC network control

   FMD NS(ma) function management data, network services, management services (Note: formerly maintenence services)

   FMD NS(s) function management data, network services, session services

2. All values for field-formatted requests that are not defined in this section are reserved. (The formats of character-coded FMD NS requests are implementation dependent.)

3. The request-code value X'FF' and the NS-header values X'(3|7|B|F)F****' and X'**(3|7|B|F)F**' are set aside for implementation internal use, and will not be otherwise defined in SNA.

4. Throughout the format descriptions, *reserved* is used as follows:

   - Reserved bits, or fields, are ones that currently are set to 0's (unless explicitly stated otherwise).
   - Reserved values are those that currently are invalid.

   Correct usage of reserved fields is enforced by the sender; no receive checks are made on these fields.

5. Throughout the format descriptions, *retired* fields and values are those that were once defined in SNA but are no longer defined. To accommodate implementations of back-level SNA, current implementations of SNA treat retired fields as follows: send checks enforce the setting of retired fields to all 0's except where other unique values are required (described individually); no receive checks are made on these fields, thereby accepting back-level settings of these fields. Special handling of retired fields, such as echoing or passing on retired fields as received, is discussed where appropriate.

6. User data, control vectors, and session keys referred to in the format descriptions are described in Chapter 7, "User Data Structured Subfields" and Chapter 8, "Common Fields."
7. The character sets referred to in the descriptions of names and other symbol strings in this chapter are defined in Appendix A, "SNA Character Sets and Symbol-String Types."
8. A type 2.1 (T2.1) node contains a control point (CP) rather than a physical unit (PU). However, it can support SSCP-PU T2.0 flows, in which case the designations "SSCP ←→ PU T2" or "SSCP ←→ PU" in the RU descriptions should be assumed to apply to the T2.1 node as well.

Request Unit Summary Information

The following is a categorized list of RU abbreviations, followed by a list of RUs indexed by NS headers and request codes.

Summary of Request RUs by Category

Request RUs prefixed by an asterisk (*) require response RUs that, if positive, have an extended format containing data in addition to the NS header or request code. The RUs prefixed by a plus sign (+) are retired from SNA; see product documentation for information on support.

SC Requests

*ACTLU	CRV	SDT
*ACTPU	DACTLU	*STSN
*BIND	DACTPU	UNBIND
CLEAR	RQR	

DFC Requests

BID	QC	SBI
BIS	QEC	SHUTC
CANCEL	RELQ	SHUTD
CHASE	RSHUTD	SIG
LUSTAT	RTR	

FMD NS(ma) Requests

NMVT	+RECFMS	+REQMS

FMD NS(s) Requests

INIT-SELF	NSPE	TERM-SELF
NOTIFY		

Index of RUs by NS Headers and Request Codes

Within DFC, NC, SC, or any specific FMD NS category, the request code is unique. However, while a request code has only one meaning in a specific category, a given code (e.g., X'05') can represent different requests in separate categories (e.g., DFC, NC, and configuration services).

FMD NS Headers (third byte is the request code)

X'01021B' REQDISCONT
X'010601' NSPE
X'010681' INIT-SELF (Format 0)
X'010683' TERM-SELF (Format 0)
X'410304' REQMS
X'410305' TESTMODE
X'410307' ROUTE-TEST
X'410384' RECFMS
X'410385' RECTR
X'410386' ER-TESTED
X'41038D' NMVT
X'810620' NOTIFY
X'810681' INIT-SELF (Format 1)
X'810683' TERM-SELF (Format 1)

DFC, NC, and SC Request Codes

X'04' LUSTAT (DFC)
X'05' RTR (DFC)
X'0D' ACTLU (SC)
X'0E' DACTLU (SC)
X'11' ACTPU (SC)
X'12' DACTPU (SC)
X'31' BIND (SC)
X'32' UNBIND (SC)
X'70' BIS (DFC)
X'71' SBI (DFC)
X'80' QEC (DFC)
X'81' QC (DFC)
X'82' RELQ (DFC)
X'83' CANCEL (DFC)
X'84' CHASE (DFC)
X'A0' SDT (SC)
X'A1' CLEAR (SC)
X'A2' STSN (SC)
X'A3' RQR (SC)
X'C0' SHUTD (DFC)
X'C0' CRV (SC)
X'C1' SHUTC (DFC)
X'C2' RSHUTD (DFC)
X'C8' BID (DFC)
X'C9' SIG (DFC)

Descriptions of Request Units

ACTLU (ACTIVATE LOGICAL UNIT)

SSCP → LU, Exp; SC

ACTLU is sent from an SSCP to an LU to activate a session between the SSCP and the LU and to establish common session parameters.

ACTLU (ACTIVATE LOGICAL UNIT)

Byte	Bit	Content
0		X'0D' request code
1		Indicators:
	0–5	Reserved
	6–7	Type activation requested: 01 cold (retired) 10 ERP
2	0–3	FM profile: X'0' FM profile 0
	4–7	TS profile: X'1' TS profile 1 (only value defined)

ACTPU (ACTIVATE PHYSICAL UNIT)

SSCP → PU, Exp; SC

ACTPU is sent by the SSCP to activate a session with the PU, and to obtain certain information about the PU.

ACTPU (ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
0		X'11' request code
1	0–3	Format: X'0' Format 0 X'1' Format 1 – same as Format 0, except that it always includes one or more control vectors in bytes 9–n (sent only to type 2 nodes that use XID3 with byte 10, bit 3 set = 1)
	4–7	Type activation requested: X'1' cold (retired) X'2' ERP

ACTPU (ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
2	0–3	FM profile: X'0' FM profile 0
	4–7	TS profile: X'1' TS profile 1
3–8		A 6-byte field that specifies the ID of the SSCP issuing ACTPU; the first four bits specify the format for the remaining bits:
	0–3	Format: 0000 (only value defined)
	4–7	type of the node containing the SSCP
	8–47	Implementation and installation-dependent binary identification
9–n		Control vectors as described following this RU or in the section "Control Vectors" on page 8-3 *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). X'80' PU Capabilities control vector (present for Format 1 only)

PU Capabilities (X'80') ACTPU Control Vector

PU Capabilities (X'80') ACTPU Control Vector

Byte	Bit	Content
0–1		Vector header; Key= X'80'
2		Vector Data
	0	Unsolicited NMVT support: 0 Sending node does not support unsolicited NMVTs for PSID. 1 Sending node supports unsolicited NMVTs for PSID.
	1–7	Reserved

BID (BID)

LU → LU, Norm; DFC

BID is used by the bidder to request permission to initiate a bracket, and is used only when using brackets. This RU is not used for LU 6.2.

BID (BID)

Byte	Bit	Content
0		X'C8' request code

BIND (BIND SESSION)
PLU → SLU, Exp; SC

BIND is sent from a primary LU to a secondary LU to activate a session between the LUs. The secondary LU uses the BIND parameters to help determine whether it will respond positively or negatively to BIND.

BIND (BIND SESSION)

Byte	Bit	Content
0		X'31' request code
1	0–3	Format: 0000 (only value defined)
	4–7	Type: 0000 negotiable (only value defined for LU 6.2) 0001 nonnegotiable
2		FM profile: X'02' FM profile 2 X'03' FM profile 3 X'04' FM profile 4 X'07' FM profile 7 X'12' FM profile 18 X'13' FM profile 19 (only value defined for LU 6.2)
3		TS profile: X'02' TS profile 2 X'03' TS profile 3 X'04' TS profile 4 X'07' TS profile 7 (only value defined for LU 6.2) FM Usage—Primary LU Protocols for FM Data
4	0	Chaining use selection: 0 only single-RU chains allowed from primary LU half-session 1 multiple-RU chains allowed from primary LU half-session (only value defined for LU 6.2)
	1	Request control mode selection: 0 immediate request mode (only value defined for LU 6.2) 1 delayed request mode

BIND (BIND SESSION)

Byte	Bit	Content
	2–3	Chain response protocol used by primary LU half-session for FMD requests; chains from primary will ask for: 00 no response 01 exception response 10 definite response 11 definite or exception response (only value defined for LU 6.2)
	4	2-phase commit for sync point (reserved if any TS profile other than 4): 0 2-phase commit not supported 1 2-phase commit supported
	5	Reserved
	6	FMH-1 SCB compression indicator (reserved for LU 6.2): 0 FMH-1 SCB compression will not be used on requests from primary 1 FMH-1 SCB compression may be used
	7	Send End Bracket indicator: 0 primary will not send EB (only value defined for LU 6.2) 1 primary may send EB
		FM Usage—Secondary LU Protocols for FM Data
5	0	Chaining use selection: 0 only single-RU chains allowed from secondary LU half-session 1 multiple-RU chains allowed from secondary LU half-session (only value defined for LU 6.2)
	1	Request control mode selection: 0 immediate request mode (only value defined for LU 6.2) 1 delayed request mode
	2–3	Chain response protocol used by secondary LU half-session for FMD requests; chains from secondary will ask for: 00 no response 01 exception response 10 definite response 11 definite or exception response (only value defined for LU 6.2)
	4	2-phase commit for sync point (reserved if any TS profile other than 4): 0 2-phase commit not supported 1 2-phase commit supported
	5	Reserved
	6	FMH-1 SCB compression indicator (reserved for LU 6.2): 0 FMH-1 SCB compression will not be used on requests from secondary 1 FMH-1 SCB compression may be used
	7	Send End Bracket indicator: 0 secondary will not send EB (only value defined for LU 6.2) 1 secondary may send EB
		FM Usage—Common LU Protocols

BIND (BIND SESSION)

Byte	Bit	Content
6	0	Whole-BIUs required indicator (reserved in nonextended, non-LU 6.2 BINDs, i.e., when control vector X'60' is not present): 0 the sending node supports receipt of segments on this session 1 the sending node does not support receipt of segments on this session; the maximum send-RU size specified in bytes 10 and 11 of BIND and RSP(BIND) are negotiated so that BIUs on this session are not segmented when sent to a node requiring whole BIUs
	1	FM header usage: 0 FM headers not allowed 1 FM headers allowed (only value defined for LU 6.2)
	2	Brackets usage and reset state: 0 The value of this bit should be 0 if either condition (1) or condition (2) is true. 1. Brackets are not used if neither primary nor secondary will send EB (byte 4, bit 7 = 0 and byte 5, bit 7 = 0). 2. Brackets are used and the bracket state managers' reset states are INB if: • either primary or secondary, or both, may send EB (byte 4, bit 7 = 1 or byte 5, bit 7 = 1). • FM profile 19 is specified (byte 2 = X'13'). (only value defined for LU 6.2) 1 brackets are used and bracket state managers' reset states are BETB
	3	Bracket termination rule selection; byte 4, bit 7 = 0, and byte 5, bit 7 = 0; and if FM profile is not 19): 0 Rule 2 (unconditional termination) will be used during this session 1 Rule 1 (conditional termination) will be used during this session (only value defined for LU 6.2) *Note:* This bit is reserved if both of the following conditions are true. 1. Brackets are not used (byte 4, bit 7 = 0, byte 5, bit 7 = 0 , and byte 6, bit 2 = 0). 2. The FM profile is not 19 (byte 2 ≠ X'13').
	4	Alternate code set allowed indicator: 0 alternate code set will not be used 1 alternate code set may be used
	5	Sequence number availability for sync point resynchronization (reserved if any TS profile other than 4 is used): 0 sequence numbers not available 1 sequence numbers available *Note:* Sequence numbers are transaction processing program sequence numbers from the previous activation of the session with the same session name; they are associated with the last acknowledged requests and any pending requests to commit a unit of work. If no previous activation existed, the numbers are 0, and this bit is set to 0.
	6	BIS sent (reserved for TS profiles other than 4): 0 BIS not sent 1 BIS sent

BIND (BIND SESSION)

Byte	Bit	Content
	7	BIND queuing indicator: 0 BIND cannot be queued (held, pending resource availability, thus delaying the BIND response) 1 BIND sender allows the BIND receiver to queue the BIND for an indefinite period, thus delaying the sending of the BIND response *Note:* BIND sender may provide a timer or operator interface to send UNBIND if session-activation time exceeds BIND sender's implementation-defined limits. BIND queuing is terminated by sending UNBIND to the BIND receiver.
7	0−1	Normal-flow send/receive mode selection: 00 full-duplex 01 half-duplex contention 10 half-duplex flip-flop (only value defined for LU 6.2) 11 reserved
	2	Recovery responsibility: (reserved if normal-flow send/receive mode is FDX, i.e., if byte 7, bits 0−1 = 00): 0 contention loser responsible for recovery (see byte 7, bit 3 for specification of which half-session is the contention loser) 1 symmetric responsibility for recovery (only value defined for LU 6.2)
	3	Contention winner/loser: 0 secondary is contention winner and primary is contention loser 1 primary is contention winner and secondary is contention loser *Note 1:* Contention winner is also brackets first speaker. *Note 2:* This bit is reserved if either condition (1) or condition (2) holds. 1. The normal-flow send/receive mode is FDX (byte 7, bits 0−1 = 00). 2. All of the following are true. • The normal flow send/receive mode is HDX-FF (byte 7, bits 0−1 = 10). • Brackets are not used (byte 4, bit 7 = 0, byte 5, bit 7 = 0 , and byte 6, bit 2 = 0). • The FM profile is not 19 (byte 2 ≠ X'13'). • Symmetric responsibility for recovery is used (byte 7, bit 2 = 1).
	4−5	Reserved
	6	Control vectors included indicator: 0 control vectors are not included after the SLU name (bytes r+1−s) 1 control vectors are included after the SLU name (bytes r+1−s), in which case the BIND is called an *extended BIND*
	7	Half-duplex flip-flop reset states: 0 HDX-FF reset state is RECEIVE for the primary and SEND for the secondary (e.g., the secondary sends normal-flow requests first after session activation) 1 HDX-FF reset state is SEND for the primary and RECEIVE for the secondary (e.g., the primary sends normal-flow requests first after session activation) (only value defined for LU 6.2) *Note:* This bit is reserved unless both of the following conditions are true. 1. The normal-flow send/receive mode is half-duplex flip-flop (byte 7, bits 0−1 = 10). 2. Brackets are not used or the bracket state manager's reset state is INB (byte 6, bit 2 = 0).

BIND (BIND SESSION)

Byte	Bit	Content
		TS Usage
8	0	Staging indicator for session-level pacing of the secondary-to-primary normal flow: 0 the secondary send window size (byte 8, bits 2−7) and the primary receive window size (byte 13, bits 2−7) are for one-stage (or APPN hop-by-hop) pacing (The secondary send window size is always equal to the primary receive window size.) 1 the secondary send window size (byte 8, bits 2−7) and the primary receive window size (byte 13, bits 2−7) are for two-stage pacing *Note:* The meanings of 0 and 1 are reversed from the corresponding staging indicator for the primary-to-secondary normal flow.
	1	Reserved
	2−7	Secondary send window size, in binary, for session-level pacing: a value of 0 indicates that there will be no pacing of requests flowing from the secondary. *Note:* If pacing on a session stage in a particular direction is not to be performed, the values for the window size on that stage are set to 0. For example, if there is to be no pacing in the secondary to primary direction, the primary receive and secondary send window sizes are both set to 0.
9	0	Adaptive session-level pacing support (reserved for nonextended BIND) 0 adaptive pacing not supported by the sending node: pacing window values in bits 2−7 of bytes 8, 9, 12, and 13 specify the fixed value implied in each pacing response; a 0 value specifies no pacing 1 adaptive pacing supported by the sending node: pacing window values in bits 2−7 of bytes 8, 9, 12, and 13 specify the *preferred minimum value* for each ISOLATED PACING MESSAGE; a 0 value specifies that the preferred minimum value is as large as possible; each adaptive pacing partner initializes its own send window size to 1 at session activation *Note:* If the PLU specifies adaptive pacing in BIND, and the SLU is able to support adaptive pacing, the SLU responds with this bit set to 1 in RSP(BIND). If the PLU indicates it does not support adaptive pacing, or if the SLU does not support adaptive pacing, this bit will be set to 0 in RSP(BIND). See Chapter 4, "Request/Response Headers (RHs)" for further discussion of adaptive pacing.
	1	Reserved
	2−7	Secondary receive window size, in binary, for session-level pacing: a value of 0 causes the boundary function to substitute the value set by a system definition pacing parameter (if the system definition includes such a parameter) before it sends the BIND RU toward the secondary node; a value of 0 received at the secondary is interpreted to mean no pacing of requests flowing to the secondary. When fixed session-level pacing is used (byte 9, bit 0 = 0), this value is the fixed window size for the primary-to-secondary direction of the session stage. When adaptive session-level pacing is used (byte 9, bit 0 = 1), this value is the preferred minimum window size the primary end of the session stage recommends the secondary end of the session stage place in the IPMs it sends.

BIND (BIND SESSION)

Byte	Bit	Content
10		Maximum RU size sent on the normal flow by the secondary half-session. Bit 0 is interpreted as follows. 1. If bit 0 is set to 0, no maximum is specified and the remaining bits 1−7 are ignored. 2. If bit 0 is set to 1 (only value defined for LU 6.2), the byte is interpreted as X'ab' = a×2^b (Notice that, by definition, a≥8 and therefore X'ab' is a normalized floating point representation.) See Figure 5-1 on page 5-16 for all possible values.
11		Maximum RU size sent on the normal flow by the primary half-session: identical encoding as described for byte 10
12	0	Staging indicator for session-level pacing of the primary-to-secondary normal flow: 0 the primary send window size (byte 12, bits 2−7) and the secondary receive window size (byte 9, bits 2−7) are for two-stage pacing 1 the primary send window size (byte 12, bits 2−7) and the secondary receive window size (byte 9, bits 2−7) are for one-stage (or APPN hop-by-hop) pacing (The primary send window size is always equal to the secondary receive window size.) *Note:* The meanings of 0 and 1 are reversed from the corresponding staging indicator for the secondary-to-primary normal flow (byte 8, bit 0).
	1	Reserved
	2−7	Primary send window size, in binary, for session-level pacing: a value of 0 causes the value set by a system definition pacing parameter (if the system definition includes such a parameter) to be assumed for the session; if this is also 0, it means no pacing of requests flowing from the primary (For one-stage pacing in the primary-to-secondary direction, this field is redundant with, and will indicate the same value as, the secondary receive window size—see byte 9, bits 2−7, above.)
13	0−1	Reserved
	2−7	Primary receive window size, in binary, for session-level pacing: a value of 0 means no pacing of requests flowing to the primary (For one-stage pacing in the secondary-to-primary direction, this field is redundant with, and will indicate the same value as, the secondary send window size—see byte 8, bits 2−7, above.)
		PS Profile
14	0	PS Usage field format: 0 basic format (only value defined)
	1−7	LU type: 0000000 LU type 0 0000001 LU type 1 0000010 LU type 2 0000011 LU type 3 0000100 LU type 4 0000110 LU type 6 0000111 LU type 7

BIND (BIND SESSION)

Byte	Bit	Content
		PS Usage field *Note:* The following format for bytes 15−25 applies only to LU 6.2; for information on PS usage bytes 15−25 for other than LU 6.2 (indicated by byte 14, bits 1−7 = 0000110 and byte 15 = 00000010), see *SNA: Sessions Between Logical Units*. X'02' Level 2 (i.e., LU 6.2)
16−22		Reserved
23		Security Support Indicators
	0−2	Retired
	3	Conversation-level security support: 0 Access Security Information field will not be accepted on incoming FMH-5s. 1 Access Security Information field will be accepted on incoming FMH-5s.
	4−5	Reserved
	6	Already-verified function support: 0 Already-Verified indicator will not be accepted on incoming FMH-5s. 1 Already-Verified indicator will be accepted on incoming FMH-5s.
	7	Persistent verification capability: 0 Persistent Verification indicator is not supported on incoming FMH-5s. 1 Persistent Verification indicator is supported on incoming FMH-5s.
24	0	Reserved
	1−2	Synchronization level: 01 confirm is supported 10 confirm, sync point, and backout are supported
	3	Reserved
	4−5	Responsibility for session reinitiation (reserved when bit 6 of this byte is set to 1): 00 operator controlled 01 primary half-session will reinitiate 10 secondary half-session will reinitiate 11 either may reinitiate
	6	Parallel session support for LU-LU pair: 0 not supported 1 supported
	7	Change Number of Sessions GDS variable flow support (set to 1 if byte 24, bit 6 = 1): 0 not supported 1 supported
25	0	Reserved
	1	Limited-resource session indicator: 0 not a limited-resource session and thus the contention-winner LU will not deactivate it when it is no longer busy 1 a limited-resource session and thus the contention-winner LU will deactivate it when it is no longer busy
	2−5	Reserved

BIND (BIND SESSION)

Byte	Bit	Content
	6–7	Usage varies by LU type
	•	*For LU types 4 and 7:*
	6	Reserved
	7	System microcode update offer: 0 not offered 1 offered (used only to first active LU)
	•	*For all other LU types:*
	6–7	Length-checked compression options (see "Length-Checked Compression" for details of the compression header): 00 no compression 01 compression offered — compression to be determined by the SLU 10 reserved 11 compression mandated *Note:* On extended BINDs, the Length-Checked Compression (X'66') control vector serves to define the requested compression options.
		End of PS Usage Field
26–k		Cryptography Options
26	0–1	Private cryptography options (reserved for LU 6.2): 00 no private cryptography supported 01 private cryptography supported: the session cryptography key and cryptography protocols are privately supplied by the end user
	2–3	Session-level cryptography options: 00 no session-level cryptography supported 01 session-level selective cryptography supported; all cryptography key management is supported by the SSCP and LU; exchange (via + RSP(BIND)) and verification (via CRV) of the cryptography session-seed value is supported by the LUs for the session; all FMD requests carrying ED are enciphered/deciphered by the TCs 10 reserved 11 session-level mandatory cryptography supported; all cryptography key management is supported by the SSCP and LU; exchange (via + RSP(BIND)) and verification (via CRV) of the cryptography session-seed value is supported by the LUs for the session; all FMD requests are enciphered/deciphered by TC
	4–7	Session-level cryptography options field length: X'0' no session-level cryptography specified; following additional cryptography options fields (bytes 27–k) omitted X'9' session-level cryptography specified; additional options follow in next nine bytes
27	0–1	Session cryptography key encipherment method: 00 session cryptography key enciphered under SLU master cryptography key using a seed value of 0 (only value defined)
	2–4	Reserved

BIND (BIND SESSION)

Byte	Bit	Content
	5–7	Cryptography cipher method: 000 block chaining with seed and cipher text feedback, using the Data Encryption Standard (DES) algorithm (only value defined)
28–k		Session cryptography key enciphered under secondary LU master cryptography key; an 8-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FMD requests
k+1–m		Network Services (NS) Primary LU Name Field (always present) This parameter is always network-qualified for implementations at the current level of SNA; back-level implementations may omit it.
k+1		Length of primary LU name (values 1 to 17 are valid) *Note:* Value 0 is retired.
k+2–m		Primary LU name or, if the secondary LU issued an INIT-SELF (or INIT-OTHER), the uninterpreted name as carried in that RU (and also in CDINIT for a cross-domain session)
m+1–n		User Data Field
m+1		Length of user data *Note:* X'00' = no User Data field present; if unstructured user data present, values 1 to 65 are valid.
m+2–n		User data
m+2		User data key: X'00' Structured subfields follow (only value defined for LU 6.2). *Note:* Individual structured subfields may be omitted entirely. When present, they appear in ascending subfield-number order. ¬X'00' First byte of unstructured user data.
For unstructured user data:		
m+3–n		Remainder of unstructured user data
For structured user data:		
m+3–n		Structured subfields (For detailed definitions, see "Introduction" on page 7-1.)
n+1–p		User Request Correlation Field (present only if carried in INIT from SLU, or if Secondary LU name field or control vectors are included)
n+1		Length of user request correlation (URC) field (values 0 to 12 are valid) *Note:* X'00' = no URC present.
n+2–p		URC: LU-defined identifier (present only if carried in INIT from SLU)
p+1–r		Network Services (NS) Secondary LU Name Field (present only for negotiable BINDs and for non-negotiable BINDs that include control vectors) This parameter is always network-qualified for implementations at the current level of SNA; back-level implementations may omit it.

BIND (BIND SESSION)

Byte	Bit	Content
p+1		Length of secondary LU name (values 1 to 17 are valid) *Note:* Value 0 is retired.
p+2−r		Secondary LU name

Bytes r+1−s are included only if byte 7, bit 6 specified that control vectors are included after the SLU name.

r+1−s		Control vectors, as described in "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). X'0E' Network Name control vector: PLU network name, X'F3' (present in extended BINDs when bytes k+2−m contain a non-network-qualified name) X'0E' Network Name control vector: CP network name, X'F4' (present in extended BINDs when neither the Fully-Qualified PCID [X'60'] control vector nor the Route Selection [X'2B'] control vector contains the CP[PLU] name) X'2B' Route Selection control vector (present when BIND sender has the information available as the result of a Locate search and the session-route calculation) X'2C' COS/TPF control vector (present when the BIND sender supports mode-to-COS mapping or when it received the control vector on a Locate search reply) X'2D' Mode control vector (present in non-LU6.2 extended BIND unless the default mode name — eight X'40' characters — is intended) X'60' Fully-Qualified PCID control vector (present on an *extended BIND*) X'66' Length-Checked Compression control vector (present when compression is supported) *Note:* The receiving LU simply ignores unrecognized control vectors.
Note:		The length of the BIND RU cannot exceed 256 or 512 bytes. The length of the basic BIND RU is restricted to 256 bytes including the X'0E', X'2C', X'2D', and X'60' control vectors; any additional control vectors may cause the length to increase up to 512 bytes.
Note:		If the last byte of a format 0 BIND request not having control vectors is a length field and that field is 0, that byte may be omitted from the BIND request.

	Mantissa (a)							
Exponent (b)	8	9	A (10)	B (11)	C (12)	D (13)	E (14)	F (15)
0	8	9	10	11	12	13	14	15
1	16	18	20	22	24	26	28	30
2	32	36	40	44	48	52	56	60
3	64	72	80	88	96	104	112	120
4	128	144	160	176	192	208	224	240
5	256	288	320	352	384	416	448	480
6	512	576	640	704	768	832	896	960
7	1024	1152	1280	1408	1536	1664	1792	1920
8	2048	2304	2560	2816	3072	3328	3584	3840
9	4096	4608	5120	5632	6144	6656	7168	7680
A (10)	8192	9216	10240	11264	12288	13312	14336	15360
B (11)	16384	18432	20480	22528	24576	26624	28672	30720
C (12)	32768	36864	40960	45056	49152	53248	57344	61440
D (13)	65536	73728	81920	90112	98304	106496	114688	122880
E (14)	131072	147456	163840	180224	196608	212992	229376	245760
F (15)	262144	294912	327680	360448	393216	425984	458752	491520

Note: A value of X'ab' in byte 10 or byte 11 of BIND represents $a \times 2^b$. For example, X'C5' represents (in decimal) $12 \times 2^5 = 384$.

Figure 5-1. RU Sizes Corresponding to Values X'ab' in BIND

BIS (BRACKET INITIATION STOPPED)

LU → LU, Norm; DFC

BIS is sent by a half-session to indicate that it will not attempt to begin any more brackets.

BIS (BRACKET INITIATION STOPPED)

Byte	Bit	Content
0		X'70' request code

CANCEL (CANCEL)

LU → LU, Norm; DFC

CANCEL may be sent by a half-session to terminate a partially sent chain of FMD requests. CANCEL may be sent only when a chain is in process. The sending half-session may send CANCEL to end a partially sent chain if a negative response is received for a request in the chain, or for some other reason. This RU is not used for LU 6.2.

CANCEL (CANCEL)

Byte	Bit	Content
0		X'83' request code

CHASE (CHASE)

LU → LU, Norm; DFC

CHASE is sent by a half-session to request the receiving half-session to return all outstanding normal-flow responses to requests previously received from the issuer of CHASE. The receiver of CHASE sends the response to CHASE after processing (and sending any necessary responses to) all requests received before the CHASE. This RU is not used for LU 6.2.

CHASE (CHASE)

Byte	Bit	Content
0		X'84' request code

CLEAR (CLEAR)

PLU → SLU, Exp; SC

CLEAR is sent by primary session control to reset the data traffic FSMs and subtrees (for example, brackets, pacing, sequence numbers) in the primary and secondary half-sessions (and boundary function, if any). CLEAR also resets compression and decompression tables in sessions using length-checked compression. This RU is not used for LU 6.2.

CLEAR (CLEAR)

Byte	Bit	Content
0		X'A1' request code

CRV (CRYPTOGRAPHY VERIFICATION)

PLU → SLU, Exp; SC

CRV, a valid request only when session-level cryptography was selected in BIND, is sent by the primary LU session control to verify cryptography security and thereby enable sending and receiving of FMD requests by both half-sessions.

CRV (CRYPTOGRAPHY VERIFICATION)

Byte	Bit	Content
0		X'C0' request code
1−8		A transform of the (deciphered) cryptography session-seed value received (enciphered) in bytes 28−k of +RSP(BIND), re-enciphered under the session cryptography key using a seed value of 0; the transform is the cryptography session-seed value with the first four bytes inverted. *Note:* The cryptography session-seed is used as the seed for all session-level cryptography encipherment and decipherment provided for FMD RUs.

DACTLU (DEACTIVATE LOGICAL UNIT)

SSCP → LU, Exp; SC

DACTLU is sent to deactivate the session between the SSCP and the LU.

DACTLU (DEACTIVATE LOGICAL UNIT)

Byte	Bit	Content
0		X'0E' request code
Note:		End of short (1-byte) request
1		Type of deactivation requested: X'01' normal deactivation X'03' session-outage notification (SON)

DACTLU (DEACTIVATE LOGICAL UNIT)

Byte	Bit	Content
2		Cause (reserved if byte 1 ≠ X'03'): X'07' virtual route inoperative: the virtual route serving the SSCP-LU session has become inoperative, thus forcing the deactivation of the session X'08' route extension inoperative: the route extension serving the SSCP-LU session has become inoperative, thus forcing the deactivation of the session X'09' hierarchical reset: the identified session is being deactivated because of a + RSP(ACTPU, Cold) X'0B' virtual route deactivated: the SSCP-LU session is being deactivated because of a forced deactivation of the virtual route being used by the session X'0C' SSCP or LU failure—unrecoverable: the SSCP-LU session had to be reset because of an abnormal termination; recovery from the failure was not possible X'0D' session override: the SSCP-LU session has to be deactivated because of a more recent session activation request for the SSCP to subarea PU session over a different virtual route X'0E' SSCP or LU failure—recoverable: the SSCP-LU session had to be deactivated because of an abnormal termination of the SSCP or LU of the session; recovery from the failure may be possible X'0F' cleanup: the SSCP is resetting its half-session before receiving the response from the LU being deactivated

DACTPU (DEACTIVATE PHYSICAL UNIT)

SSCP → PU, PU → SSCP, Exp; SC

DACTPU is sent to deactivate the session between the SSCP and the PU.

DACTPU (DEACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
0		X'12' request code
1		Type deactivation requested: X'01' final use, physical connection may be broken X'02' not final use, physical connection should not be broken X'03' session-outage notification (SON)

DACTPU (DEACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
2		Cause (not present if byte 1 ≠ X'03'): X'07' virtual route inoperative: the virtual route for the SSCP-PU session has become inoperative, thus forcing the deactivation of the SSCP-PU session X'08' route extension inoperative: the route extension serving the SSCP-PU session has become inoperative, thus forcing the deactivation of the SSCP-PU session X'09' hierarchical reset: the identified session is being deactivated because of a +RSP(ACTPU, Cold) X'0B' virtual route deactivated: the identified SSCP-PU session is being deactivated because of a forced deactivation of the virtual route being used by the session X'0C' SSCP or PU failure—unrecoverable: the identified SSCP-PU session had to be deactivated because of an abnormal termination of the SSCP or PU of the session; recovery from the failure was not possible X'0D' session override: the SSCP-PU session has to be deactivated because of a more recent session activation request for the SSCP to subarea PU session over a different virtual route X'0E' SSCP or PU failure—recoverable: the identified SSCP-PU session had to be deactivated because of an abnormal termination of the SSCP or PU of the session; recovery from the failure may be possible X'0F' cleanup: the SSCP is resetting its half-session before receiving the response from the PU that is being deactivated X'10' ALS reset: peripheral ALSs (and subordinate LUs and LU-LU sessions) owned by the sending SSCP should be reset X'11' give-back: the sending SSCP relinquishes ownership of resources; active LU-LU sessions should not be disrupted for LUs subordinate to ALSs whose nodes support ACTPU(ERP)

INIT-SELF Format 0 (INITIATE-SELF)

ILU → SSCP, Norm; FMD NS(s)

INIT-SELF from the ILU requests that the SSCP authorize and assist in the initiation of a session between the LU sending the request (that is, the ILU, which also becomes the OLU) and the LU named in the request (the DLU). This RU is not used for LU 6.2; refer to INIT-SELF Format 1.

INIT-SELF Format 0 (INITIATE-SELF)

Byte	Bit	Content
0–2		X'010681' NS header

INIT-SELF Format 0 (INITIATE-SELF)

Byte	Bit	Content
3	0–3	Format: 0000 Format 0: specifies a subset of the parameters shown in Format 1 of INIT-SELF (described separately, because the NS header differs in the first byte), with the receiver supplying default values
	4–5	Reserved
	6	PLU/SLU specification: 0 DLU is PLU (only value defined)
	7	0 initiate only (I): do not enqueue. 1 initiate/enqueue (I/Q): enqueue the request if it cannot be satisfied immediately
4–11		Mode name: an 8-character symbolic name (implementation and installation dependent) that identifies the set of rules and protocols to be used for the session; used by the SSCP(SLU) to select the BIND image that will be used by the SSCP(PLU) to build the CINIT request
12–m		Uninterpreted Name of DLU
12		Type: X'F3' logical unit
13		Length, in binary, of DLU name
14–m		EBCDIC character string
m+1–m+2		Retired
m+3–n		User Field
m+3		Length, in binary, of user data *Note:* X'00' = no user data is present.
m+4–n		User data: user-specific data that is passed to the primary LU on the CINIT request
m+4		User data key: X'00' structured subfields follow ¬X'00' first byte of unstructured user data *Note:* Individual structured subfields may be omitted entirely. When present, they appear in ascending field number order.
For unstructured user data		
m+5–n		Remainder of unstructured user data
For structured user data		
m+5–n		Structured subfields (For detailed definitions, see Chapter 7, "User Data Structured Subfields" on page 7-1.)
n+1–p		Control vectors, as described in the section "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT (see "Substructure Encoding/Parsing Rules" on page 8-1). X'34' LU Definition Override control vector (present if terminal operator entered one or more of the model terminal support override parameters with an implementation logon request)

INIT-SELF Format 0 (INITIATE-SELF)

Byte	Bit	Content
Note:		The following default values are supplied by the SSCP(ILU) receiving the Format 0 INIT-SELF request: • Queuing conditions (if queuing is specified): – Enqueue if session limit exceeded. – Enqueue this request FIFO, i.e., the request will be dequeued after the other requests already in the queue.

INIT-SELF Format 1 (INITIATE-SELF)

ILU → SSCP, Norm; FMD NS(s)

INIT-SELF from the ILU requests that the SSCP authorize and assist in the initiation of a session between the LU sending the request (that is, the ILU, which also becomes the OLU) and the LU named in the request (the DLU).

INIT-SELF Format 1 (INITIATE-SELF)

Byte	Bit	Content
0–2		X'810681' NS header
3	0–3	Format: 0001 Format 1
	4–7	Reserved
4		Type:
	0–1	01 initiate only (I): do not enqueue 11 initiate/enqueue (I/Q): enqueue the request if it cannot be satisfied immediately (See byte 5 for further specification of queuing conditions.)
	2–3	Reserved
	4	Reserved
	5	Reserved
	6	PLU/SLU specification: 0 DLU is PLU (only value defined)
	7	Reserved

INIT-SELF Format 1 (INITIATE-SELF)

Byte	Bit	Content
5		Queuing conditions for DLU:
	0	0 do not enqueue if session limit exceeded 1 enqueue if session limit exceeded
	1	0 do not enqueue if DLU is not currently able to comply with the PLU/SLU specification (as given in byte 4, bit 6) 1 enqueue if DLU is not currently able to comply with the PLU/SLU specification
	2–4	Reserved
	5–6	Queuing position/service: 01 enqueue this request FIFO, i.e., the request will be dequeued after the requests already in the queue
	7	Reserved
	Note:	Since queuing conditions are specified for the DLU only, the following default values are used by SSCP(OLU) for the OLU: • Enqueue if session limit exceeded. • Enqueue this request at the foot of the queue (FIFO).
6–7		Reserved
8–15		Mode name: an 8-character symbolic name (implementation and installation dependent) that identifies the set of rules and protocols to be used for the session; used by the SSCP(SLU) to select the BIND image that will be used by the SSCP(PLU) to build the CINIT request
16–n		Uninterpreted Name of DLU
16		Type: X'F3' logical unit
17		Length, in binary, of DLU name
18–n		EBCDIC character string
n+1–n+2		Retired
n+3–r(=n+3)		Reserved
r+1–s		User Request Correlation (URC) Field
r+1		Length, in binary, of URC *Note:* X'00' = no URC. (The length field is always present.)
r+2–s		URC: LU-defined identifier; may be returned by the SSCP in a subsequent NOTIFY to correlate a given session to this initiating request
s+1–t		Control vectors, as described in the section "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT (see "Substructure Encoding/Parsing Rules" on page 8-1). X'34' LU Definition Override control vector (present if terminal operator entered one or more of the model terminal support override parameters with an implementation logon request)

LUSTAT (LOGICAL UNIT STATUS)

LU → LU, Norm; DFC

LUSTAT is used by one half-session to send up to four bytes of status information to its paired half-session. The RU format allows the sending of either end-user information or LU status information. If the high-order two bytes of the status information are 0, the low-order two bytes carry end-user information and may be set to any value. In general, LUSTAT is used to report about failures and error recovery conditions for a local device of an LU.

LUSTAT (LOGICAL UNIT STATUS)

Byte	Bit	Content
0		X'04' request code
1–4		Status value + status extension field (two bytes each):
		X'0000' + 'uuuu' user status (no system-defined status) + user-defined field
		X'0001' + 'ccdd' component now available + component identification (see Note)
		X'0002' + 'rrrr' sender will have no (more) FMD requests to transmit during the time that this session remains active + reserved field
		X'0003' + 'ccdd' component entering attended mode of operation + component identification (see Note)
		X'0004' + 'ccdd' component entering unattended mode of operation + component identification (see Note)
		X'0005' + 'iiii' prepare to commit all resources required for the unit of work + information field: X'0001' request End Bracket be sent on next chain (only value defined)
		X'0006' + 'rrrr' no-op (used to allow an RH to be sent when no other request is available or allowed) + reserved field (only value defined for LU 6.2)
		X'0007' + 'rrrr' sender currently has no FMD requests to transmit (but may have later during the time that this session remains active) + reserved field
		X'0801' + 'ccdd' component not available (e.g., not configured) + component identification (see Note)
		X'0802' + 'ccdd' component failure (intervention required) + component identification (see Note)
		X'081C' + 'ccdd' component failure (permanent error) + component identification (see Note)
		X'0824' + 'ccdd' function canceled + reserved field
		X'082B' + 'ccdd' component available, but presentation space integrity lost + component identification (see Note)
		X'0831' + 'ccdd' component disconnected (power off or some other disconnecting condition) + component identification (see Note)
		X'0848' + 'rrrr' cryptography component failure + reserved field

LUSTAT (LOGICAL UNIT STATUS)

Byte	Bit	Content
		X'400A' + 'ssss' no-response mode not allowed + sequence number of the request specifying no-response
		Note: Values for cc byte are:
		X'00' LU itself rather than a specific LU component (For this cc value, dd = X'00'.)
		X'FF' The dd byte specifies the LU component medium class and device address. (See *SNA: Sessions Between Logical Units* for definitions of these terms and usage of the values according to LU type.)
		¬X'(00|FF)' LU component medium class and device address (For these cc values, dd = X'00'.)

NMVT (NETWORK MANAGEMENT VECTOR TRANSPORT)

SSCP ←→ PU Norm; FMD NS(ma)

NMVT carries management services (MS) requests and replies between an SSCP and a PU.

NMVT (NETWORK MANAGEMENT VECTOR TRANSPORT)

Byte	Bit	Content
0–2		X'41038D' NS header
3–4		Retired: Set to network address by subarea node sender; set to 0, the PU local address, by peripheral node sender; ignored by receivers implementing the current level of SNA
5–6	0–1	Reserved
	2–3	Retired: Set to 01 by subarea PU sender; set to 00 by peripheral node sender; ignored by receivers implementing the current level of SNA
	4–15	Procedure related identifier (PRID) *Note:* For unsolicited replies (byte 7, bit 0 = 0), the PRID field contains X'000'. For solicited replies (byte 7, bit 0 = 1), the PRID field echoes the PRID from the NMVT RU request. For requests that need no replies, this field contains X'000'.
7		Flags:
	0	Solicitation indicator: used only for PU-to-SSCP flow (reserved for SSCP-to-PU flow): 0 unsolicited NMVT 1 solicited NMVT
	1–2	Sequence field—used only for PU-to-SSCP flow (reserved for SSCP-to-PU flow): 00 only NMVT for this PRID 01 last NMVT for this PRID 10 first NMVT for this PRID 11 middle NMVT for this PRID

NMVT (NETWORK MANAGEMENT VECTOR TRANSPORT)

Byte	Bit	Content
	3	SNA Address List subvector indicator: 0 *For the SSCP-to-PU flow:* MS major vector in this NMVT does not contain an SNA Address List subvector *For the PU-to-SSCP flow:* MS major vector in this NMVT does not contain an SNA Address List subvector, or it contains an SNA Address List subvector that does not require address-to-name translation by the SSCP 1 *For the SSCP-to-PU flow:* MS major vector in this NMVT contains an SNA Address List subvector *For the PU-to-SSCP flow:* MS major vector in this NMVT contains an SNA Address List subvector that requires address-to-name translation by the SSCP
	4–7	Reserved
8–m		One or more MS major vectors, as described (using 0-origin indexing) in the table in Chapter 9, "SNA/MS Encodings" on page 9-1.

NOTIFY (NOTIFY)
SSCP ←→ LU, Norm; FMD NS(s)

NOTIFY is used to send information from an SSCP to an LU, or from an LU to an SSCP. NOTIFY carries information in the form of a (vector key, vector data) pair.

NOTIFY (NOTIFY)

Byte	Bit	Content
0–2		X'810620' NS header
3–p		One NOTIFY vector as described in detail below: X'03' ILU/TLU Notification: used to inform the sender of an INIT or TERM request of the status of the procedure X'0C' LU-LU Session Services Capabilities: used to inform the SSCP having an active session with the sending LU of the current LU-LU session services capability of that LU

ILU/TLU Notification NOTIFY Vector

ILU/TLU Notification NOTIFY Vector

Byte	Bit	Content
0		Key: X'03'

ILU/TLU Notification NOTIFY Vector

Byte	Bit	Content
1		Status: X'00' SSCP(OLU) and SSCP(DLU) not logically connected, i.e., no session or session setup path (if rerouting is required) exists between them X'01' session terminated X'02' session set up X'03' procedure error
2–9		Reserved
10		Reason (defined for Status field value of X'03' only) *Note:* There are two encodings of the Reason byte: • If bit 4 = 0, the Reason byte is encoded for a setup procedure error. • If bit 4 = 1, the Reason byte is encoded for a takedown procedure error.
		Setup Procedure Error
	0	1 CINIT error in reaching the PLU
	1	1 BIND error in reaching the SLU
	2	1 setup reject at the PLU
	3	1 setup reject at the SLU
	4	0 setup procedure error
	5	Reserved
	6	1 setup reject at SSCP
	7	Reserved
		Takedown Procedure Error
	0	1 CTERM error in reaching the PLU
	1	1 UNBIND error in reaching the SLU
	2	1 takedown reject at the PLU
	3	1 takedown reject at the SLU
	4	1 takedown procedure error
	5	1 takedown reject at the SSCP
	6	0 (see following Note)
	7	Reserved
		Note: For bits 4 and 6, the bit combination of 11 is set aside for implementation internal use and will not be otherwise defined.
11–14		Sense data (defined for Status value of X'03' only)
15–m		Session key, as described in "Session Keys" on page 8-27 *Note:* The following session key is used: X'06' network name pair: PLU and SLU
m+1–n		User Request Correlation (URC) Field
m+1		Length, in binary, of the URC
m+2–n		URC: the URC carried in the URC field in INIT (bytes r+1–s) or TERM (bytes n+3–p); used to correlate the NOTIFY to the initiating or terminating requests

LU-LU Session Services Capabilities NOTIFY Vector

Note: This NOTIFY vector should not be confused with control vector X'0C', which carries similar information.

LU-LU Session Services Capabilities NOTIFY Vector

Byte	Bit	Content
0		Key: X'0C'
1		Length of Vector Data field, encoded in binary
2−m		Vector Data
2		LU-LU session capability:
	0−3	Reserved
	4−7	Secondary LU capability: 0000 SLU capability is inhibited: sessions can be neither queued nor started 0001 SLU capability is disabled: sessions can be queued but not started 0010 reserved 0011 SLU capability is enabled: sessions can be queued or started
3−4		Retired (set to X'0001')
5−7		Retired
8−15(=m)		Retired (set to all space (X'40') characters, or omitted)

NSPE (NS PROCEDURE ERROR)

SSCP → ILU or TLU, Norm; FMD NS(s)

NSPE is used by the SSCP to inform an ILU or TLU that a session initiation or termination attempt has failed after a positive response has been sent to the corresponding initiation or termination request. (NSPE is used only if Format 0 of INIT-SELF or TERM-SELF was issued. Otherwise, NOTIFY is used.)

NSPE (NS PROCEDURE ERROR)

Byte	Bit	Content
0−2		X'010604' NS header
Note:		The remainder of this RU has two formats: a *comprehensive* form and a *condensed* form, based upon the setting of bit 7 of the Reason byte (byte 3). The choice is implementation-dependent.
		Comprehensive Format

NSPE (NS PROCEDURE ERROR)

Byte	Bit	Content
3		Reason *Note:* There are two encodings of the Reason byte in the comprehensive format: • If bit 4 = 0, the Reason byte is encoded for a setup procedure error. • If bit 4 = 1, the Reason byte is encoded for a takedown procedure error.
		Setup Procedure Error
	0	1 CINIT error in reaching the PLU
	1	1 BIND error in reaching the SLU
	2	1 setup reject at the PLU
	3	1 setup reject at the SLU
	4	0 setup procedure error
	5	Reserved
	6	1 setup reject at SSCP
	7	1 comprehensive format of Reason byte
		Takedown Procedure Error
	0	1 CTERM error in reaching the PLU
	1	1 UNBIND error in reaching the SLU
	2	1 takedown reject at the PLU
	3	1 takedown reject at the SLU
	4	1 takedown procedure error
	5	1 takedown reject at SSCP
	6	0 see following Note
	7	1 comprehensive format of Reason byte
		Note: The bit combination of 11 for bits 4 and 6 is set aside for implementation internal use and will not be otherwise defined.
4–7		Sense data
8–n		Session key, as described in the section "Session Keys" on page 8-27 *Note:* One of the following session keys is used: X'06' uninterpreted name pair: PLU and SLU, respectively (only value defined)
		Condensed Format
3		Reason:
	0	1 CINIT error in reaching the PLU
	1	1 BIND error in reaching the SLU
	2	1 setup reject at the PLU
	3	1 setup reject at the SLU
	4	1 takedown failure
	5	1 takedown reject at SSCP
	6	1 setup reject at SSCP
	7	0 condensed format
4–m		Uninterpreted name of PLU
4		Type: X'F3' logical unit
5		Length, in binary, of PLU name
6–m		EBCDIC character string

NSPE (NS PROCEDURE ERROR)

Byte	Bit	Content
m+1−n		Uninterpreted name of SLU
m+1		Type: X'F3' logical unit
m+2		Length, in binary, of SLU name
m+3−n		EBCDIC character string

QC (QUIESCE COMPLETE)

LU → LU, Norm; DFC

QC is sent by a half-session after receiving QEC, to indicate that it has quiesced. This RU is not used for LU 6.2

QC (QUIESCE COMPLETE)

Byte	Bit	Content
0		X'81' request code

QEC (QUIESCE AT END OF CHAIN)

LU → LU, Exp; DFC

QEC is sent by a half-session to quiesce its partner half-session after it (the partner) finishes sending the current chain (if any). This RU is not used for LU 6.2.

QEC (QUIESCE AT END OF CHAIN)

Byte	Bit	Content
0		X'80' request code

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

PU → SSCP, Norm; FMD NS(ma)

RECFMS has been retired from SNA for T2 nodes.

RECFMS permits the passing of maintenance related information from a PU to management services at the SSCP.

Consult product documentation for further information on product support.

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
0–2		X'410384' NS header
3–4		CNM target ID, as specified in bytes 5–6, bits 2–3
5–6	0–1	Reserved
	2–3	CNM target ID descriptor: 00 byte 4 contains a local address for a PU or LU in a T2 node or an LSID for a PU or LU in a T1 node; byte 3 is reserved 01 bytes 3–4 contain the element address of a link, adjacent link station, PU, or LU in the origin subarea, if ENA is supported; otherwise, its network address
	4–15	Procedure related identifier (PRID) (see Note below)
7		Request-Specific Information
	0	Solicitation indicator: 0 unsolicited request 1 reply request
	1	Not-last request indicator: 0 last request in a series of related unsolicited or reply requests, e.g., last reply request in a series corresponding to a single soliciting request 1 not last request
	2–7	Request-specific type code (see below)
Note:		For reply (i.e., solicited) requests, bytes 3–6 and byte 7, bits 2–7, echo the corresponding fields in the CNM header received in the request that solicited the reply requests. For unsolicited requests, these fields—the CNM target ID descriptor, the CNM target ID, the PRID, and the request-specific information—are generated by the request sender. For unsolicited requests, the PRID field contains X'000'. The PU does not interleave requests belonging to different series of related unsolicited requests from the same target.
8–13		Node Identification
8–11	0–11	Block number: an IBM product-specific number; see the individual product specifications for the specific values used
	12–31	ID number: a binary value that, together with the block number, identifies a specific station uniquely within a customer network installation; the ID number can be assigned in various ways, depending on the product; see the individual product specifications for details

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
12–13		Reserved
7–n		Alert (retired: supported only for PUs not at the current level of SNA)
7	0–1	00 (only value defined—Alert is always sent unsolicited and as a single RU)
	2–7	Type code: 000000
8–13		Node Identification
8–11	0–11	Block number
	12–31	ID number
12–13		Reserved
14–19		Alert Classification
14	0–1	Format: 01 format 1 (only value defined)
14	2–7	Reserved
15	0–3	Alert type: indicates the reason for the Alert being generated and differentiates between errors, operational problems, performance problems, and other exceptional conditions; valid Alert types are: X'1' permanent error: cannot be retried or recovered without help external to the SNA node X'2' temporary error: recovered within recovery procedure limit X'3' performance: exceeded performance parameter threshold X'4' operational or procedure: unsupported or invalid use, busy X'5' application generated X'6' operator triggered X'7' SNA summary: exceeded threshold count of SNA negative responses
	4–7	Major probable cause: indicates the general category of the probable cause, e.g., hardware, software, or protocol failure; valid major probable cause (details of these causes are given in specific implementation documentation): X'1' hardware X'2' software X'3' link connection: characterized by transmission medium, modem, DTE-DCE cable, drivers, X'4' protocol: invalid response or command sequence, system definition error X'5' environment: thermal, installation restriction X'6' removable media, e.g., paper, cards, tape, pack, diskette X'7' hardware or software X'8' logical X'9' operator of sending product X'F' undetermined
16		Minor probable cause: indicates the lowest level category with which the Alert may be associated, e.g., printer, power, program, X.25 network; valid minor probable cause (details of these causes are given in specific implementation documentation): X'01' base processor

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content	
		X'02'	service processor for support of maintenance services
		X'03'	microcode *Note:* Microcode may be classified as IBM Licensed Internal Code. See "Notices" at the beginning of this document for more information.
		X'04'	main storage
		X'05'	DASD drive
		X'06'	printer
		X'07'	card reader/punch
		X'08'	tape drive
		X'09'	keyboard
		X'0A'	selector pen
		X'0B'	magnetic stripe reader
		X'0C'	display or printer
		X'0D'	display unit
		X'0E'	remote product: error attributed to product at adjacent link station on this link
		X'0F'	power internal to this product
		X'10'	I/O attached controller if distinguishable from drive
		X'11'	communications controller scanner
		X'12'	communications link adapter
		X'13'	link adapter
		X'14'	channel adapter: secondary attachment to System/370 channel
		X'15'	loop adapter: attachment to loop communication link
		X'16'	adapter for directly attaching devices
		X'17'	miscellaneous adapter
		X'18'	System/370 channel
		X'19'	link: transmission medium—ownership unknown
		X'1A'	link: common carrier transmission medium
		X'1B'	link: customer transmission medium
		X'1C'	loop: transmission medium—ownership unknown
		X'1D'	loop: common carrier transmission medium
		X'1E'	loop: customer transmission medium
		X'1F'	X.21 link connection external to this product
		X'20'	X.25 network external to this product
		X'21'	local X.21 interface: DTE-DCE
		X'22'	local X.25 interface: DTE-DCE
		X'23'	local modem
		X'24'	remote modem
		X'25'	local modem interface: DTE-DCE
		X'26'	remote modem interface: DTE-DCE
		X'27'	local probe
		X'28'	remote probe
		X'29'	local probe interface
		X'2A'	remote probe interface
		X'2B'	network connection

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content	
		X'2C'	IBM host program if not distinguishable as control program, application, or access method
		X'2D'	IBM host application program supplied by IBM
		X'2E'	IBM host telecommunication access method
		X'2F'	customer host application program
		X'30'	IBM communication controller program
		X'31'	IBM control program
		X'32'	remote modem interface or remote product
		X'33'	transmission medium or remote modem
		X'34'	SDLC format exception
		X'35'	BSC format exception
		X'36'	start/stop format exception
		X'37'	SNA format exception
		X'38'	power external to product
		X'39'	thermal
		X'3A'	paper
		X'3B'	tape
		X'3C'	DASD: removable media
		X'3D'	card
		X'3E'	magnetic stripe card
		X'3F'	negative SNA response
		X'40'	system definition error (whether diskette loaded, keyed, or otherwise customized)
		X'41'	installation restrictions
		X'42'	adjacent link station offline: no status received
		X'43'	adjacent link station busy (switched link)
		X'44'	controller or device
		X'45'	local probe or modem
		X'46'	tape or drive
		X'47'	card reader/punch or display/printer
		X'48'	controller application program
		X'49'	keyboard or display
		X'4A'	storage control unit
		X'4B'	channel or storage control unit
		X'4C'	storage control unit or controller
		X'4D'	control unit
		X'4E'	DASD data or media or drive
		X'4F'	DASD data or media
		X'50'	diskette
		X'51'	diskette or drive
		X'FF'	undetermined
17		Reserved	

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
18		User action code: 0 reserved ¬0 a code associated with predefined text that describes user actions appropriate to the cause
19		Reserved
20−m		Appended CNM vectors (described at the end of this RU): zero or more CNM vectors may be appended to the request to convey data available to the PUMS when the Alert event was originated; appended vectors are ordered according to the binary value of the Vector Type field (nondescending) *Note:* The sending of information in appended CNM vectors does *not* cause reset of any counters.
m+1(=n)		X'00' indicating end of appended vectors
7−17		SDLC Test Command/Response Statistics
7	0 1 2−7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000001; the CNM target ID identifies a PU T1|2
8−13	 0−11 12−31	Node identification Block number ID number
12−13		Reserved
14−15		Counter: the number of times the secondary SDLC station has received an SDLC Test command with or without a valid FCS
16−17		Counter: the numberof times the secondary SDLC station has received an SDLC Test command with a valid FCS and has transmitted an SDLC Test response *Note:* All counters are in binary.
7−22		Summary error data
7	0 1 2−7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000010; the CNM target ID identifies a PU
8−13	 0−11 12−31	Node identification Block number ID number
12−13		Reserved
14−16		Summary counter validity mask:
14	0 1 2 3−7	Set to 1 if product error counter is valid Set to 1 if communication adapter error counter is valid Set to 1 if SNA negative response counter is valid Reserved
15		Reserved

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
16	0–6	Reserved
	7	Communications adapter error flag for products implementing unsolicited RECFMS types 02 or 03; otherwise, reserved: 0 no cumulative communication adapter errors 1 indicates presence of communication adapter errors not yet reported by RECFMS 03
17–18		Product error counter: a count for the product identified by the Node Identification field (bytes 8–13) of certain product-detected hardware errors whose origins are failures designated as internal by that product's own logic capability (The identified product has the responsibility for further isolation of these failures using its own product-specific problem determination and maintenance procedures.)
19–20		Communication adapter error counter for communication adapter errors whose source is either external or internal to the product identified by the node ID; this field is reserved in products reporting counter overflows via unsolicited RECFMS type 02 or 03
21–22		Count of SNA negative responses originating at this node *Note:* All counters are in binary.
7–n		Communication Adapter Error Statistics: counts of selected errors, useful for problem determination, that have been supplied by the communication adapter (For these errors, the RECFMS Type 000010 communication adapter error counter is always incremented; the RECFMS Type 000010 product error counter is also incremented for those errors classified as internal errors by the product identified by the node ID.)
7	0	Solicitation indicator (see above)
	1	Not-last request indicator (see above)
	2–7	Type code: 000011; the CNM target ID identifies a PU T1|2
8–13		Node identification
	0–11	Block number
	12–31	ID number
12–13		Reserved
14		Communication adapter error counter sets: X'01' counter set 1 X'02' counter set 2 X'03' counter set 3 X'04' counter set 4 X'05' counter set 5 (retired: supported only for PUs not at the current level of SNA) X'06' counter set 6 (retired: supported only for PUs not at the current level of SNA)
15–n		Data for Counter Sets 1 and 2
15–17		Communication adapter counter validity mask bytes

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
15		Mask byte 1 (bit is set to 1 if the counter is valid):
	0	Nonproductive time-out or receive overrun counter
	1	Idle time-out counter
	2	Write retry counter
	3	Overrun counter
	4	Underrun counter
	5	Connection problem counter
	6	FCS error counter
	7	Primary station abort counter
16		Mask byte 2 (bit is set to 1 if the counter is valid):
	0	Command reject counter
	1	SDLC DCE error counter
	2	Write time-out counter
	3	Invalid status counter
	4	Communication adapter machine check counter
	5–7	Reserved
17		Reserved
18		Nonproductive time-out counter: no valid SDLC frames have been received within the time interval specified by the communication adapter; or receive overrun counter: the line is "hung" or insufficient buffer space has been allocated *Note:* Receive overrun applies only to counter set 2.
19		Idle time-out counter: no SDLC Flag octets received for *n* seconds, where *n* is specified by the communication adapter
20		Write retry counter: the number of retransmissions of one or more SDLC I-frames
21		Overrun counter: the number of times one or more received characters have been overlaid
22		Underrun counter: the number of times one or more characters have been transmitted more than once
23		Connection problem counter: incremented by *1* for every *n* retries of commands that establish connection with a station, when RLSD drops, or whenever write retry is updated—*n* is specified by the communication adapter
24		FCS error counter: the number of times a received SDLC frame had an invalid FCS
25		Primary station abort counter: number of times seven or more consecutive *1* bits have been received
26		SDLC command reject counter
27		DCE error counter: number of DCE interrupts or other unexpected conditions (e.g., "data set ready" drops)
28		Write time-out counter: number of time-outs during write operations, e.g., because of transmit clock failures

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
29		Invalid status counter: number of times status generated by the adapter was not meaningful
30(=n)		Communication adapter machine check counter: number of times the communication adapter has been identified as causing a machine check *Note:* All counters are in binary.
15–n		Data for Counter Set 3
15–17		Communication adapter counter validity mask (bit is set to 1 if the counter is valid):
15	0	Total transmitted I-frames counter
	1	Write retry counter
	2	Total received I-frames counter
	3	FCS error counter
	4	SDLC command reject counter
	5	DCE error counter
	6	Nonproductive time-out counter
	7	Reserved
16–17		Reserved
18–19		Total transmitted I-frames counter: the total number of SDLC I-frames transmitted successfully
20–21		Write retry counter: the number of retransmissions of one or more SDLC I-frames
22–23		Total received I-frames counter: the number of SDLC I-frames successfully received
24–25		FCS error counter: the number of SDLC frames received with FCS errors
26–27		SDLC command reject counter
28–29		DCE error counter: the number of DCE interrupts and other unexpected conditions (e.g., "data set ready" drops)
30–31(=n)		Nonproductive time-out counter: the number of times an SDLC frame has not been received within the time interval specified by the adapter *Note:* All counters are in binary.
15–n		Data for Counter Set 4 *Note:* For a definition of adapter, control unit, and System/370 channel commands, and orders see implementation documentation.
15–17		Adapter counter validity mask bytes

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
15		Mask byte 1 (bit is set to 1 if the counter is valid):
	0	Command-reject-while-not-initialized counter
	1	Command-not-recognized counter
	2	Sense-while-not-initialized counter
	3	Channel-parity-check-during-selection-sequence counter
	4	Channel-parity-check-during-data-write-sequence counter
	5	Output-parity-check-at-control-unit counter
	6	Input-parity-check-at-control-unit counter
	7	Input-parity-check-at-adapter counter
16		Mask byte 2 (bit is set to 1 if the counter is valid):
	0	Data-error-at-adapter counter
	1	Data-stop-sequence counter
	2	Short-frame-or-length-check counter
	3	Connect-received-when-already-connected counter
	4	Disconnect-received-while-PU-active counter
	5	Long-RU counter
	6	Connect-parameter-error counter
	7	Read-Start-Old-received counter
17		Reserved
18		Command-reject-while-not-initialized counter: an initial Control command containing a valid Connect order was not received prior to a Restart Reset, Read Start 0/1, Write Start 0/1, Read, Write, or Write Break command
19		Command-not-recognized counter: control unit channel adapter received a command code that it did not recognize (invalid or not supported)
20		Sense-while-not-initialized counter: Sense command was received in response to the initial asynchronous interrupt (device-end, unit check), or Sense command was received without a preceding unit check ending status
21		Channel-parity-check-during-selection-sequence counter: control unit channel adapter detected a parity error from the channel during the selection sequence from the channel
22		Channel-parity-check-during-data-write-sequence counter: control unit channel adapter detected a parity error on channel bus-out during a channel Write operation
23		Output-parity-check-at-control-unit counter: control unit channel adapter detected a control unit parity error during a channel Write operation
24		Input-parity-check-at-control-unit counter: control unit detected a control unit parity error during a channel Read operation
25		Input-parity-check-at-adapter counter: control unit channel adapter detected that it transmitted bad parity on channel bus-in during a channel Read operation
26		Data-error-at-adapter counter: control unit detected a channel adapter error during an internal channel adapter cycle-steal operation

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
27		Data-stop-sequence counter: the number of data bytes accepted by the System/370's Read command was less than that specified in Connect
28		Short-frame-or-length-check counter: a minimum four bytes have not been transferred as a link header; or the byte count specified in the first two bytes of the header did not equal the number of bytes received during a Control, Write, or Write Break operation
29		Connect-received-when-already-connected counter: a Connect was received when the control unit was already connected; this is an error condition and the PU is deactivated
30		Disconnect-received-while-PU-active counter: a Disconnect order was received from the System/370 while the PU is active (i.e., with no DACTPU preceding the Disconnect); this is an error condition
31		Long-RU counter: primary link station has sent an RU greater than the secondary link station can accept
32		Connect-parameter-error counter: the Connect was rejected because it specified an odd-number buffer length, or it specified a buffer size insufficient to hold the link header, TH, RH, and at least a 64-byte RU
33(=n)		Read-Start-Old-received counter: the secondary link station received a Read Start Old command *Note:* All counters are in binary.
15−n		Data for Counter Set 5 (for X.25 physical circuit) (retired: supported only for PUs not at the current level of SNA) *Note:* Sent only from the primary end of an X.25 physical circuit.
15−17		Communication adapter counter validity mask
15		Mask byte 1 (bit is set to 1 if the counter is valid):
	0	Number of I-frames transmitted counter
	1	Number of I-frames received counter
	2	Number of RR frames transmitted counter
	3	Number of RR frames received counter
	4	Number of RNR frames transmitted counter
	5	Number of RNR frames received counter
	6	Number of REJ frames transmitted counter
	7	Number of REJ frames received counter
16		Mask byte 2 (bit is set to 1 if the counter is valid):
	0	Number of retransmissions counter
	1	Number of frames received with FCS errors counter
	2	Number of errors on receive side counter
	3	Number of overruns on receive side counter
	4	Number of underruns on transmit side counter
	5−7	Reserved
17		Reserved

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
18–19		Number of I-frames transmitted
20–21		Number of I-frames received
22–23		Number of RR frames transmitted
24–25		Number of RR frames received
26–27		Number of RNR frames transmitted
28–29		Number of RNR frames received
30–31		Number of REJ frames transmitted
32–33		Number of REJ frames received
34–35		Number of retransmissions
36–37		Number of frames received with FCS errors
38–39		Number of errors on receive side
40–41		Number of overruns on receive side
42–43(=n)		Number of underruns on transmit side *Note:* All counters are in binary.
15–n		Data for Counter Set 6 (for X.25 virtual circuit) (retired: supported only for PUs not at the current level of SNA) *Note:* Sent only from the primary end of an X.25 virtual circuit.
15–17		Communication adapter counter validity mask
15		Mask byte 1 (bit is set to 1 if the counter is valid):
	0	Number of data packets transmitted counter
	1	Number of data packets received counter
	2	Number of RR packets transmitted counter
	3	Number of RR packets received counter
	4	Number of RNR packets transmitted counter
	5	Number of RNR packets received counter
	6	Number of interrupt packets transmitted counter
	7	Number of interrupt packets received counter
16		Mask byte 2 (bit is set to 1 if the counter is valid):
	0	Number of connection requests counter
	1	Number of connections counter
	2	Number of reset indications counter
	3	Number of clear indications counter
	4	Number of data packets with D-bit transmitted counter
	5	Number of data packets with D-bit received counter
	6–7	Reserved
17		Reserved
18–19		Number of I packets transmitted

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
20 – 21		Number of I packets received
22 – 23		Number of RR packets transmitted
24 – 25		Number of RR packets received
26 – 27		Number of RNR packets transmitted
28 – 29		Number of RNR packets received
30 – 31		Number of interrupt packets transmitted
32 – 33		Number of interrupt packets received
34 – 35		Total number of connection requests (call request and incoming calls)
36 – 37		Total number of connections (calls connected and accepted)
38 – 39		Number of reset indications
40 – 41		Number of clear indications
42 – 43		Number of data packets with D-bit transmitted
44 – 45(=n)		Number of data packets with D-bit received *Note:* All counters are in binary.
7 – n		PU/LU Dependent Data
7	0 1 2 – 7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000100; the CNM target ID identifies a PU|LU
8 – 13	 0 – 11 12 – 31	Node identification Block number ID number
12 – 13		Reserved
14 – n		PU/LU dependent data
7 – n		Engineering Change Levels
7	0 1 2 – 7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000101; the CNM target ID identifies a PU
8 – 13	 0 – 11 12 – 31	Node identification Block number ID number
12 – 13		Reserved
14 – n		Implementation defined data describing hardware, microcode, and programming levels
7 – n		Link Connection Subsystem Data (retired: supported only for PUs not at the current level of SNA)

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
7	0	Solicitation indicator (see above)
	1	Not-last request indicator (see above)
	2–7	Type code: 000110; the CNM target ID identifies an adjacent link station in the origin subarea
8–13		Node identification:
	0–11	Block number
	12–31	ID number
12–13		Reserved
14		Data selection, echoed from the soliciting REQMS command: X'02' link status command sequence X'03' remote DTE interface status X'04' remote modem self test
15		Link connection subsystem type: X'01' link type 1 (links that use 3863, 3864, or 3865 modems; also links that use 5865, 5866, or 5868 modems running LPDA-1) X'02' link type 2 (3867 link diagnostic unit)
16–17		Validity indicators, bits 0–9 (how the PU sending this RU views the data): *Note:* The values to follow are used in each of the validity indicator fields. 00 data valid, from the modem 01 data invalid, no response from the modem 10 data invalid, response in error from the modem 11 data invalid, execution not attempted by the PU sending this RU
	0–1	Remote modem status
	2–3	Local modem status
	4–5	Modem self test *Note:* If byte 14 = X'02', bits 4–5 are for local modem self-test. If byte 14 = X'04', bits 4–5 are for remote modem self-test.
	6–7	Reserved
	8–9	Remote DTE interface status
	10–13	Reserved
	14–15	Link connection subsystem data format indicator: 00 format 0 01 format 1: same as format 0, plus; remote modem self test results, channelization status, local and remote modem status extensions, and general status extensions
18–19		Remote modem status:
	0–5	Hit count (noise spikes) for link type 1, reserved for link type 2 *Note:* For bits 6–7 and 12–14, when the condition exists, the bit value will be 1.
	6	Modem reinitialization was performed
	7	Loss of receive line signal
	8–11	Quadratic error value for link type 1, number of byte errors during test for link type 2
	12	Remote DTE power off detected
	13	Data Terminal Ready loss detected

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
	14	Switched-Network-Back-Up connected
	15	DTE streaming condition detected
20 – 21		Local modem status:
	0 – 5	Hit count (noise spike) for link type 1, reserved for link type 2 *Note:* For bits 6 – 7 and 12 – 14, when the condition exists, the bit value will be 1.
	6	Modem reinitialization was performed
	7	Loss of receive line signal
	8 – 11	Quadratic error value for link type 1, number of byte errors during test for link type 2
	12	Remote modem power loss detected
	13	Speed,for link type 1 (always full for link type 2): 0 half 1 full
	14	Switched-Network-Back-Up connected
	15	Reserved
22 – 24		Local Modem Self-Test and Remote-Tone Results, Remote Modem Self-Test Results: *Note:* If byte 14 is X'02', link status command sequence, then bytes 22 – 24 pertain to the local modem. If byte 14 is X'04', remote modem self-test, then bytes 22 – 24 pertain to the remote modem.
	0 – 2	Model bits, concatenated to the right of the bit-string formed by bits 18, 19, 8, and 15 (in this order) represents the modem model returned as modem self-test result in the bit-string formed by bits 2 and 3 of byte 3, bits 0 and 7 of byte 2, and bits 0, 1, and 2 of byte 1 (in this order), see "LPDA-1 Results Message Information Fields" in *IBM 5865/5866 Modem Models 2, 3 Maintenance Information and Parts Catalog*, SY33-2048.
	3	Link connection type: 0 nonswitched 1 switched
	4	Configuration: 0 point to point 1 multipoint
	5	Modem role: 0 primary or control modem 1 secondary or tributary modem
	6	Clear To Send delay for link type 1 (reserved for link type 2): 0 normal 1 exceptional
	7	Received line signal detector sensitivity for link type 1 (reserved for link type 2): 0 normal 1 limited
	8	Model bit, see bits 0 – 2 specification
	9	Modem self-test result: 0 passed 1 failed

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
	10	Remote tone test result for local modem self test (reserved for remote modem self test): 0 passed 1 failed *Note:* For the following bits, when the condition exists, the bit value will be 1.
	11	Feature card suspected in error
	12	Receiver card suspected in error for link type 1 (reserved for link type 2)
	13	Receiver card extension suspected in error for link type 1 (reserved for link type 2)
	14	Front end card is suspected in error for link type 1 (reserved for type 2)
	15	Model bit, see bits 0 – 2 specification
	16	Feature card installed (tone alarm card installed if nonswitched link connection; integral protection coupler installed if switched link connection)
	17	Switched-Network-Back-Up installed
	18	Model bit, see bits 0 – 2 specification; also if its value is 1 then channelization feature installed
	19	Model bit, see bits 0 – 2 specification; also if its value is 1 then fan-out feature installed
	20 – 23	Microcode EC level
25 – 26		Remote DTE Interface Status
25		Current state of the RS-232C or V.24 interface leads (for bits 0 – 5 and 7, when the condition exists, the bit value is set to 1):
	0	Request To Send
	1	Clear To Send
	2	Reserved
	3	Transmit Data
	4	Reserved
	5	Data Terminal Ready
	6	Speed: 0 half 1 full
	7	DTE power loss
26		Indication of transition of RS-232C or V.24 leads since last test occurrence (for the following bits, when the condition exists, the bit value is set to 1):
	0	Request To Send changed at least once
	1	Clear To Send changed at least once
	2	Received Data changed state
	3	Transmit Data changed state
	4	Received Line Signal loss was detected at least once
	5	Data Terminal Ready dropped at least once
	6	Modem speed was changed at least once
	7	DTE power loss was detected at least once

End of format 0, Format 1 continues below.

27 – 29		Channelization status

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
27		Channelization and tailing flags (for the following bits, when the condition exists, the bit value is set to 1):
	0	This data is associated with a channelized modem
	1	This data is associated with a tailed link of a channelized modem
	2	This data is associated with channel A of a channelized modem
	3–7	Reserved
28–29		Channelization correlation number: a user assigned value used to correlate link connections with a channelized modem. The same value may be assigned to each of the link connections of a channelized modem so that those link connections can be associated with that particular modem
30–37		Local Modem Status Extension
30		Local modem receive dB level (with all code points representing dB units):
		X'00' function not supported
		X'01'–X'40' ignore data
		X'41' not available
		X'42'–X'4B' < -48 dB
		X'4C' -48 dB
		X'4D' -47 dB
		X'4E'–X'60' -46 dB to -28 dB
		X'61' -27 dB
		X'62'–X'6B' -26 dB to -17 dB
		X'6C' -16 dB
		X'6D'–X'75' -15 dB to - 7 dB
		X'76' - 6 dB
		X'77'–X'7D' - 5 dB to + 1 dB
		X'7E' + 2 dB
		X'7F' > + 2 dB
		X'80'–X'FF' ignore data
31–37		Reserved
38–45		Remote Modem Status Extension
38		Remote modem receive dB level (with all code points representing dB units):
		X'00' function not supported
		X'01'–X'40' ignore data
		X'41' not available
		X'42'–X'4B' < -48 dB
		X'4C' -48 dB
		X'4D' -47 dB
		X'4E'–X'60' -46 dB to -28 dB
		X'61' -27 dB
		X'62'–X'6B' -26 dB to -17 dB
		X'6C' -16 dB
		X'6D'–X'75' -15 dB to - 7 dB
		X'76' - 6 dB

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
		X'77'−X'7D' - 5 dB to + 1 dB X'7E' + 2 dB X'7F' > + 2 dB X'80'−X'FF' ignore data
31−37		Reserved
46−53(=n)		General status extension
46		Link-level address used to address the remote modem
47		Remote DTE Interface Extension
48−53(=n)		Reserved
CNM Vectors (described 0-origin)		
(Retired: CNM vectors are supported only for PUs not at the current level of SNA)		
0		Vector length: a binary count of the length in bytes of this vector (bytes 1−n)
1		Type field:
	0−1	Reserved
	2−7	Vector type: an identifier of the information contained in bytes 2−n.
2−n		Vector data
0−n		Embedded Text Vector
0		Vector length: a binary count of the length in bytes of this vector (bytes 1−n)
1		Type field:
	0−1	Reserved
	2−7	Vector type: 000000 the vector contains a text message, composed of SCS characters (only value defined)
2−n		Vector data in SCS text
0−n		Embedded Name List Vector
0		Vector length: a binary count of the length in bytes of this vector (bytes 1−n)
1		Type field:
	0−1	Reserved
	2−7	Vector type: 001100
2		Hierarchy name options: X'01' reserved X'02' only value defined X'03' retired
3		Number of name entries to follow

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
4−n		Hierarchy name list: identifies network elements for which there is no name known to the controlling SSCP; examples of such elements are disk drive, display head; the hierarchy name list can contain up to five entries in hierarchy sequence; first is nearest to the PU; each entry has the following format:
0		Binary count of the length in bytes of the name
1−m		Name in EBCDIC (any SCS character string)
m+1−m+4		Resource type: if byte m+1 is not equal to X'00', no translation is required and the resource type is the EBCDIC value of the four bytes (e.g., "loop," "disk," or "adap"); if byte m+1=X'00' and byte m+2=X'00', bytes m+3 and m+4 are assumed to contain an encoded value that can be translated into resource type; if byte m+1=X'00' and byte m+2=X'01', bytes m+3 and m+4 are qualifiers of the Alert originator block number, creating a unique type code by product
0−n		User Action Qualifier
0		Vector length: a binary count of the length in bytes of this vector (bytes 1−n)
1		Type field:
	0−1	Reserved
	2−7	Vector type: 001101
2−n		User action qualifier: a product-defined value represented in SCS characters that is to distinguish, for example, among multiple instances of an element (e.g., reporting which scanner of several has failed)

RELQ (RELEASE QUIESCE)

LU → LU, Exp; DFC

RELQ is used to release a half-session from a quiesced state. This RU is not used for LU 6.2

RELQ (RELEASE QUIESCE)

Byte	Bit	Content
0		X'82' request code

REQDISCONT (REQUEST DISCONTACT)

PU T1|2 → SSCP, Norm; FMD NS(c)

With REQDISCONT, the PU T1|2 requests the SSCP to start a procedure that will ultimately discontact the secondary station in the T1|2 node.

REQDISCONT (REQUEST DISCONTACT)

Byte	Bit	Content
0–2		X'01021B' NS header
3	0–3	Type: X'0' normal X'8' immediate
	4–7	CONTACT information: X'0' do not send CONTACT immediately X'1' send CONTACT immediately *Note:* Bits 4–7 are reserved for switched connections.

REQMS (REQUEST MAINTENANCE STATISTICS)

SSCP → PU, Norm; FMD NS(ma)

REQMS has been retired from SNA for T2 nodes.

REQMS requests the management services associated with the PU to provide maintenance statistics for the resource indicated by the CNM target ID in the CNM header.

Consult product documentation for further information on product support.

REQMS (REQUEST MAINTENANCE STATISTICS)

Byte	Bit	Content
0–2		X'410304' NS header
3–4		CNM target ID, as specified in bytes 5–6, bits 2–3

REQMS (REQUEST MAINTENANCE STATISTICS)

Byte	Bit	Content
5–6	0–1	Reserved
	2–3	CNM target ID descriptor: 00 byte 4 contains a local address for a PU or LU in a T2 node or an LSID for a PU or LU in a T1 node; byte 3 is reserved 01 bytes 3–4 contain the element address of a link, adjacent link station, PU, or LU in the destination subarea, if ENA is supported; otherwise, its network address
	4–15	Procedure related identifier (PRID): a CNM application program generated value for CNM application program correlation, or an SSCP generated value for SSCP routing
7		Request-Specific Information
	0	Reset indicator (or reserved, as shown below for each Type code): 0 do not reset data when RECFMS is sent in reply 1 reset data when RECFMS is sent in reply
	1	Reserved
	2–7	Request-specific type code (see below)
Note:		For reply (i.e., solicited) requests, bytes 3–6 and byte 7, bits 2–7, echo the corresponding fields in the CNM header received in the request that solicited the reply requests.
7		SDLC Test Command/Response Statistics
	0	Reset indicator
	1	Reserved
	2–7	Type code: 000001; the CNM target ID identifies a PU T1|2
7		Summary Error Data
	0	Reset indicator
	1	Reserved
	2–7	Type code: 000010; the CNM target ID identifies a PU
7		Communication Adapter Data
	0	Reset indicator
	1	Reserved
	2–7	Type code: 000011; the CNM target ID identifies a PU T1|2
7–n		PU- or LU-Dependent Data
7	0	Reset indicator
	1	Reserved
	2–7	Type code: 000100; the CNM target ID identifies a PU|LU
8–n		PU- or LU-dependent request parameters: implementation-dependent information (See CNM application product specifications for details).
7		Engineering Change Levels
	0–1	Reserved
	2–7	Type code: 000101; the CNM target ID identifies a PU
7–8		Link Connection Subsystem Data (retired: supported only for PUs not at the current level of SNA)

REQMS (REQUEST MAINTENANCE STATISTICS)

Byte	Bit	Content
7	0	Reset indicator
	1	Reserved
	2–7	Type code: 000110; the CNM target ID identifies an adjacent link station in the destination subarea
8		Data selection requested: X'02' link status command sequence X'03' remote DTE interface status X'04' remote modem self test

RQR (REQUEST RECOVERY)

SLU → PLU, Exp; SC

RQR is sent by the secondary to request the primary to initiate recovery for the session by sending CLEAR or to deactivate the session. This RU is not used for LU 6.2.

RQR (REQUEST RECOVERY)

Byte	Bit	Content
0		X'A3' request code

RSHUTD (REQUEST SHUTDOWN)

SLU → PLU, Exp; DFC

RSHUTD is sent from the secondary to the primary to indicate that the secondary is ready to have the session deactivated. RSHUTD does *not* request a shutdown; therefore, SHUTD is not a proper reply; RSHUTD requests an UNBIND. This RU is not used for LU 6.2.

RSHUTD (REQUEST SHUTDOWN)

Byte	Bit	Content
0		X'C2' request code

RTR (READY TO RECEIVE)

LU → LU, Norm; DFC

RTR indicates to the bidder that it is now allowed to initiate a bracket. RTR is sent only by the first speaker.

RTR (READY TO RECEIVE)

Byte	Bit	Content
0		X'05' request code

SBI (STOP BRACKET INITIATION)

LU → LU, Exp; DFC

SBI is sent by either half-session to request that the receiving half-session stop initiating brackets by continued sending of BB and the BID request. This RU is not used for LU 6.2.

SBI (STOP BRACKET INITIATION)

Byte	Bit	Content
0		X'71' request code

SDT (START DATA TRAFFIC)

PLU → SLU, SSCP → PU|SSCP, Exp; SC

SDT is sent by the primary session control to the secondary session control to enable the sending and receiving of FMD and DFC requests and responses by both half-sessions. This RU is not used for LU 6.2.

SDT (START DATA TRAFFIC)

Byte	Bit	Content
0		X'A0' request code

SHUTC (SHUTDOWN COMPLETE)

SLU → PLU, Exp; DFC

SHUTC is sent by a secondary to indicate that it is in the shutdown (quiesced) state. This RU is not used for LU 6.2.

SHUTC (SHUTDOWN COMPLETE)

Byte	Bit	Content
0		X'C1' request code

SHUTD (SHUTDOWN)

PLU → SLU, Exp; DFC

SHUTD is sent by the primary to request that the secondary shut down (quiesce) as soon as convenient. This RU is not used for LU 6.2.

SHUTD (SHUTDOWN)

Byte	Bit	Content
0		X'C0' request code

SIG (SIGNAL)

LU → LU, Exp; DFC

SIG is an expedited request that can be sent between half-sessions, regardless of the status of the normal flows. It carries a four-byte value, of which the first two bytes are the signal code and the last two bytes are the signal extension value.

SIG (SIGNAL)

Byte	Bit	Content
0		X'C9' request code

SIG (SIGNAL)

Byte	Bit	Content
1–2		Signal code: X'0000' no-op (no system-defined code) X'0001' request to send (only value defined for LU 6.2) X'0002' assistance requested X'0003' intervention required (no data loss)
3–4		Signal extension: set by the sending end user or NAU services manager, or set to X'0001' for LU 6.2 by data flow control

STSN (SET AND TEST SEQUENCE NUMBERS)

PLU → SLU, Exp; SC

STSN is sent by the primary half-session sync point manager to resynchronize the values of the half-session sequence numbers, for one or both of the normal flows at both ends of the session. This RU is not used for LU 6.2.

STSN (SET AND TEST SEQUENCE NUMBERS)

Byte	Bit	Content
0		X'A2' request code
1	0–1	Action code for S → P flow (related data in bytes 2–3)
	2–3	Action code for P → S flow (related data in bytes 4–5) *Note:* Each action code is set and processed independently. Values for either action code are: 00 ignore; this flow not affected by this STSN 01 set; the half-session value is set to the value in bytes 2–3 or 4–5, as appropriate 10 sense; secondary half-session's sync point manager returns the transaction processing program's sequence number for this flow in the response RU 11 set and test; the half-session value is set to the value in appropriate bytes 2–3 or 4–5, and the secondary half-session's sync point manager compares that value against the transaction processing program's number and responds accordingly
	4–7	Reserved
2–3		Secondary-to-primary sequence number data to support S → P action code
4–5		Primary-to-secondary sequence number data to support P → S action code

STSN (SET AND TEST SEQUENCE NUMBERS)

Byte	Bit	Content
Note:		For action codes 01 and 11, the appropriate bytes 2–3 or 4–5 contain the value to which the half-session value is set and against which the secondary half-session's sync point manager tests the transaction processing program's value for the respective flow. For action codes 00 and 10, the appropriate bytes 2–3 or 4–5 are reserved.

TERM-SELF Format 0 (TERMINATE-SELF)

TLU → SSCP, Norm; FMD NS(s)

TERM-SELF from the TLU requests that the SSCP assist in the termination of one or more sessions between the sender of the request (TLU = OLU) and the DLU. This RU is not used for LU 6.2; refer to TERM-SELF Format 1.

TERM-SELF Format 0 (TERMINATE-SELF)

Byte	Bit	Content
0–2		X'010683' NS header
3		Type:
	0–1	00 the request applies to active and pending-active sessions 01 the request applies to active, pending-active, and queued sessions 10 the request applies to queued only sessions 11 reserved
	2	Reserved if byte 3, bit 4 = 1; otherwise: 0 forced termination—session to be deactivated immediately and unconditionally 1 orderly termination—permitting an end-of-session procedure to be executed at the PLU before the session is deactivated
	3	0 do not send DACTLU to OLU; another session initiation request will be sent for OLU 1 send DACTLU to OLU when appropriate; no further session initiation request will be sent (from this sender) for OLU
	4	0 orderly or forced (see byte 3, bit 2) 1 clean up
	5–6	00 select session(s) for which DLU is PLU 01 select session(s) for which DLU is SLU 10 select session(s) regardless of whether DLU is SLU or PLU 11 reserved
	7	0 indicates that the format of the RU is Format 0 and that byte 3 is the Type byte.
4–5		Uninterpreted Name of DLU (retired):
4		Type: X'F3' logical unit

TERM-SELF Format 0 (TERMINATE-SELF)

Byte	Bit	Content
5		Length: X'00' only value allowed, and always present *Note:* Because the length value of the DLU name is 0, the TERM-SELF applies to all sessions, as specified in the Type byte, where the TLU is a partner.
Note:		The following defaults are supplied by the SSCP receiving a Format 0 TERM-SELF: • Reason: network user, normal • Notify: do not notify • URC is not used in mapping to subsequent requests.

TERM-SELF Format 1 (TERMINATE-SELF)

TLU → SSCP, Norm; FMD NS(s)

TERM-SELF from the TLU requests that the SSCP assist in the termination of one or more sessions between the sender of the request (TLU = OLU) and the DLU.

TERM-SELF Format 1 (TERMINATE-SELF)

Byte	Bit	Content
0–2		X'810683' NS header
3	0–3	Format: 0001 Format 1 (only value defined)
	4–6	Reserved
	7	1 indicates that byte 3, bits 0–3, contain the format value
4		Type:
	0–1	00 the request applies to active and pending-active sessions 01 the request applies to active, pending-active, and queued sessions (only value defined for LU 6.2) 10 the request applies to queued sessions only 11 reserved
	2	Reserved if byte 4, bit 7 = 1; otherwise: 0 forced termination—session to be deactivated immediately and unconditionally 1 orderly termination—permitting an end-of-session procedure to be executed at the PLU before the session is deactivated
	3	0 do not send DACTLU to OLU; another session initiation request will be sent for OLU 1 send DACTLU to OLU when appropriate; no further session initiation request will be sent (from this sender) for OLU (only value defined for LU 6.2)
	4	Reserved
	5–6	00 select session(s) for which DLU is PLU 01 select session(s) for which DLU is SLU

TERM-SELF Format 1 (TERMINATE-SELF)

Byte	Bit	Content
		10 select session(s) regardless of whether DLU is SLU or PLU
		11 reserved
	7	0 orderly or forced (see byte 4, bit 2)
		1 clean up
5		Reason:
	0	0 network user
		1 network manager
	1	0 normal termination
		1 abnormal termination
	2–7	Reserved
6		NOTIFY specifications (reserved for LU 6.2):
	0–5	Reserved
	6	0 do not notify TLU when the session takedown procedure is complete
		1 notify the TLU when the session takedown procedure is complete
	7	Reserved
7		Reserved
8–n		Session key, as described in the section "Session Keys" on page 8-27
		X'0A' URC
		Note: This URC is the one carried in the INIT issued previously by the same LU (i.e., ILU = TLU), and differs from the one in bytes n+4 through p.
n+1–n+2		Retired
n+3–p		User Request Correlation (URC) Field
n+3		Length, in binary, of URC field
		Note: X'00' = no URC.
n+4–p		URC: LU-defined identifier; this value can be returned by the SSCP in a subsequent NOTIFY to correlate the NOTIFY to this terminating request

UNBIND (UNBIND SESSION)
LU → LU, Exp; SC

UNBIND is sent to deactivate an active session between the two LUs.

UNBIND (UNBIND SESSION)

Byte	Bit	Content
0		X'32' request code

UNBIND (UNBIND SESSION)

Byte	Bit	Content
1		UNBIND type (for UNBIND types X'00' through X'06' and X'80' through X'FF', the session is ended when the response is received; for UNBIND types X'07' through X'7F', the session is ended immediately): X'01' normal end of session X'02' BIND forthcoming; retain the node resources allocated to this session, if possible X'06' invalid session parameters: the BIND negotiation has failed because the primary half-session cannot support parameters specified by the secondary X'07' virtual route inoperative: the virtual route used by the LU-LU session has become inoperative, thus forcing the deactivation of the identified LU-LU session X'08' route extension inoperative: the route extension used by the LU-LU session has become inoperative, thus forcing the deactivation of the identified LU-LU session X'09' hierarchical reset: the identified LU-LU session is being deactivated because of a +RSP((ACTPU | ACTLU), Cold) X'0A' SSCP gone: the identified LU-LU session had to be deactivated because of a forced deactivation of the SSCP-PU or SSCP-LU session (e.g., DACTPU, DACTLU, or DISCONTACT was received) X'0B' virtual route deactivated: the identified LU-LU session had to be deactivated because of a forced deactivation of the virtual route being used by the LU-LU session X'0C' LU failure—unrecoverable: the identified LU-LU session had to be deactivated because of an abnormal termination of the PLU or SLU; recovery from the failure was not possible X'0E' LU failure—recoverable: the identified LU-LU session had to be deactivated because of an abnormal termination of one of the LUs of the session; recovery from the failure may be possible X'0F' cleanup: the node sending UNBIND is resetting its half-session before receiving the response from the partner node X'11' gateway node cleanup: a gateway node is cleaning up the session because a gateway SSCP has directed the gateway node (via NOTIFY) to deactivate the session (e.g., a session setup error or session takedown failure has occurred) X'FE' session failure: the session has failed for a reason specified by the associated sense data

For session stages that were established with extended BIND, bytes 2–n are included; otherwise, bytes 6–n are omitted and bytes 2–5 are included only for Type = X'FE'.

2–5	Sense data: same value as generated at the time the error was originally detected (e.g., for a negative response, receive check, or EXR) *Note:* For Type=X'FE' the Sense Data field in bytes 2–5 of the UNBIND RU is the same as that in bytes 2–5 of the Extended Sense Data control vector; otherwise, this field (bytes 2–5 of the UNBIND RU) is reserved.
6–n	Control vectors, as described in the section "Control Vectors" on page 8-3.

UNBIND (UNBIND SESSION)

Byte	Bit	Content
		Note: The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). X'35' Extended Sense Data control vector (present if and only if the Fully Qualified PCID [X'60'] control vector is also present) X'60' Fully Qualified PCID control vector (present on session stages that were established with extended BIND)

Note: An UNBIND is sent instead of a -RSP(BIND) as a reply to BIND (to reject the BIND) only if the BIND is extended and no errors limit recognition of the BIND as extended.

Introduction to Response Units

Apart from the exceptions cited below, response units return the number of bytes specified in the following table; only enough of the request unit is returned to include the field-formatted request code or NS header.

RU Category of Response	Number of Bytes
DFC	1
SC	1
NC	1
FMD NS (FI = 1) (field-formatted)	3
FMD NS (FI = 0) (character-coded)	0
FMD (LU-LU)	0

All negative responses return four bytes of sense data in the RU, followed by either:

1. The number of bytes specified in the table above, or
2. Three bytes (or the entire request unit, if shorter than three bytes).

The second option applies where a sensitivity to SSCP-based sessions versus LU-LU sessions does not necessarily exist and can be chosen for implementation simplicity. Refer to Chapter 10, "Sense Data" on page 10-1 for sense data values and their corresponding meanings.

Some positive response units return the request code or NS header followed by additional data. "Positive Response Units with Extended Formats" on page 5-61 contains detailed formats of these response units, arranged in alphabetical order. Each format description begins with the following heading:

"RSP(ABBREVIATED RU NAME); Origin-NAU → Destination-NAU, Normal (Norm) or Expedited (Exp) Flow; RU Category"

Positive Response Units with Extended Formats

RSP(ACTLU)
LU → SSCP, Exp; SC

RSP(ACTLU)

Byte	Bit	Content
0		X'0D' request code
1		Type of activation selected: X'01' cold (retired) X'02' ERP
2	0–3 4–7	FM profile: Same as the corresponding request TS profile: same as the corresponding request
Note:		Two versions of this RU are defined.

A full response can be sent in which bytes 0–m are present.

3–m		Control vectors as described in the section "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). When present, they appear in the order specified. X'00' SSCP-LU Session Capabilities control vector (present to override the defaults of a 2-byte response, in which case always first) X'0C' LU-LU Session Services Capabilities control vector (present to override the defaults of a 2-byte response, in which case always second)

A two-byte response may be received; it means maximum RU size = 256 bytes, LU-LU session limit = 1, the LU can act as a secondary LU, and all other fields in control vectors X'00' and X'0C' are defaulted to 0's.

RSP(ACTPU)
PU → SSCP, Exp; SC

RSP(ACTPU)

Byte	Bit	Content
0		X'11' request code

RSP(ACTPU)

Byte	Bit	Content
1	0–1	Reserved
	2–3	Format of response: 00 format 0 01 format 1
	4–7	Type activation selected: X'1' cold, IPL not required X'2' ERP
2–9		Contents ID: 8-character EBCDIC symbolic name of the load module currently operating in the node; eight space (X'40') characters is the default value
Note:		End of Format 0 ; Format 1 continues below.
10–n		Format 1 Continues
10–11		Reserved
12–n		Control vectors as described in the section "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). X'07' PU FMD-RU-Usage control vector

RSP(BIND)
SLU → PLU, Exp; SC

A +RSP(BIND) carries the session parameters as indicated by the SLU or by intermediate nodes along the session path.

- A short (1-byte) response may be sent for a nonextended nonnegotiable BIND request that specifies no session-level cryptography.
- A cryptography response (bytes 0–k) may be sent for a nonextended nonnegotiable BIND request that specifies session-level cryptography.
- A nonextended negotiable response (bytes 0–r) may be sent for an extended or nonextended negotiable BIND request.
- An extended response (bytes 0–s) may be sent for an extended (negotiable or nonnegotiable) BIND request. Intermediate nodes along the session path may extend short, cryptography, and negotiable responses.

RSP(BIND)

Byte	Bit	Content
0		X'31' request code

RSP(BIND)

Byte	Bit	Content
1	0–3	Format: 0000 (only value defined)
	4–7	Type: 0000 negotiable (only value defined for LU 6.2) 0001 nonnegotiable
2–24		Bytes 2–24 of the BIND request: for an extended or negotiable response, the negotiated values may differ; for a cryptography response, the values are the same as those received in the BIND request
25	0	Reserved
	1	Negotiated or echoed from the BIND as described above for bytes 2-24.
	2–5	Reserved
	6–7	Usage varies by LU type
	•	*For LU types 4 and 7:*
	6	Reserved
	7	System microcode update reply: 0 Update refused 1 Update acceptable *Note:* Actual downloading of the update is contingent on the outcome of additional implementation-specific status exchanges.
	•	*For all other LU types:*
	6–7	Length-checked compression response — an LU that is not compression-capable will return a short RSP(BIND), echo these bits as received, or set these bits to 0's. In any of these three events, the offered or mandated compression is refused (see "Length-Checked Compression" for details of the compression header): 00 no compression — returned by a compression-capable LU that is refusing the offered or mandated compression, or by an LU that is not compression-capable, or by any LU when the PLU has indicated that compression will not be used on this session 01 no compression — returned only by an LU that is not compression-capable and is echoing the BIND 10 offered or mandated compression accepted — returned only by a compression-capable LU to indicate acceptance of compression on the session 11 no compression — returned only by an LU that is not compression-capable and is echoing the BIND
26–k		Cryptography Options (see Note 3) for a nonnegotiable response, same value returned as received for a nonnegotiable response or an LU 6.2 response
	4–7	Session-level cryptography options field length: same value (Bytes 27–k are omitted if this length field is omitted or set to 0.)
27	0–1	Session cryptography key encipherment method: same value returned as received in the request, if present
	2–4	Reserved
	5–7	Cryptography cipher method: same value returned as received
28–k		An 8-byte implementation-chosen, nonzero, pseudo random session-seed cryptography value enciphered under the session cryptography key, if session-level cryptography is specified; otherwise, omitted

RSP(BIND)

Byte	Bit	Content
k+1(=m)		Retired: set to 0 by implementations at the current level of SNA
m+1		Length of user data
m+2−n		User data: for an extended or negotiable response, the user data may differ from that received on the BIND request
n+1		Length of URC
n+2−p		URC as received on the BIND request
p+1(=r)		Retired: set to 0 by implementations at the current level of SNA
r+1−s		Control vectors, as described in "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). X'0E' Network Name control vector: CP network name (present in extended BIND responses when neither the Fully Qualified PCID [X'60'] control vector nor the Route Selection [X'2B'] control vector contains the CP(PLU) name) X'2B' Route Selection control vector (optionally present if received on BIND) X'60' Fully Qualified PCID control vector (present if received on BIND) X'66' Length-Checked Compression control vector (present when the SLU received a X'66' control vector on the BIND and also requests compression)
Note 1:		On a response, if the last byte of a response without control vectors (byte 7, bit 6 = 0) is a length field and that field is 0, that byte may be dropped from the response. This applies also to byte 26 (where the count occupies only bits 4−7) if bits 0−3 are also 0 — the entire byte may be dropped if no bytes follow.
Note 2:		In negotiable or extended BIND responses, reserved fields in the BIND are set by the SLU to binary 0's in the RSP(BIND); any fields at the end of the BIND that are not recognized by the SLU are discarded and not returned in the RSP(BIND).
Note 3:		The first byte of the Cryptography Options field (byte 26) is returned on the response for a nonextended nonnegotiable BIND only when session-level cryptography was specified in the BIND. Byte 26 is always present in any extended response. It is also present in any nonextended negotiable response if not truncated as allowed in Note 1. In all cases, however, the remaining bytes of the Cryptography Options field (bytes 27−k) are present only if session-level cryptography was specified in the BIND.
Note 4:		On a response, when the adaptive session-level pacing support bit (byte 9, bit 0) is set to 1 (adaptive session pacing supported), the window sizes (byte 8, bits 2−7; byte 9, bits 2−7; byte 12, bits 2−7 and byte 13, bits 2−7) are all set to 0.
Note 5:		An extended short response to a nonnegotiable BIND is of the following form:
0		X'31' request code
1	0−3 4−7	Format: 0000 (only value defined) 0001 nonnegotiable

RSP(BIND)

Byte	Bit	Content
2–5		Reserved
6	0	Whole-BIUs required indicator (reserved in nonextended non-LU6.2 BIND responses): 0 the sending node (SLU-side of the session stage) supports receipt of segments on this session 1 the sending node (SLU-side of the session stage) does not support receipt of segments on this session; the maximum sent-RU size specified in bytes 10 and 11 of RSP(BIND) are negotiated so that BIUs on this session are not segmented when sent to a node requiring whole BIUs
	1–7	Reserved
7	0–5	Reserved
	6	Control vectors included indicator: 1 control vectors are present (only value defined)
	7	Reserved
8	0	Secondary-to-primary pacing staging indicator: 0 pacing in the secondary-to-primary direction occurs in one stage (only value defined)
	1–7	Reserved
9	0	Adaptive session-level pacing support: 0 adaptive pacing not supported by the sending node 1 adaptive pacing supported by the sending node
	1–7	Reserved
10		Maximum RU size sent on the normal flow by the secondary side of the session
11		Maximum RU size sent on the normal flow by the primary side of the session
12	0	Primary-to-secondary pacing staging indicator: 1 pacing in the primary-to-secondary direction occurs in one stage (only value defined)
	1–7	Reserved
13–30(=r)		Reserved
r+1–s		Control vectors, as described in the section "Control Vectors" on page 8-3 *Note:* The following control vectors may be used; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" on page 8-1). X'2B' Route Selection control vector (optionally present if received on the BIND) X'60' Fully Qualified PCID control vector (always present) X'66' Length-Checked Compression control vector (present when the SLU received a X'66' control vector on the BIND and also requests compression)

RSP(STSN)
SLU → PLU, Exp; SC

RSP(STSN)

Byte	Bit	Content
0		X'A2' request code
1	0–1	Result code for S → P action code in the request (related data in bytes 2–3)
	2–3	Result code for P → S action code in the request (related data in bytes 4–5) *Note:* Values for either result code are: For set or ignore action code: 01 ignore (other values reserved); appropriate bytes 2–3 or 4–5 reserved For sense action code: 00 for LU type 0: user-defined meaning; for all other LU types: reserved (appropriate bytes 2–3 or 4–5 reserved) 01 reserved 10 secondary half-session's sync point manager does not maintain or cannot return a valid transaction processing program sequence number (appropriate bytes 2–3 or 4–5 reserved) 11 transaction processing program sequence number, as known at the secondary, is returned in bytes 2–3 or 4–5, as appropriate For set and test action code: 00 for LU type 0: user-defined meaning; for all other LU types: invalid sequence numbers have been detected by the secondary (appropriate bytes 2–3 or 4–5 return the secondary transaction processing program sequence number) *Note:* An invalid determination results when the sequence number indicated could not have occurred. For example, the mounting of an incorrect sync point log tape by the operator at one of the LUs would cause this condition. 01 value received in STSN request equals the transaction processing program sequence number value as known at the secondary (appropriate bytes 2–3 or 4–5 return the secondary's value for the transaction processing program sequence number) 10 secondary half-session's sync point manager does not maintain or cannot return a valid transaction processing program sequence number (appropriate bytes 2–3 or 4–5 reserved) 11 value received in STSN request does not equal the transaction processing program sequence number value as known at the secondary (appropriate bytes 2–3 or 4–5 return the secondary's value for the transaction processing program sequence number)
	4–7	Reserved
2–3		Secondary-to-primary normal-flow sequence number data to support S → P result code, or reserved (see Note above)
4–5		Primary-to-secondary normal-flow sequence number data to support P → S result code or reserved (see Note above)

RSP(STSN)

Byte	Bit	Content
Note:		Where the STSN request specified as action codes two "sets," two "ignores," or a combination of "set" and "ignore," the positive response RU optionally may consist of one byte—X'A2' (the STSN request code)—rather than all six bytes.

Chapter 6. Profiles

Introduction

Some of the session protocols (such as for request and response control modes, brackets, and pacing) are selectable at session activation. Specific combinations of these selectable protocol options are known as profiles.

Those profiles that refer to transmission control (TC) options are called transmission services (TS) profiles; those profiles that refer to data flow control (DFC) and function management data services (FMDS) options are called function management (FM) profiles.

The TS and FM profiles to be used in any session are specified at the time of session activation via parameters in the appropriate session activation request and response (see ACTPU, ACTLU, BIND, and their responses in Chapter 5).

Transmission Services (TS) Profiles

This section describes the transmission services (TS) profiles and their use for LU-LU, CP-CP, SSCP-LU, and SSCP-PU sessions. Profile numbers not shown are reserved in these sessions.

Note: If the TS Usage field in BIND specifies a value for a parameter, that value is used unless it conflicts with a value specified by the TS profile. The TS profile overrides the TS Usage field.

Figure 6-1 identifies the different sessions and logical unit (LU) types that use each TS profile.

TS Profile	Session Types	LU Types
1	SSCP-PU(T1|2),[1] SSCP-LU	-
2	LU-LU	0
3	LU-LU	0, 1, 2, 3
4	LU-LU	0, 1, 6.1
7	LU-LU CP-CP	0, 4, 6.2, 7 -

[1] The boundary function serves in place of the PU type 1 (e.g., to process ACTPU).

Figure 6-1. TS Profiles and Their Usage

TS Profile 1

Profile 1 (used on SSCP-PU and SSCP-LU sessions) specifies the following session rules:

- No pacing.
- Identifiers rather than sequence numbers are used on the normal flows (whenever the TH format used includes a sequence number field).
- SDT, CLEAR, RQR, STSN, and CRV are not supported.
- Maximum RU size on the normal flow between an SSCP and a peripheral LU is 256, unless a different value is specified in RSP(ACTLU) in control vector X'00'.
- Maximum RU size on the normal flow for an SSCP sending to a peripheral PU is 256; in the reverse direction it is 512.

No TS Usage field is associated with this profile.

TS Profile 2

Profile 2 (used on LU-LU sessions) specifies the following session rules:

- Primary-to-secondary and secondary-to-primary normal flows are paced.
- Sequence numbers are used on the normal flows (whenever the TH format used includes a sequence number field).
- CLEAR is supported.
- SDT, RQR, STSN, and CRV are not supported.

The TS Usage subfields defining the options for this profile are:

- Pacing window counts
- Maximum RU sizes on the normal flows

TS Profile 3

Profile 3 (used on LU-LU sessions) specifies the following session rules:

- Primary-to-secondary and secondary-to-primary normal flows are paced.
- Sequence numbers are used on the normal flows (whenever the TH format used includes a sequence number field).
- CLEAR and SDT are supported.
- RQR and STSN are not supported.
- CRV is supported when session-level cryptography is selected (via a BIND parameter).

The TS Usage subfields defining the options for this profile are:

- Pacing window counts
- Maximum RU sizes on the normal flows

TS Profile 4

Profile 4 (used on LU-LU sessions) specifies the following session rules:

- Primary-to-secondary and secondary-to-primary normal flows are paced.
- Sequence numbers are used on the normal flows (whenever the TH format used includes a sequence number field).
- SDT, CLEAR, RQR, and STSN are supported.

- CRV is supported when session-level cryptography is selected (via a BIND parameter).

The TS Usage subfields defining the options for this profile are:

- Pacing window counts
- Maximum RU sizes on the normal flows

TS Profile 7

Profile 7 (used on LU-LU and CP-CP sessions) specifies the following session rules:

- Primary-to-secondary and secondary-to-primary normal flows are optionally paced.
- Sequence numbers are used on the normal flows (whenever the TH format used includes a sequence number field).
- SDT, CLEAR, RQR, and STSN are not supported.
- CRV is supported when session-level cryptography is selected (via a BIND parameter).

The TS Usage subfields in BIND defining the options for this profile are:

- Pacing window counts
- Maximum RU sizes on the normal flows

Function Management (FM) Profiles

This section describes the function management (FM) profiles and their use for LU-LU, CP-CP, SSCP-PU, and SSCP-LU sessions. Profile numbers not shown are reserved in these sessions.

Note: If the FM Usage field in BIND specifies a value for a parameter, that value is used unless it conflicts with a value specified by the FM profile. The FM profile overrides the FM Usage field. Figure 6-2 identifies the different sessions and logical unit (LU) types that use each FM profile.

FM Profile	Session Types	LU Types
0	SSCP-PU(T1|2),[1] SSCP-LU	-
2	LU-LU	0
3	LU-LU	0, 1, 2, 3
4	LU-LU	0, 1
6	SSCP-LU	-
7	LU-LU	0, 4, 7
18	LU-LU	0, 6.1
19	LU-LU CP-CP	6.2 -

[1] The boundary function serves in place of the PU type 1 (e.g., to process ACTPU).
[2] For usage of FM profiles 0 and 6 by LU 6.2, see the discussion of ACTLU in *SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2.*

Figure 6-2. FM Profiles and Their Usage

FM Profile 0

Profile 0 (used on SSCP-PU and SSCP-LU sessions) specifies the following session rules:

- Primary and secondary half-sessions use immediate request mode and immediate response mode.
- Only single-RU chains allowed.
- Primary and secondary half-session chains indicate definite response. Half-session chains generated by a boundary function on behalf of the peripheral LU may indicate no-response or definite response.
- No compression.
- Primary half-session sends no DFC RUs.
- No FM headers.
- No brackets.
- No alternate code.
- Normal-flow send/receive mode is full-duplex.

FM Profile 2

Profile 2 (used on LU-LU sessions) specifies the following session rules:

- Secondary LU half-session uses delayed request mode.
- Secondary LU half-session uses immediate response mode.
- Only single-RU chains allowed.
- Secondary LU half-session requests indicate no-response.
- No FMH-1 SCB compression.
- Length-checked compression allowed.
- No DFC RUs.

- No FM headers.
- Secondary LU half-session is first speaker if brackets are used.
- Bracket termination rule 2 is used if brackets are used.
- Primary LU half-session will send EB.
- Secondary LU half-session will not send EB.
- Normal-flow send/receive mode is FDX.
- Primary LU half-session is responsible for recovery.

The FM Usage fields defining the options for Profile 2 are:

- Primary request control mode selection
- Primary chain response protocol (no-response may not be used)
- Brackets usage and reset state
- Alternate code

FM Profile 3

Profile 3 (used on LU-LU sessions) specifies the following session rules:

- Primary LU half-session and secondary LU half-session use immediate response mode.
- Primary LU half-session and secondary LU half-session support the following DFC functions:
  - CANCEL
  - SIGNAL
  - LUSTAT (allowed secondary-to-primary only)
  - CHASE
  - SHUTD
  - SHUTC
  - RSHUTD
  - BID and RTR (allowed only if brackets are used)
- Length-checked compression allowed.

The FM Usage fields defining the options for Profile 3 are:

- Chaining use (primary and secondary)
- Request control mode selection (primary and secondary)
- Chain response protocol (primary and secondary)
- FMH-1 SCB Compression indicator (primary and secondary)
- Send EB indicator (primary and secondary)
- FM header usage
- Brackets usage and reset state
- Bracket termination rule
- Alternate Code Set Allowed indicator
- Normal-flow send/receive mode
- Recovery responsibility
- Contention winner/loser
- Half-duplex flip-flop reset states

FM Profile 4

Profile 4 (used on LU-LU sessions) specifies the following session rules:

- Primary LU half-session and secondary LU half-session use immediate response mode.
- Primary LU half-session and secondary LU half-session support the following DFC functions:
  - CANCEL
  - SIGNAL
  - LUSTAT
  - QEC
  - QC
  - RELQ
  - SHUTD
  - SHUTC
  - RSHUTD
  - CHASE
  - BID and RTR (allowed only if brackets are used)
- Length-checked compression allowed.

The FM Usage fields defining the options for Profile 4 are:

- Chaining use (primary and secondary)
- Request control mode selection (primary and secondary)
- Chain response protocol (primary and secondary)
- FMH-1 SCB Compression indicator (primary and secondary)
- Send EB indicator (primary and secondary)
- FM header usage
- Brackets usage and reset state
- Bracket termination rule
- Alternate Code Set Allowed indicator
- Normal-flow send/receive mode
- Recovery responsibility
- Contention winner/loser
- Half-duplex flip-flop reset states

FM Profile 6

Profile 6 (used on SSCP-LU sessions) specifies the following session rules:

- Only single-RU chains allowed.
- Primary and secondary half-sessions use delayed request mode and delayed response mode.
- Primary and secondary half-session chains may indicate definite response, exception response, or no response.
- Primary half-session sends no DFC RUs.
- No FM headers.
- No compression.
- No brackets.
- No alternate code.
- Normal-flow send/receive mode is full-duplex.

FM Profile 7

Profile 7 (used on LU-LU sessions) specifies the following session rules:

- Primary LU half-session and secondary LU half-session use immediate response mode.
- Primary LU half-session and secondary LU half-session support the following DFC functions:
  - CANCEL
  - SIGNAL
  - LUSTAT
  - RSHUTD
- Length-checked compression is allowed on LU 0 only.

The FM Usage fields defining the options for Profile 7 are:

- Chaining use (primary and secondary)
- Request control mode selection (primary and secondary)
- Chain response protocol (primary and secondary)
- FMH-1 SCB Compression indicator (primary and secondary)
- Send EB indicator (primary and secondary)
- FM header usage
- Brackets usage and reset state
- Bracket termination rule
- Alternate Code Set Allowed indicator
- Normal-flow send/receive mode
- Recovery responsibility
- Contention winner/loser
- Half-duplex flip-flop reset states

FM Profile 18

Profile 18 (used on LU-LU sessions) specifies the following session rules:

- Primary LU half-session and secondary LU half-session use immediate response mode.
- Primary LU half-session and secondary LU half-session support the following DFC functions:
  - CANCEL
  - SIGNAL
  - LUSTAT
  - BIS and SBI (allowed only if brackets are used)
  - CHASE
  - BID and RTR (allowed only if brackets are used)
- Length-checked compression allowed.

The FM Usage fields defining the options for Profile 18 are:

- Chaining use (primary and secondary)
- Request control mode selection (primary and secondary)
- Chain response protocol (primary and secondary)
- FMH-1 SCB Compression indicator (primary and secondary)
- Send EB indicator (primary and secondary)
- FM header usage

- Brackets usage and reset state
- Bracket termination rule
- Alternate Code Set Allowed indicator
- Normal-flow send/receive mode
- Recovery responsibility
- Contention winner/loser
- Half-duplex flip-flop reset states

FM Profile 19

Profile 19 (used on LU-LU and CP-CP sessions) specifies the following session rules:

- Primary LU half-session and secondary LU half-session use immediate request and immediate response mode.
- Multiple RU chains allowed.
- Primary LU half-session and secondary LU half-session chains indicate definite or exception response.
- No FMH-1 SCB compression.
- Length-checked compression allowed.
- Brackets are used.
- FM headers (types 5, 7, and 12 only) are allowed.
- Conditional termination for brackets (specified by CEB) will be used—primary and secondary half-sessions may send CEB.
- Normal-flow send/receive mode is half-duplex flip-flop.
- Half-duplex flip-flop reset state is *send* for the primary LU half-session and *receive* for the secondary LU half-session after RSP(BIND).
- Symmetric responsibility for recovery.
- Contention winner/loser polarity is negotiated at BIND time; the contention winner is the first speaker and the contention loser is the bidder.
- Primary and secondary half-sessions support the following DFC functions:
  - SIGNAL
  - LUSTAT
  - BIS
  - RTR
- Alternate code permitted.
- The following combinations of RQE, RQD, CEB, and CD are allowed on end-chain RUs:
  - RQE*, CD, ¬CEB
  - RQD2, CD, ¬CEB
  - RQD3, CD, ¬CEB
  - RQE1, ¬CD, CEB
  - RQD*, ¬CD, CEB
  - RQD*, ¬CD, ¬CEB

Chapter 7. User Data Structured Subfields

Introduction

The structured subfields of the User Data field are defined as follows (shown with 0-origin indexing of the subfield bytes—see the individual RU description for the actual displacement within the RU). Each subfield starts with a one-byte binary Length field and is identified by a subfield number in the following byte. The length does not include the Length byte itself. When more than one subfield is included, they appear in ascending order by subfield number.

For LU type 6.2, the Structured User Data field of BIND and RSP(BIND) may contain the Unformatted Data, Mode Name, Network-Qualified PLU Network Name, Network-Qualified SLU Network Name, Random Data, Enciphered Data, Session Qualifier, Security Data 1, Security Data 2, Security Data 3, and Session Instance Identifier subfields. Any subfields received in the Structured User Data field of BIND that are not recognized by the SLU are discarded and not returned as part of the Structured User Data field of the RSP(BIND).

Descriptions

Unformatted Data Structured Data Subfield

The Unformatted Data subfield may optionally be sent in BIND, RSP(BIND), or any of the INITIATE RUs. The content is implementation-defined.

Unformatted Data Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Unformatted Data subfield: values 1 to 17 (X'11') are valid for LU 6.2; otherwise, values 1 to 65 (X'41') are valid
1		X'00'
2−n		Unformatted data: a type-G symbol string

Session Qualifier Structured Data Subfield

The Session Qualifier subfield is used for LU 6.1. It may be carried in BIND, RSP(BIND), or any of the INITIATE RUs.

Session Qualifier Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Session Qualifier subfield: values 3 to 19 (X'13') are valid
1		X'01'
2		Length of primary resource qualifier: values 0 to 8 are valid (X'00' means no primary resource qualifier is present)
3−m		Primary resource qualifier
m+1		Length of secondary resource qualifier: values 0 to 8 are valid (X'00' means no secondary resource qualifier is present)
m+2−n		Secondary resource qualifier

Mode Name Structured Data Subfield

The Mode Name subfield is present in both BIND and RSP(BIND) if the PLU knows the mode name being used by the session. If this subfield is omitted, it is equivalent to specifying the SNA-defined default name (see below).

Mode Name Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Mode Name subfield: values 1 to 9 are valid
1		X'02'
2−n		Mode name: A 0- to 8-character type-1134 symbol-string (see Appendix A, "SNA Character Sets and Symbol-String Types" on page A-1), the first character of which is an uppercase letter. The symbol string may be padded with X'40's on the right, but these X'40's (although affecting the Length field value) are not considered part of the mode name. For LU 6.2, certain mode names are architecturally defined. These include SNASVCMG (used for CNOS and management services LU-LU sessions, as well as generally by service transaction programs), CPSVCMG (used for APPN CP-CP sessions), and some that are used for user sessions; these user-session mode names use the prefix X'7B' (indicating SNA-defined) in byte 2 of the User Data Subfields followed by the SNA-defined mode name, or use a string of eight X'40' bytes to indicate the SNA-defined default mode, which results in default values being assumed for various session parameters (see *SNA LU 6.2 Reference: Peer Protocols* for details). Omission of the mode name (byte 0 set to 1) also implies the SNA-defined default name.

Session Instance Identifier Structured Data Subfield

The Session Instance Identifier subfield may be present in both BIND and RSP(BIND).

Session Instance Identifier Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Session Instance Identifier subfield: values 2 to 9 are valid
1		X'03'
2−n		Session Instance Identifier

Session Instance Identifier Structured Data Subfield

Byte	Bit	Content
2		Format: X'00' retired in BIND, used in RSP(BIND) only when Format X'00' was used in BIND and PLU name ≤ SLU name X'01' used in BIND only X'02' used in RSP(BIND) only in response to Format X'01' in BIND X'F0' used in RSP(BIND) only when Format X'00' was used in BIND and PLU name > SLU name
3–n		Type-G symbol string identifying the session instance (generated by PLU; echoed by SLU, except for Format X'02'): null for Format X'02'; otherwise, 1 to 7 bytes.

Network-Qualified PLU Network Name Structured Data Subfield

BIND contains the Network-Qualified PLU Network Name subfield (if the name is known by the PLU).

Network-Qualified PLU Network Name Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Network-Qualified PLU Network Name subfield: values 2 to 18 (X'12') are valid
1		X'04'
2–n		Network-Qualified PLU network name *Note:* The network-qualified PLU network name is 1 to 17 bytes in length, consisting of an optional 1- to 8-byte network ID and a 1- to 8-byte LU name, both of which are type-1134 symbol strings (a character string consisting of one or more EBCDIC uppercase letters A through Z; numerics 0 through 9; the first character of which is an uppercase letter). When present, the network ID is concatenated to the left of the LU name, using a separating period and having the form "NETID.NAME"; when the network ID is omitted, the period is also omitted.

Network-Qualified SLU Network Name Structured Data Subfield

The RSP(BIND) contains the Network-Qualified SLU Network Name subfield (if the name is known by the SLU).

Network-Qualified SLU Network Name Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Network-Qualified SLU Network Name subfield: values 2 to 18 (X'12') are valid
1		X'05'
2−n		Network-Qualified SLU network name *Note:* The network-qualified SLU network name is 1 to 17 bytes in length, consisting of an optional 1- to 8-byte network ID and a 1- to 8-byte LU name, both of which are type-1134 symbol strings (a character string consisting of one or more EBCDIC uppercase letters A through Z; numerics 0 through 9; the first character of which is an uppercase letter). When present, the network ID is concatenated to the left of the LU name, using a separating period and having the form "NETID.NAME"; when the network ID is omitted, the period is also omitted.

Random Data Structured Data Subfield

The Random Data subfield contains the random data used in session-level security verification. When session-level security verification is in effect, this subfield is present in both BIND and RSP(BIND). In BIND, it carries random data to be returned enciphered in RSP(BIND); in RSP(BIND), it carries random data to be returned enciphered in FMH-12.

Random Data Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Random Data subfield: 10 is the only valid value
1		X'11'
2		Reserved
3−10		Random data: a type-G random value generated for subsequent checking in RSP(BIND) or FMH-12

Enciphered Data Structured Data Subfield

The Enciphered Data subfield is present in the RSP(BIND) when session-level security verification is in effect. This subfield contains the enciphered version of the clear random data received in BIND.

Enciphered Data Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Enciphered Data subfield: 9 is the only valid value
1		X'12'
2–9		Enciphered version of the Random Data subfield carried in BIND (using the DES algorithm and the installation-defined LU-LU password as the cryptographic key)

Chapter 8. Common Fields

Introduction

This chapter contains detailed formats of the following common fields used in message units:

- Control vectors
- Session keys

Substructure Encoding/Parsing Rules

Rules for Common Substructures

The following rules apply to encodings defined in this and the following chapter; they govern the encoding of SNA-defined RU substructures, i.e., structures such as control vectors, subvectors, and subfields that are carried within some enclosing structure and that have one-byte keys identifying the substructures. The terms *key* and *type* are used interchangeably here, since both terms are used in the substructures to which the following rules apply.

Partitioning of Key/Type Values

The use of one-byte keys means that 256 values are available for defining substructures. The available values are partitioned as follows.

Category-wide Keys: Within the category of control vectors, keys in the range X'00' to X'7F' are unique; within the independent category of management services (MS) subvectors (described in the following chapter), they are also unique.

Context-Sensitive Keys: Keys in the range X'80' to X'FD' are context-sensitive. These are unique only within the enclosing structure (e.g., a specific control vector or GDS variable). Thus, a subfield key X'80' may be defined for use within control vector X'30' and also within control vector X'31', and the subfields may be different. The only exception to this rule is found in the management services subfields (described in the following chapter). Keys in the range X'00' to X'7F' are unique only within the enclosing subvector. However, keys in the range X'80' to X'FF' are unique across the entire group of unique subvectors defined for a given management services major vector.

Parsing Rules

Common substructures with variable-length formats, such as control vectors, may be parsed in one of two ways. The parsing rule used is format specific—see the individual format description for the parsing rule used:

KL The Key field precedes the Length field and the length is the number of bytes, in binary, of the substructure's Data field (e.g., Vector Data field). The Length field value does *not* include the length of the substructure Vector Header field (consisting of the Length and Key fields).

LT The Length field precedes the Key field (also called the "type" field—hence "LT") and the length is the number of bytes, in binary, of the substructure including *both* the Vector Header field (consisting of the Length and Key fields) and the Data field.

Example of Common Substructure Format

Byte	Bit	Content
The general format of a control vector, for example, is shown as:		
0–1		Vector header; Key=X'45' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2–n		Vector Data
When the enclosing structure indicates use of parsing rule KL, the first two bytes are interpreted as:		
0		Key
1		Length (n–1), in binary, of the Vector Data field (i.e., excluding the length of the Vector Header field)
When the enclosing structure indicates use of parsing rule LT, the first two bytes are interpreted as:		
0		Length (n+1), in binary, of the control vector (i.e, including the Vector Header and Vector Data fields)
1		Type (=Key)

Some early control vectors (i.e., with low-valued Keys) have no explicit length field; these perforce appear only in the KL parsing context.

Enclosing Rule for Substructures

All substructures that are enclosed by other structures within an RU (e.g., another substructure or a GDS variable) are constructed and parsed LT. This is the case even when, for example, an enclosing control vector is parsed KL. This rule holds true for all levels of nesting.

Consider the Product Set ID (X'10') control vector as an example of this rule. Imbedded within this substructure are other substructures, specifically Product Identifier (X'11') MS common subvectors.

When the Product Set ID (X'10') is present in XID format 3, it is parsed KL, whereas when it is present within a major vector in NMVT, it is parsed LT. In both cases, the Product Identifier (X'11') subvectors are parsed LT.

Control Vectors

Introduction

The following table shows, by (category-wide) key value, the control vector (i.e., with Key < X'80' or Key = X'FE'), and the message-unit structures that can carry the control vector. This section defines only the category-wide control vectors. Context-sensitive control vectors (Keys X'80' – X'FD') are defined in-line with their enclosing structures (e.g., GDS variables).

Figure 8-1 (Page 1 of 2). Control Vector Usage

Key	*Control Vector Name*	*Applicable Message-Unit Structures by Parsing Rule*	
		KL	*LT*
X'00'	SSCP-LU Session Capabilities Control Vector	RSP(ACTLU)	
X'07'	PU FMD-RU-Usage Control Vector	RSP(ACTPU)	
X'0C'	LU-LU Session Services Capabilities Control Vector	RSP(ACTLU)	
X'0E'	Network Name Control Vector	BIND, RSP(BIND), XID	Locate GDS Variable, Route Selection Control Vector
X'10'	Product Set ID Control Vector	XID	
X'22'	XID Negotiation Error Control Vector	XID	
X'2B'	Route Selection Control Vector	BIND, RSP(BIND)	Locate GDS Variable, CD-Initiate (Reply from NN server to client EN) GDS Variable
X'2C'	COS/TPF Control Vector	BIND	CD-Initiate GDS Variable
X'2D'	Mode Control Vector	BIND	
X'33'	ENCP Search Control, Control Vector		CP Capabilities GDS Variable
X'34'	LU Definition Override Control Vector		INIT-SELF Format 0, INIT-SELF Format 1
X'35'	Extended Sense Data Control Vector	UNBIND	Locate (Reply) GDS Variable
X'36'	Directory Error Control Vector		Register Resource (Reply) GDS Variable, Delete Resource (Reply) GDS Variable
X'37'	Directory Entry Correlator Control Vector		Register Resource (Request|Reply) GDS Variable, Delete Resource (Request|Reply) GDS Variable
X'3C'	Associated Resource Entry Control Vector		Register Resource (Request) GDS Variable, Find Resource GDS Variable, Found Resource GDS Variable, Delete Resource (Request) GDS Variable

Figure 8-1 (Page 2 of 2). Control Vector Usage

Key	*Control Vector Name*	*Applicable Message-Unit Structures by Parsing Rule*	
		KL	*LT*
X'3D'	Directory Entry Control Vector		Register Resource (Request) GDS Variable, Find Resource GDS Variable, Found Resource GDS Variable, Delete Resource (Request) GDS Variable
X'40'	Real Associated Resource Entry		Find Resource GDS Variable, Found Resource GDS Variable
X'44'	Node Descriptor		Topology Database Update GDS Variable
X'45'	Node Characteristics Control Vector		Topology Database Update GDS Variable
X'46'	TG Descriptor Control Vector	XID	Route Selection Control Vector, Topology Database Update GDS Variable, CD-Initiate GDS Variable
X'47'	TG Characteristics Control Vector		Topology Data Base Update GDS Variable, CD-Initiate GDS Variable
X'60'	Fully Qualified PCID Control Vector	BIND, RSP(BIND), UNBIND	Locate GDS Variable
X'66'	Length-Checked Compression	BIND, RSP(BIND)	

Control Vector Formats

The control vectors having Key < X'80' are defined as follows (with 0-origin indexing of the vector bytes—see the individual RU description for the actual displacement within the RU). Control vectors having Key ≥ X'80' are defined following the substructure (such as a GDS variable) in which they appear.

Note: When more than one control vector may appear in an RU, unless otherwise stated, the vectors may appear in any order.

SSCP-LU Session Capabilities (X'00') Control Vector

SSCP-LU Session Capabilities (X'00') Control Vector

Byte	Bit	Content
0		Key: X'00'
1		Maximum RU size sent on the normal flow by either half-session: if bit 0 is set to 0, then no maximum is specified and the remaining bits 1 – 7 are ignored; if bit 0 is set to 1, then the byte is interpreted as X'ab' = $a \times 2^b$ (Notice that, by definition, a≥8 and therefore X'ab' is a normalized floating point representation.) See Figure 5-1 on page 5-16 for all possible values.

SSCP-LU Session Capabilities (X'00') Control Vector

Byte	Bit	Content
2–3		LU Capabilities
	0	Character-coded capability: 0 The SSCP may not send unsolicited character-coded requests; a *solicited* request is a reply request or a request that carries additional error information to supplement a previously sent negative response or error information after a positive response has already been sent. 1 The SSCP may send unsolicited character-coded requests.
	1	Field-formatted capability: 0 The SSCP may not send unsolicited field-formatted requests. 1 The SSCP may send unsolicited field-formatted requests.
	2–15	Reserved
4		Reserved

PU FMD-RU-Usage (X'07') Control Vector

PU FMD-RU-Usage (X'07') Control Vector

Byte	Bit	Content
0		Key: X'07'
1	0–5	Reserved
	6	Adjacent PU load capability (initialized to 0 by the PU T2): 0 Adjacent PU cannot load the T2 node. 1 Adjacent PU can load the T2 node (set by the boundary function in the adjacent subarea node).
	7	FMD request capability of the node: 0 PU cannot receive FMD requests from the SSCP. 1 PU can receive FMD requests from the SSCP.
2–7		Reserved

LU-LU Session Services Capabilities (X'0C') Control Vector

Note: Do not confuse control vector X'0C' with NOTIFY vector X'0C', which carries similar information.

LU-LU Session Services Capabilities (X'0C') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'0C' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–m		Vector Data
2	0–3	Reserved
	4–7	Secondary LU capability: 0000 SLU capability is inhibited: sessions can be neither queued nor started. 0001 SLU capability is disabled: sessions can be queued but not started. 0010 reserved 0011 SLU capability is enabled: sessions can be queued or started.
3–4		LU-LU session limit: 0001 session limit of 1 (only value allowed for peripheral LUs)
5–6		LU-LU session count: the number of LU-LU sessions that are not reset for this LU, and for which SESSEND will be sent to the SSCP
7		Reserved

Network Name (X'0E') Control Vector

Network Name (X'0E') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'0E' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3) *Note:* A null X'0E' control vector consists of a vector header with no vector data. The length field is set appropriately.
2–n		Vector Data
2		Network name type: X'F1' PU name (not network-qualified) X'F3' LU name X'F4' CP name (see Note) X'F5' SSCP name X'F6' NNCP name X'F7' link station name (not network-qualified) *Note:* When this control vector is carried in some message units, such as XID3 or BIND, X'F4' means simply "CP name," without specifying the CP type (e.g., EN or NN), and X'F6' is not used; see each individual message-unit structure in which this control vector appears for details on such usage.

Network Name (X'0E') Control Vector

Byte	Bit	Content
3–n		Network-qualified name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type 1134-symbol-string network identifier concatenated with a period (when the qualifier is not present, the period is omitted). The network-qualified name appears, for example, as follows: NETID.NAME, with optional (but not significant) trailing, but no imbedded, space (X'40') characters. As noted in Appendix A, "SNA Character Sets and Symbol-String Types," implementation usage constrains the leading character of the name to be alphabetic.

Product Set ID (X'10') Control Vector

Product Set ID (X'10') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'10' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–n		Vector Data
2		Retired
3–n		Network product identifier: one or two Product Identifier (X'11') MS common subvectors, as described in "MS Common Subvectors" on page 9-325, one for each hardware product and software product in the implementation of the node

XID Negotiation Error (X'22') Control Vector

XID Negotiation Error (X'22') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'22' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–n(=4|8)		Vector Data
2–3		Error byte offset: the binary offset (0-origin in the XID information field) of the first byte of the field in error
4		Error bit offset: the binary offset (0-origin in the byte pointed to in the Error Byte Offset field) of the first bit of the field in error

XID Negotiation Error (X'22') Control Vector

Byte	Bit	Content
5–8		Optional sense data

Route Selection (X'2B') Control Vector

The Route Selection control vector (RSCV) is carried in BIND, RSP(BIND), and other RUs to describe the path through an APPN network that a session is to take or has taken; or in Locate to define the Locate search procedure path.

Route Selection (X'2B') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'2B' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–n		Vector Data
2		Maximum hop count: the number, in binary, of TG Descriptor control vectors in the Route Selection control vector
3		Current hop count: the index, in binary, of the last TG Descriptor control vector that was processed; the index divides the traversed from the to-be-traversed portions of the path and, thus, points to the next hop to be traversed *Note:* When the values of the Maximum Hop Count field and the Current Hop Count field are equal, all nodes specified in the control vectors have processed the RU.
4–n		Control vectors, as described in the section "Control Vectors" on page 8-3 *Note:* The following control vectors may be included; they are parsed according to parsing rule LT. X'0E' Network Name: one for each control point (Type = X'F4') on the procedure path (present when the RSCV is carried on Locate) X'46' TG Descriptor control vector: one for each TG on the session path (present when the RSCV is carried on a BIND or RSP(BIND), or on a reply CD-Initiate from an NN server to its client ENCP)

COS/TPF (X'2C') Control Vector

COS/TPF (X'2C') Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'2C' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–m		Vector Data
2	0–4	Reserved
	5	Network priority indicator: 0 PIUs for this session flow at the priority specified in the Transmission Priority field (bits 6–7). 1 PIUs for this session flow at network priority, which is the highest transmission priority.
	6–7	Transmission priority (reserved if byte 2, bit 5 = 1): 00 low priority 01 medium priority 10 high priority 11 reserved
3		Length, in binary, of COS Name field
4–m		COS name: 0 to 8 type-1134 symbol-string characters with optional (but not significant) trailing space (X'40') characters

Mode (X'2D') Control Vector

Mode (X'2D') Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'2D' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–n		Vector Data
2		Length, in binary, of Mode Name field
3–n		Mode name: 0 to 8 type-1134 symbol-string characters with optional (but not significant) trailing space (X'40') characters

ENCP Search Control (X'33') Control Vector

ENCP Search Control (X'33') Control Vector

Byte	Bit	Content
0–1		Vector Header; Key = X'33' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2	0	Search status indicator: 0 Do not search the sending ENCP for resources of the type indicated in ENCP Resource Type field (bytes 4–5) unless there is a directory entry for the resource in the network directory database. 1 Search the sending ENCP when needed for resources of the type indicated in ENCP Resource Type field (bytes 4–5).
	1–7	Reserved
3		Reserved
4–5(=n)		ENCP resource type: X'00F3' logical unit (only value defined)

LU Definition Override (X'34') Control Vector

The LU Definition Override control vector carries data from the SLU to temporarily override specifications in the SLU's SSCP. The data items that can be overridden are the model name and associated LU names that will be passed to the PLU via CINIT in control vector X'2F' during implementation logon processing.

LU Definition Override (X'34') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'34' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–n		Vector Data
2		Length, in binary, of model name (0 if none)
3–f		Model name (omitted if Length of Model Name = 0, in which case the next field, Length of Primary Printer Name, immediately follows Length of Model Name): a 1- to 8-byte type-A symbol-string giving the name of the model definition that the PLU should use for the SLU that is logging on; or, alternatively, an all-space (X'40') string representing a null name
f+1		Length, in binary, of primary printer name (0 if none)

LU Definition Override (X'34') Control Vector

Byte	Bit	Content
f+2−g		Primary printer name (omitted if Length of Primary Printer Name = 0, in which case the next field, Length of Alternate Printer Name, immediately follows Length of Primary Printer Name): a 1- to 8-byte type-A symbol-string giving the name of the primary printer that the PLU should associate with the SLU that is logging on (*Note:* This is always the name by which the printer is known in the SLU's network); or, alternatively, an all-space (X'40') string representing a null name
g+1		Length, in binary, of alternate printer name (0 if none)
g+2−n		Alternate printer name (omitted if Length of Alternate Printer Name = 0, in which case the Length of Alternate Printer Name field is at byte n): a 1- to 8-byte type-A symbol-string giving the name of the alternate printer to be associated with the SLU that is logging on (*Note:* This is always the name by which the printer is known in the SLU's network); or, alternatively, an all-space (X'40') string representing a null name

Extended Sense Data (X'35') Control Vector

Extended Sense Data (X'35') Control Vector

Byte	Bit	Content
0−1		Vector header; Key=X'35' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2−p		Vector Data
2−5		Sense data
Note: *The shorter abbreviated form (now retired) of the control vector ends here.*		
6−p		Extended Sense Information

Extended Sense Data (X'35') Control Vector

Byte	Bit	Content
6	0	RU information included: 0 RU information not included (bits 1−2 set to 00 and bytes 8−m are not included) 1 RU information included (see bytes 8−m below)
	1−2	RU category of the RU in error (reserved when bit 0 = 0): 00 FMD 01 NC 10 DFC 11 SC
	3	FMD message-unit type (reserved when RU category is not FMD): 0 FMD message unit is not a GDS variable. 1 FMD message unit is a GDS variable (only value used on Locate).
	4	Generator of Extended Sense Data control vector (reserved when Termination Procedure Origin Name field not present): 0 the termination procedure origin (only value used on Locate) 1 a node other than the termination procedure origin
	5	Contents of Termination Procedure Origin Name field (reserved when Termination Procedure Origin Name field not present): 0 termination procedure origin name (only value used on Locate) 1 name of node other than termination procedure origin, as described below; termination procedure origin name not known
	6−7	Reserved
7		Length, in binary, of RU or GDS Variable Identifier field (set to 0 when byte 6, bit 0 = 0)
8−m		Identifier: request code, NS header, or GDS variable identifier; if present, this field identifies the request or response that triggered the generation of the

Note: *The longer abbreviated form of the control vector ends here.*

m+1		Length of Termination Procedure Origin Name field (values 3 to 26 are valid)

Extended Sense Data (X'35') Control Vector

Byte	Bit	Content
m+2−n		Termination procedure origin name: if the field contains the termination procedure origin name (see byte 6, bit 5), network-qualified CP name of the node that caused the session termination procedure to be executed; otherwise, the network-qualified CP name of the node that generated the Extended Sense Data control vector, with, when available, a local or network name (in the Related Resource Name field) that indicates the direction from which the RU signaling the termination procedure was received *Note 1:* When the termination procedure origin is a CP, the network-qualified CP name is used (e.g., NETID.CPNAME); when the termination procedure origin is an SSCP and a T4|T5 node caused the SSCP to begin session termination, the T4|T5 PU name is included in the Related Resource Name field; when a boundary function is the termination procedure origin, the network-qualified BF PU name is used; when a boundary function generates the Extended Sense Data control vector, but the termination procedure origin name is unknown, the adjacent link station name is appended to the network-qualified PU name with a period as the separator (e.g., NETID.PUNAME[.ALSNAME]). *Note 2:* The network identifier is always included in the termination procedure origin name.
Note: *The following fields are omitted when this control vector is used in Locate.*		
n+1−p		Related resource (If the length in byte n+1 is 0, the Related Resource field may be omitted.)
n+1		Length (0−17), in binary, of Related Resource Name field (always 0 when used on Locate)
n+2−p		Related resource name: the name of a related resource used to identify the source of the error (for example, the name of the PU that rejected the RNAA for an address assignment error reported cross-domain) *Note:* The name always belongs to the same network as the termination procedure origin name; therefore, the network identifier is not included.

Directory Error (X'36') Control Vector

The Directory Error X'36' control vector is used to report a directory request error.

Directory Error (X'36') Control Vector

Byte	Bit	Content
0−1		Vector header; Key = X'36' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)

Directory Error (X'36') Control Vector

Byte	Bit	Content
2–5(=n)		Sense data

Directory Entry Correlator (X'37') Control Vector

The Directory Entry Correlator (X'37') control vector contains a correlator value generated by a request sender and echoed by a reply sender to correlate an error signaled in a reply with the resource entry in the request that caused the processing error.

Directory Entry Correlator (X'37') Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'37' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–n		Implementation-defined correlator value

Associated Resource Entry (X'3C') Control Vector

The Associated Resource Entry (X'3C') control vector is used to specify hierarchical associations between directory entries. The resource identified by the X'3C' control vector is hierarchically related immediately above the resource identified by a following X'3C' control vector or above the resources identified by one or more following X'3D' control vectors.

Associated Resource Entry Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'3C' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–3		Resource type: X'00F4' ENCP X'00F6' NNCP
4–n		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted)

Associated Resource Entry Control Vector

Byte	Bit	Content
		Note: The network ID is always present when different from the network ID of the receiver.

Directory Entry (X'3D') Control Vector

The Directory Entry (X'3D') control vector provides the resource name and type for a directory entry.

Directory Entry (X'3D') Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'3D' (see "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–3		Resource type: X'00F3' logical unit X'00F4' ENCP X'00F6' NNCP *Note:* The hierarchically highest type may be specified in the case of a merged CP/LU (CP=LU) — i.e., where the identified resource is both a control point and an LU serving end-user sessions. In this case, the duplicative Associative Resource Entry (X'3C') control vector is omitted. So for NNCP=LU for example, X'00F6' may be specified here, and no X'3C' control vector precedes this X'3D' control vector. For ENCP=LU, X'00F4' may be specified here, and the X'3C' control vector that would otherwise (i.e., in the LU≠CP case) precede this control vector to specify the ENCP is omitted.
4–n		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted) *Note:* The network ID, if omitted, is assumed to be the same as that of the hierarchically related X'3C' control vector, or, if that is absent, the same as that of the receiver.

Real Associated Resource Entry (X'40') Control Vector

The Real Associated Resource Entry (X'40') control vector is used to specify hierarchical associations for directory entries. The X'40' control vector is used to specify the resource identified in the Directory Entry (X'3D') control vector preceding it. The Real Associated Resource Entry control vector is used when an Associated Resource Entry (X'3C') in the hierarchy (preceding the subject X'3D' control vector) does not represent the real hierarchical superior of the target resource but rather is a surrogate.

Real Associated Resource Entry Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'40' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2–3		Resource type: X'00F6' NNCP
4–n		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted) *Note:* The network ID is always present when different from the network ID used in the corresponding Directory Entry control vector.

Node Descriptor (X'44') Control Vector

The Node Descriptor control vector identifies the node for which a topology update is being reported.

Node Descriptor (X'44') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'44' (See "Substructure Encoding/Parsing Rules" on page 8-1 and Figure 8-1 on page 8-3)
2–p		Vector Data
2		Length (1–17), in binary, of the node's network-qualified CP name (some back-level nodes omit the network ID qualifier)
3–m		Network-qualified CP name: a 2-part name consisting of a qualifier concatenated by a CP name, each part being a 1- to 8-byte type-1134 symbol string

Node Descriptor (X'44') Control Vector

Byte	Bit	Content
m+1		Length (0 for a T2.1 node; n−m−1 otherwise), in binary, of Additional Address Information field
m+2−n		Additional address information: reserved
n+1	0	Connection network indicator: 0 The Network-Qualified CP Name field does not identify a connection network. 1 The Network-Qualified CP Name field does identify a connection network.
	1−7	Reserved

Node Characteristics (X'45') Control Vector

The Node Characteristics control vector carries the characteristics of a node that may be reported as part of a topology update.

Node Characteristics (X'45') Control Vector

Byte	Bit	Content
0−1		Vector header; Key=X'45' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2−n		Vector Data *Note:* The following subfields may be included. They are parsed according to subfield parsing rule LT. X'80' Node Type and Status subfield (always present)

Node Type and Status (X'80') Node Characteristics Subfield

The Node Type and Status subfield carries the node type and status data that may be reported as part of a topology update. It is carried on the Node Characteristics (X'45') control vector

Node Type and Status (X'80') Node Characteristics Subfield

Byte	Bit	Content
0		Length of Node Type and Status subfield
1		Key: X'80'

Node Type and Status (X'80') Node Characteristics Subfield

Byte	Bit	Content
2–n		Subfield Data: Reserved bytes may be truncated (and the Length field adjusted accordingly), but implementations can always accept, store, and forward 8 bytes of node type and status data.
2–5		Resource sequence number: a 32-bit binary value that uniquely identifies a topology update. It is always incremented by 2 by the node that creates the update. When the low-order bit of the sequence number is set, a receiving node has recognized inconsistent data; that is, more than one update with the same node descriptor and with the same sequence number but with different data has been received. The number space is linear; that is, it does not wrap. If the end of the number space is reached, the CP of the node is renamed, and sequence numbering recommences with 2.
6		Route addition resistance: a binary value in the range 0–255; the greater the value, the less the ability of the node to accept additional routes
7		Node status (a value of 1 indicates that the condition is true):
	0	Node congested
	1	Intermediate routing resources depleted
	2	Endpoint routing resources depleted
	3–4	Reserved
	5	Quiescing
	6–7	Reserved
8		Node Type and Support
	0–1	Reserved
	2	1 (only value defined)
	3	1 (only value defined)
	4–5	Reserved for management services
	6–7	Node type: 11 T2.1 (only value currently defined)
9(=n)		Additional Node Support
	0	Adjacent subnet border node support: 0 The node lacks such support. 1 The node has such support.
	1–7	Reserved

TG Descriptor (X'46') Control Vector

The TG Descriptor control vector identifies a transmission group (TG).

TG Descriptor (X'46') Control Vector

Byte	Bit	Content
0−1		Vector header; Key = X'46' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2−n		Vector Data *Note:* The following subfields may be included. They are parsed according to subfield parsing rule LT. X'80' TG Identifier subfield (always present) X'82' DLC Signaling Information subfield (present only when the X'80' subfield indicates the TG is to a link connection network) X'83' Real Partner CP name subfield (present in a CD-Initiate in a Locate reply [or request] when a border node modifies the associated resource hierarchy such that the CP (DLU) [or NNS (OLU)] is not adjacent to the NNS (DLU) [or NNS (OLU)]; or in an RSCV when an NNS (OLU) calculates a route that includes a TG vector carrying it): when present, used in preference to the TG-partner node's CP name in the TG Identifier (X'80') subfield

TG Identifier (X'80') TG Descriptor Subfield

TG Identifier (X'80') TG Descriptor Subfield

Byte	Bit	Content
0		Length, in binary, of TG Identifier subfield
1		Key: X'80'
2		TG number: the binary integer negotiated during XID exchange to represent the TG to the partner node on the TG
3		Length, in binary, of TG-partner node's network-qualified CP name; values 0 to 17 are valid
4−n		TG-partner node's network-qualified CP name: the name of the CP in the node at the opposite end of the TG
n+1	0	Link connection network indicator: 0 The TG-Partner Node's Network-Qualified CP Name field does not identify a link connection network (e.g., a local-area network). 1 The TG-Partner Node's Network-Qualified CP Name field does identify a link connection network; in this case, bytes 4−n contain the CP name representing the virtual routing node.
	1−4	Reserved
	5	Intersubnet link indicator: 0 This link is not an intersubnet link 1 This link is an intersubnet link (defines a border between subnets).
	6−7	Reserved

DLC Signaling Information (X'82') TG Descriptor Subfield

DLC Signaling Information (X'82') TG Descriptor Subfield

Byte	Bit	Content
0		Length, in binary, of DLC Signaling Information subfield
1		Key: X'82'
2–n		Subfield Data
2–n		DLC-specific data related to the link connection network:
For Token-Ring (Only DLC Currently Allowed)		
2–7		MAC address *(See IBM Token-Ring Network Architecture Reference)*
8(=n)		LSAP address *(See IBM Token-Ring Network Architecture Reference)*

Real Partner CP Name (X'83') TG Descriptor Subfield

Real Partner CP Name (X'83') TG Descriptor Subfield

Byte	Bit	Content
0		Length, in binary, of Real Partner CP Name subfield
1		Key: X'83'
2		Length, in binary, of the TG-partner node's network-qualified CP name; values 1 to 17 are valid.
3–n		Network-qualified CP name of the real TG-partner node: the name of the CP in the node at the opposite end of the TG. This subfield indicates the name specified in the X'80' subfield does not reflect the true name of the CP located on the opposite end of the TG.

TG Characteristics (X'47') Control Vector

The TG Characteristics control vector carries the characteristics of a transmission group (TG).

TG Characteristics (X'47') Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'47' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2–n		Vector Data
2–5		Resource sequence number (reserved except in TDUs): a 32-bit binary value that uniquely identifies a topology update. It is always incremented by 2 by the node that creates the update. When the low-order bit of the sequence number is set to 1, a node has recognized inconsistent receiving data; that is, more than one update with the same TG descriptor and with the same sequence number but with different data has been received. The number space is linear; that is, it does not wrap for a given CP name. If the end of the number space is reached, the node's CP is renamed, and sequence numbering recommences with 2.
6		Status:
	0	Operational status: 0 The TG is not operational. 1 The TG is operational (only value sent by an end node).
	1	Reserved
	2	Quiescing (reserved except in TDUs): 0 The TG is not quiescing. 1 The TG is quiescing.
	3	CP-CP session support status: 0 CP-CP sessions supported on this TG 1 CP-CP sessions not supported on this TG
	4–7	Reserved
7		Effective capacity: a floating-point number, in units of 300 bits per second, representing the product of a user-defined maximum load factor and the bit transmission rate of the link underlying the TG (see Note 1 at the end of this control vector for the encoding of this field)
8–12		Reserved
13		Cost per connect time: a value representing the relative cost per unit time of using the TG; permissible values are 0–255, where the value 0 means free
14		Cost per byte transmitted: a value representing the relative cost of transmitting a byte over the TG; permissible values are 0–255, where the value 0 means free
15		Reserved

TG Characteristics (X'47') Control Vector

Byte	Bit	Content
16		Security: X'01' not secure X'20' public switched network used; secure in the sense that traffic takes no predetermined route X'40' underground cable; located in a secure country (as determined by the network administrator) X'60' secure conduit containing the transmission medium, not guarded (e.g., pressurized pipe) X'80' guarded conduit containing the transmission medium, protected against physical tapping X'A0' link-level encryption used X'C0' guarded conduit containing the transmission medium, protected against physical and radiation tapping
17		Propagation delay: propagation delay of the TG; given as a floating-point number (see Note 1) specifying microsecond units. The following default values and ranges are defined.
18		Reserved
19		User-defined parameter 1
20		User-defined parameter 2
21(=n)		User-defined parameter 3

default	meaning	range(decimal)	range(hex)
X'00'	"minimum"	–	–
X'4C'	negligible	<0.48 ms	<X'4F'
X'57'	terrestrial	0.48-49.152 ms	X'4F'-X'84'
X'91'	packet	49.152-245.76 ms	X'84'-X'97'
X'99'	long	>245.76 ms	>X'97'
X'FF'	"maximum"	–	–

Note 1: Floating-point numbers represented above include (effectively) 4 bits of mantissa and 5 bits of exponent, encoded as follows. The binary value is first normalized and the exponent adjusted appropriately. The mantissa to be encoded consists of the four bits to the right of the binary point. The remaining bits are truncated. The high-order bit of the mantissa is not included in the encoding, since by the normalization it is 1.

The encoded value in the corresponding byte fields above consists of the remaining three bits of the mantissa as the low-order three bits and the exponent as the high-order five bits.

As an example, consider the value 23, binary 10111, which normalized is $.10111_2 \times 2^5$. Truncating all but the most significant four bits, leaves $.1011_2 \times 2^5$ (equals 22). The high-order 1 of the mantissa is assumed, so the encoded value is 00101 011.

Note 2: All implementations can accept, store, and forward 20 bytes of TG characteristics.

Fully Qualified PCID (X'60') Control Vector

The Fully Qualified Procedure Correlation Identifier (FQPCID) is a unique value throughout an entire network.

Fully Qualified PCID (X'60') Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'60' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2–p		Vector Data
2–9		PCID: a unique value used as a procedure identifier
10		Length of Network-Qualified CP Name field (values 3 to 17 are valid)
11–n		Network-qualified CP name (network identifier always present)

Length-Checked Compression (X'66') Control Vector

The Length-Checked Compression control vector carries information depending on the RU to which it is appended:

- For BIND and RSP(BIND), it carries the information necessary to negotiate compression.

Length-Checked Compression (X'66') Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'66'
2–n		Vector Data *Note:* One of the following subfields is included, depending on the RU; they are parsed according to parsing rule LT. X'81' RLE/LZ Compression Bid subfield (present on BIND if session is to use length-checked compression) X'82' RLE/LZ Compression Result subfield (present on RSP(BIND) if session is to use length-checked compression)

RLE/LZ Compression Bid (X'81') Length-Checked Compression Subfield

The RLE/LZ Compression Bid subfield is used to gather information about both the level of compression available and the level of compression needed along the path between the session endpoints. This subfield is used only when control vector X'66' is carried on BIND.

RLE/LZ Compression Bid (X'81') Length-Checked Compression Subfield

Byte	Bit	Content
0		Length, in binary, of this subfield
1		Key: X'81'
2–4		Subfield Data
2		Length-Checked Compression Flags
	0	Compression and decompression involvement: 0 Length-checked compression and decompression may be performed by intermediate nodes. 1 Length-checked compression and decompression are done only by the session endpoints.
	1	RLE usage: 0 LZ and RLE compression in series is not allowed. 1 LZ and RLE compression in series is allowed.
	2–7	Reserved
3		Negotiable Compression Fields
	0–3	Compression capability of nodes so far: X'0' no compression X'1' RLE compression only X'2' RLE and small-table LZ compression X'3' RLE and medium-table LZ compression X'4' RLE and large-table LZ compression
	4–7	Compression level needed by links so far: X'0' no compression X'1' RLE compression only X'2' small-table LZ compression, optionally preceded by RLE X'3' medium-table LZ compression, optionally preceded by RLE X'4' large-table LZ compression, optionally preceded by RLE
4		Static Compression Fields
	0–3	Desired compression level of RUs going from PLU to SLU: X'0' no compression X'1' RLE compression only X'2' small-table LZ compression, optionally preceded by RLE X'3' medium-table LZ compression, optionally preceded by RLE X'4' large-table LZ compression, optionally preceded by RLE *Note:* The desired compression level represents maximum capability. Actual compression may be negotiated to a lower level.

RLE/LZ Compression Bid (X'81') Length-Checked Compression Subfield

Byte	Bit	Content
	4–7	Desired compression level of RUs going from SLU to PLU: X'0' no compression X'1' RLE compression only X'2' small-table LZ compression, optionally preceded by RLE X'3' medium-table LZ compression, optionally preceded by RLE X'4' large-table LZ compression, optionally preceded by RLE *Note:* The desired compression level represents maximum capability. Actual compression may be negotiated to a lower level.

RLE/LZ Compression Result (X'82') Length-Checked Compression Subfield

The RLE/LZ Compression Result subfield establishes on the RSP(BIND) the level of compression/decompression to be performed by each compression-capable node along the session path. This subfield is used only when control vector X'66' is attached to the RSP(BIND).

RLE/LZ Compression Result (X'82') Length-Checked Compression Subfield

Byte	Bit	Content
0		Length of this subfield
1		Key: X'82'
2–3		Subfield Data
2		Length-Checked Compression Flags
	0	Compression and decompression involvement: 0 Length-checked compression and decompression may be performed by intermediate nodes. 1 Length-checked compression and decompression are done only by the session endpoints.
	1	RLE usage: 0 LZ and RLE compression in series is not allowed. 1 LZ and RLE compression in series is allowed.
	2–7	Reserved
3		Negotiable Compression Fields
	0–3	Actual compression level of RUs going from PLU to SLU: X'0' no compression X'1' RLE compression only X'2' small-table LZ compression, optionally preceded by RLE X'3' medium-table LZ compression, optionally preceded by RLE X'4' large-table LZ compression, optionally preceded by RLE

RLE/LZ Compression Result (X'82') Length-Checked Compression Subfield

Byte	Bit	Content
	4–7	Actual compression level of RUs going from SLU to PLU: X'0' no compression X'1' RLE compression only X'2' small-table LZ compression, optionally preceded by RLE X'3' medium-table LZ compression, optionally preceded by RLE X'4' large-table LZ compression, optionally preceded by RLE

Session Keys

The following table shows, by key value, the session key and the message-unit structures that can carry the session key.

Key	Session Key	Applicable Message-Unit Structures
X'06'	Uninterpreted Name Pair	NOTIFY, NSPE
X'0A'	URC	TERM-SELF

The *session keys* are defined as follows, with 0-origin indexing of the key bytes—see the individual RU description for the actual displacement within the RU.

Network Name Pair or Uninterpreted Name Pair (X'06') Session Key

Network Name Pair or Uninterpreted Name Pair (X'06') Session Key

Byte	Bit	Content
0		Key: X'06'
1		Type: X'F3' logical unit
2		Length, in binary, of PLU (or OLU or LU1) name
3−m		Name in EBCDIC characters (see Note below)
m+1		Type: X'F3' logical unit
m+2		Length, in binary, of SLU (or DLU or LU2) name
m+3−n		Name in EBCDIC characters (see Note below)

Note: *For a Network Name Pair session key, the names consist of type-1134 symbol-string characters; for an Uninterpreted Name Pair session key, the names are any EBCDIC strings.*

URC (X'0A') Session Key

URC (X'0A') Session Key

Byte	Bit	Content
0		Key: X'0A'
1		Length, in binary, of the URC
2−n		URC: LU-defined identifier

Chapter 9. SNA/MS Encodings

Introduction

The following table shows, by key value, the MS major vectors and the the message-unit structures that can carry these major vectors:

Key	MS Major Vector	Applicable Message-Unit Structures
X'0000'	Alert	NMVT, CP-MSU
X'0002'	Resolution	NMVT, CP-MSU
X'0050'	Change Control	CP-MSU
X'0061'	Reply to Execute Command	NMVT, CP-MSU
X'0062'	Reply to Analyze Status	NMVT, CP-MSU
X'0063'	Reply to Query Resource Data	NMVT, CP-MSU
X'0064'	Reply to Test Resource	NMVT, CP-MSU
X'0066'	Activation Acceptance	CP-MSU
X'0067'	Activation	CP-MSU
X'0068'	Initiation	CP-MSU
X'006F'	Send Message To Operator	NMVT, CP-MSU
X'0070'	Operate Report	CP-MSU
X'0071'	Deactivation Acceptance	CP-MSU
X'0072'	Deactivation	CP-MSU
X'0075'	Set Clock Report	CP-MSU
X'0076'	Cancelation	CP-MSU
X'0077'	Routing/Parsing Report	CP-MSU
X'0080'	RTM	NMVT
X'0090'	Reply Product Set ID	NMVT
X'8050'	Request Change Control	CP-MSU
X'8061'	Execute Command	NMVT, CP-MSU
X'8062'	Analyze Status	NMVT, CP-MSU
X'8063'	Query Resource Data	NMVT, CP-MSU
X'8064'	Test Resource	NMVT, CP-MSU
X'8066'	Request Activation	CP-MSU
X'8068'	Request Initiation	CP-MSU
X'8070'	Operate	CP-MSU
X'8071'	Request Deactivation	CP-MSU
X'8075'	Set Clock	CP-MSU
X'8076'	Request Cancelation	CP-MSU
X'8080'	Request RTM	NMVT
X'8090'	Request Product Set ID	NMVT

The following table shows, by key value, the MS parameter major vectors and the message-unit structures that can carry these parameter major vectors:

Key	MS Parameter Major Vector	Applicable Message-Unit Structures
X'1300'	Text Data	NMVT, CP-MSU
X'1307'	Structured Data	NMVT, CP-MSU
X'1309'	Transparent Coded Datastream	NMVT, CP-MSU
X'130A'	Begin Data Parameters	NMVT, CP-MSU
X'130B'	End Parameter Data	NMVT, CP-MSU
X'1730'	Initiate Agent Request	CP-MSU
X'1731'	Initiate Agent Report	CP-MSU

Note: The major vectors and parameter major vectors are defined as follows (using 0-origin indexing):

- The description of each major vector includes a matrix indicating the subvectors that may be included within it.
- Subvectors with keys X'80' through X'FE' have a meaning that is unique only to the major vector in which they are used. They are defined following each major vector.
- Subvectors with keys X'00' through X'7F' are referred to as common subvectors. Their meaning is independent of the major vector in which they are used. They are defined in "MS Common Subvectors" on page 9-325.
- Subvectors may appear in any order within a major vector unless otherwise stated.

The following table shows, by key value, the MS GDS Variables and the message-unit structures that can carry these GDS Variables:

Key	MS GDS Variables	Applicable Message-Unit Structures
X'1320'	Text Command	CP-MSU
X'1321'	Partial Format Processing Method	CP-MSU
X'1322'	Display Datastream	CP-MSU
X'1323'	Context-Identified Values	CP-MSU
X'1324'	Context Identifier Group	CP-MSU
X'1325'	Context Identifier	CP-MSU
X'1326'	Value Group	CP-MSU
X'1327'	Character Value Descriptor	CP-MSU
X'1328'	Value Processing Method	CP-MSU
X'1329'	Value	CP-MSU
X'132A'	Hexadecimal Value Descriptor	CP-MSU
X'132B'	Nested Value Processing Method	CP-MSU
X'1331'	Value Instance Identifier	CP-MSU
X'1732'	Command	CP-MSU
X'1735'	Job Element Spec	CP-MSU
X'1736'	Submission Type	CP-MSU
X'1739'	Job Element State	CP-MSU
X'173A'	Object Number	CP-MSU
X'1741'	Report Data Prefix	CP-MSU

Key	MS GDS Variables	Applicable Message-Unit Structures
X'1746'	Report Data Suffix	CP-MSU
X'1747'	Object Disposition	CP-MSU
X'1748'	Complettion Report	CP-MSU
X'1749'	Command Procedure Parameters	CP-MSU

MS Major Vector Formats

Alert (X'0000') MS Major Vector

PU → SSCP, CP → CP

This major vector provides unsolicited notification of a problem or impending problem, type of problem, identification of the cause, and identification of the component that caused the problem.

Alert (X'0000') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'0000'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Alert MS Subvectors" on page 9-7 for subvector keys X'80'–X'FE'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Alert (X'0000') Major Vector		
	NMVT	CP-MSU	Notes
Text Message (X'00')	O	O	Note 1
Date/Time (X'01')	CP	P	Note 2
Hierarchy Name List (X'03')	O	—	Note 3
SNA Address List (X'04')	CP	—	Note 4
Hierarchy/Resource List (X'05')	CP	P	Note 5
Product Set ID (X'10')	P(n)	P(n)	Note 6
Self-Defining Text Msg. (X'31')	O	O	
Relative Time (X'42')	CP	CP	Note 7
MSU Correlation (X'47')	CP	CP	Note 8
Supporting Data Corr. (X'48')	CP	CP	Note 9
Incident Identification (X'4A')	CP	CP	Note 10
LAN Link Connection Subsystem Data (X'51')	CP	CP	Note 11
LCS Configuration Data (X'52')	CP	CP	Note 12
Link Station Data (X'8C')	CP	CP	Note 12
Basic Alert (X'91')	O	O	Note 13
Generic Alert Data (X'92')	P	P	
Probable Causes (X'93')	P	P	
User Causes (X'94')	CP	CP	Note 14
Install Causes (X'95')	CP	CP	Note 14
Failure Causes (X'96')	CP	CP	Note 14
Cause Undetermined (X'97')	CP	CP	Note 15
Detailed Data (X'98')	O	O	
Detail Qualifier(X'A0' or X'A1')	O(n)	O(n)	Note 16

Key:

—	Not present
P	Present one time
P(n)	Present one or more times
CP	Conditionally present one time (See Notes for conditions)
O	Optionally present one time
O(n)	Optionally present one or more times

Notes:

1. This subvector may be optionally included by an Alert sender, to transport text in a single Alert major vector that can be processed by both a non-generic Alert and a generic Alert focal point. If this subvector is present, the X'91' subvector must also be present.

2. If the PU sending the Alert major vector has the capability of providing it, it places this subvector in the NMVT. See Note 7. It is always present in a CP-MSU.
3. This subvector may be optionally included in the NMVT by an Alert sender in order to create a single Alert major vector that can be processed by both a non-generic Alert and a generic Alert focal point. When it is present, this subvector identifies an origin of the Alert condition that is not an SNA network addressable unit. If this subvector is present, the X'91' subvector must also be present.
4. This subvector is present when it is necessary to identify, with an SNA address, the origin of the Alert condition. If the origin of the Alert condition is the PU sending the Alert, this subvector is not present.
5. This subvector is present in the NMVT instead of, or in addition to, the SNA Address List (X'04') subvector if the origin (other than the PU sending the Alert) of the Alert condition cannot be represented in the SNA Address List (X'04') subvector. This subvector is always present in a CP-MSU.
6. An instance of this subvector describing the PU or CP sending the Alert is always present. A second instance is present if the origin of the Alert condition is a hardware or software product, and is not the PU or CP sending the Alert. If a second instance is present, it is placed immediately after the first instance of the X'10' subvector.

   In an Alert containing two instances of the Product Set ID subvector, the following terms refer, respectively, to these two instances:

   - "Alert Sender PSID" identifies the PU sending the Alert
   - "Indicated Resource PSID" identifies the resource on which the Alert is reporting

   In an Alert with only one instance of the Product Set ID, this instance is referred to both as the Alert Sender Product Set ID and as the Indicated Resource Product Set ID.
7. If the PU sending the Alert cannot provide a Date/Time (X'01') subvector, it places this subvector in the NMVT instead.
8. This subvector is present if the Alert sender is including a token for correlating the Alert with another MSU reporting on the same resource as the Alert.
9. This subvector is present if the Alert sender has preserved supporting data, e.g., a storage dump, to which the Alert must be correlated.
10. If the Alert sender supports sending Resolution major vectors, this subvector is present in order to correlate the Alert with the Resolution major vector reporting that the Alert condition has been resolved.
11. This subvector is present when the Alert reports an error on a LAN, and the node sending the Alert is attached to the LAN.
12. This subvector is present when the Alert reports a problem with a logical link using the SDLC, LAN LLC, X.25, or ISDN protocols.

13. This subvector may be optionally included by an Alert sender in order to create a single Alert major vector that can be processed by both a non-generic Alert and a generic Alert focal point.
14. Any or all of these subvectors are present in an Alert, depending on the probable causes of the Alert condition identified by the Alert sender.
15. This subvector is present in an Alert if and only if none of the X'94', X'95', and X'96' subvectors is present.
16. Up to a total of three instances of these subvectors may be optionally included by an Alert sender, in order to create a single Alert major vector that can be processed by both a non-generic Alert and a generic Alert focal point. If either of these subvectors is present, the X'91' subvector is also present.

Alert MS Subvectors

Link Station Data (X'8C') Alert MS Subvector

This subvector transports information about the link station at the time of failure.

Link Station Data (X'8C') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Link Station Data subvector
1		Key: X'8C'
2–p		Subfields containing link station data (listed by key value below and described in detail following): X'01' Current N(S)/N(R) Counts X'02' Outstanding Frame Counts X'03' Last Data Link Control Field Received X'04' Last Data Link Control Field Sent X'05' Sequence Number Modulus X'06' Link Station State X'07' Data Link Reply Timer Expiration Count X'08' Last Received N(R) Count

Current N(S)/N(R) Counts (X'01') Link Station Data Subfield

This subfield transports the current N(S) and N(R) counts for a link station.

Current N(S)/N(R) Counts (X'01') Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Current N(S)/N(R) Counts subfield
1		Key: X'01'
2		N(S) count, in binary
3(=q)		N(R) count, in binary

Outstanding Frame Count (X'02') Link Station Data Subfield

This subfield transports the outstanding frame count.

Outstanding Frame Count (X'02') Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Outstanding Frame Count subfield
1		Key: X'02'
2(=q)		Outstanding frame count, in binary

Last Data Link Control Field Received (X'03') Link Station Data Subfield

This subfield transports the last data link control field received from the adjacent data link station before the error occurred.

Last Data Link Control Field Received (X'03') Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Last Data Link Control Field Received subfield
1		Key: X'03'
2 – 3(=q)		Last data link control field received; if the data link control field is only one byte long then byte 3 value is X'00'.

Last Data Link Control Field Sent (X'04') Link Station Data Subfield

This subfield transports the last data link control field sent to the adjacent data link station before the error occurred.

Last Data Link Control Field Sent (X'04') Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Last Data Link Control Field Sent subfield
1		Key: X'04'
2−3(=q)		Last data link control field sent; if the data link control field is only one byte long then byte 3 value is X'00'.

Sequence Number Modulus (X'05') Link Station Data Subfield

This subfield transports the modulus of the sequence number for the data link station.

Sequence Number Modulus (X'05') Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Sequence Number Modulus subfield
1		Key: X'05'
2(=q)		Modulus, in binary

Link Station State (X'06') Link Station Data Subfield

This subfield indicates busy conditions at the local or remote data link station.

Link Station State (X'06') Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Link Station State subfield
1		Key: X'06'

Link Station State (X'06') Link Station Data Subfield

Byte	Bit	Content
2(=q)		Link station states:
	0	State of the local link station: 0 local link station not busy 1 local link station busy (RNR sent)
	1	State of the remote link station: 0 remote link station not busy 1 remote link station busy (RNR received)
	2–7	Reserved

Data Link Reply Timer Expiration Count (X'07') Link Station Data Subfield

This subfield transports the number of times the Data Link Reply Timer expired.

Data Link Reply Timer Expiration Count (X'07') Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Data Link Reply Timer Expiration Count subfield
1		Key: X'07'
2–3(=q)		Count, in binary, of Data Link Reply Timer expirations

Last Received N(R) Count (X'08') Link Station Data Subfield

This subfield transports the most recently received N(R) count.

Link Station Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Last Received N(R) Count subfield
1		Key: X'08'
2(=q)		N(R) count, in binary

Basic Alert (X'91') Alert MS Subvector

This subvector transports Alert information, including an index to predefined screens.

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Basic Alert subvector
1		Key: X'91'
2		Flags:
	0	Initiation indicator: 0 Alert not directly initiated by an operator action 1 Alert initiated by an operator action
	1	Held-Alert indicator: 0 Alert was sent when the problem was detected. 1 Alert condition was detected earlier, but the Alert was not sent at that time because no session was available to send it on.
	2–3	Reserved
	4–7	Retired
3		Alert type: X'01' permanent loss of availability: a loss of availability to the end user that is not recovered from without intervention external to the reporting product X'02' temporary loss of availability: a momentary loss of availability that will probably be noticed by the end user, yet is recovered from without intervention external to the reporting product X'03' performance: a recognized measurement of response time has exceeded a predetermined threshold X'04' operator intervention required: the intervention of an operator is required to restore proper operational capability to the resource X'05'–X'09' retired X'0A' notification: a loss of availability to the end user is impending but has not yet happened X'0B'–X'0E' retired X'0F' delayed: the sender is reporting a previously detected alertable condition that prevented reporting when detected
4		General cause code: indicates the general classification and cause of the exception condition: X'01' hardware or microcode (not distinguished): the Alert condition was caused by either a hardware (machine or equipment) failure, or a microcode failure, but the specific cause cannot be determined. *Note:* Microcode may be classified as IBM Licensed Internal Code. See "Notices" at the beginning of this document for more information. X'02' software: the Alert condition was caused by a software (programming) failure or malfunction. X'03' retired

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content
		X'04'–X'05' reserved
		X'06' media (e.g., tape, disk, diskette, paper): a failure, imperfection, or defect in the media *Note:* This code is used for cases where a particular area of a tape, disk or diskette cannot be read or written but other areas are operational. It is also used for torn or jammed forms or paper. It is *not* used for cases where the medium is not present or the wrong medium, e.g., the wrong size forms, are present; these cases are indicated by X'17' (operator intervention required).
		X'07' hardware or software (not distinguished): the Alert condition was caused by either a hardware (machine or equipment) failure, or a software (programming) failure but the specific cause cannot be determined.
		X'08'–X'09' retired
		X'0A' media or hardware (not distinguished): the Alert condition was caused by either a hardware (machine or equipment) failure, or a failure, imperfection, or defect in the media, but the specific cause cannot be determined.
		X'0B' hardware: the Alert condition was caused by a hardware (machine or equipment) failure or malfunction.
		X'0C' microcode: the Alert condition was caused by a microcode failure or malfunction. *Note:* This code is *not* used for ROS chips that are packaged in field replaceable units (FRUs) or customer replaceable units (CRUs) and are serviced in the same manner as hardware logic is serviced. X'0B' (hardware) is used in those cases.
		X'0D' protocol above link level: the Alert condition was caused by an SNA protocol error above the link level. *Note:* This code point reports protocol errors that are caused by incorrect programming, for example, failure to include a BB bit on the first RU when in BETB state on a session that uses bracket protocol.
		X'0E' link-level protocol: the Alert condition was caused by a link-level protocol error. *Note:* Errors such as send/receive count errors that can be caused by missing a message because line hits do not fall into this category; they are indicated by X'0B' (hardware).
		X'0F' undetermined: the cause of the Alert condition cannot be determined.
		X'10' external facilities change or restriction: the number called is temporarily unobtainable. *Note:* This code point is used by X.21 networks.
		X'11' user: the Alert condition was caused by an incorrect action taken by a user. *Note:* Unavailability due to a device being varied offline does not fall into this category; it is indicated by X'13' (component offline).

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content	
		X'12'	system generation, customization, or installation consistency problem: the Alert condition was caused by an invalid system definition or customizing parameter, or by a mismatch between a system definition or customizing parameter and the hardware. *Note:* This code is used only in those cases that typically are not corrected by the action of the local operator.
		X'13'	component offline: the Alert condition was caused by a component being offline.
		X'14'	component busy: the Alert condition was caused by a component being busy.
		X'15'	external power failure: the Alert condition was caused by an external power failure.
		X'16'	thermal problem: the Alert condition was caused by temperature that is not within recommended specifications.
		X'17'	operator intervention required: the Alert condition was caused because action is required by an operator. *Note:* Unattended devices will always signal Alert when operator intervention is required. Attended devices will not signal Alert until the local operator has time to perform the required action. After the device-allocated time has expired for attended devices, the device has the option of sending an Alert.
		X'18'	microcode or software (not distinguished): the Alert condition was caused by either a software (programming) failure or malfunction or a microcode failure but the specific cause cannot be determined.
5–6		Specific component code: indicates the generic type of component, subcomponent, or logical resource that can be most closely related to the exception condition. The component indicated may be the generic type of the "target" or it may be a subcomponent of the target. The terms "local" and "remote" used below, refer to the perspective of the Alert originator. Defined codes are:	
		X'0001'	base processor
		X'0002'	service processor
		X'0003'	reserved
		X'0004'	main storage
		X'0005'	disk device
		X'0006'	printer
		X'0007'	card reader and/or punch
		X'0008'	tape device
		X'0009'	keyboard
		X'000A'	selector pen
		X'000B'	magnetic stripe reader
		X'000C'	display/printer
		X'000D'	display device
		X'000E'	remote product: used when a product to which the Alert generator is linked (in any form) has caused an Alert condition and the generic product type cannot be determined
		X'000F'	power supply internal to this product
		X'0010'	I/O attached controller

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content
		X'0011' communication controller scanner
		X'0012' communication link adapter
		X'0013' reserved
		X'0014' channel adapter
		X'0015' loop adapter
		X'0016' adapter for directly attaching devices
		X'0017' reserved
		X'0018' channel (direct memory access channel)
		X'0019' link: used only when common-carrier equipment cannot be distinguished from customer equipment
		X'001A' link: common-carrier equipment
		X'001B' link: customer equipment
		X'001C' loop: used only when common-carrier equipment cannot be distinguished from customer equipment
		X'001D' loop: common-carrier equipment
		X'001E' loop: customer equipment
		X'001F' X.21 link connection external to this product
		X'0020' X.25 network connection external to this product
		X'0021' local X.21 interface (DTE-DCE)
		X'0022' local X.25 interface (DTE-DCE)
		X'0023' local modem
		X'0024' remote modem
		X'0025' local modem interface (DTE-DCE)
		X'0026' remote modem interface (DTE-DCE)
		X'0027' local modem link monitor
		X'0028' remote modem link monitor
		X'0029' local modem link monitor interface
		X'002A' remote modem link monitor interface
		X'002B'-X'0031' reserved
		X'0032' remote modem or modem interface or remote product
		X'0033' transmission medium or remote modem
		X'0034' SDLC data link control component
		X'0035' BSC data link control component
		X'0036' start/stop data link control component
		X'0037'-X'0043' reserved
		X'0044' cluster controller or device
		X'0045' local link monitor or modem interface
		X'0046' reserved
		X'0047' card reader/punch or display/printer
		X'0048' controller application program
		X'0049' keyboard or display
		X'004A' storage control unit
		X'004B' storage control unit or storage control unit channel
		X'004C' storage control unit or controller
		X'004D' control unit (other than storage control unit)
		X'004E'-X'0051' reserved

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content
		X'0052' maintenance device
		X'0053' maintenance device interface
		X'0054' reserved
		X'0055' control program
		X'0056' application subsystem on top of control program
		X'0057' telecommunication access method
		X'0058' application program (other than application subsystem)
		X'0059' communication controller program
		X'005A'–X'005F' reserved
		X'0060' X.25 network interface: DCE to first interface node in X.25 network
		X'0061' disk device with nonremovable media
		X'0062' disk device with removable media
		X'0063' control tailed modem
		X'0064' reserved
		X'0065' remote tailed modem
		X'0066' remote tailed modem interface
		X'0067' sensor I/O unit
		X'0068' magnetic stripe reader/encoder
		X'0069' check (bank) reader
		X'006A' document feed mechanism
		X'006B' coin feed mechanism
		X'006C' envelope depository
		X'006D' timer adapter
		X'006E' encryption/decryption adapter
		X'006F' outboard, user programmable processor
		X'0070' cable connecting local device to local adapter
		X'0071'–X'007F' reserved
		X'0080' token-ring LAN error
		X'0081' Carrier-Sense-Multiple-Access/Collision-Deletion (CSMA/CD) LAN error
		X'0082'–X'00FE' reserved
		X'00FF' undetermined (the problem cannot be isolated to one of the above generic component types)
7–8		Alert description code: a code that provides an index to predefined text that explains the condition that caused the Alert *Note:* This field is product dependent.
9–10		User Action Code: a code that provides an index to predefined screens that can include predefined text and variable fields for MS User Action Qualifier subvectors *Note:* This field is product dependent.
11–12		Detail text reference code: a code that provides an index to predefined screens that can include predefined text and variable fields for MS Detail Qualifier subvectors *Note:* This field is product dependent.
13(=p)		Retired

Generic Alert Data (X'92') Alert MS Subvector

This subvector transports Alert information in the form of code points that correspond to strings of text stored at the Alert receiver. It also transports an Alert ID Number that uniquely identifies a particular Alert.

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Generic Alert Data subvector
1		Key: X'92'
2–3		Flags:
	0	Initiation indicator: 0 Alert not directly initiated by an operator action 1 Alert initiated by an operator action
	1	Held Alert indicator: 0 Alert was sent when the problem was detected. 1 Alert condition was detected earlier, but the Alert was not sent at that time because no session was available to send it on.
	2	Delayed Alert indicator: 0 Sender is not reporting a previously detected Alert condition that prevented reporting when detected. 1 Sender is reporting a previously detected Alert condition that prevented reporting when detected. *Note:* If the delayed Alert indicator is set to 1, the held Alert indicator is also set to 1.
	3	Reserved
	4	SNMP Trap indicator: 0 Alert was not built based on an SNMP Trap. 1 Alert was built based on an SNMP Trap.
	5–15	Reserved
4		Alert type: a hexadecimal value indicating the severity of the Alert condition: X'01' permanent loss of availability: a loss of availability to the end user that is not recovered from without intervention external to the reporting product X'02' temporary loss of availability: a momentary loss of availability that will probably be noticed by the end user, yet is recovered from without intervention external to the reporting product X'03' performance: performance below what is considered an acceptable level X'04'–X'0F' reserved *Note:* These values are reserved to avoid confusion with the corresponding basic Alert types. They should not be used in a generic Alert. X'10' permanently affected resource: the originator of this Alert has determined that the target resource is lost because of a persistent error in a resource other than the target

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content
		X'11' impending problem: a loss of availability to the end user impending but that has not yet happened
		X'12' unknown: the severity of the Alert condition not assessable
		X'13' retired
		X'14' bypassed: the loss of availability to the end user has been circumvented to allow the resource or an alternate resource to be used. The original problem still exists and the recovery may or may not be noticed by the end user. The recovery may be accomplished by intervention, internal or external to the reporting product.
		X'15' redundancy lost: redundant hardware and/or software provided to ensure continued operation in the event of a failure has experienced a malfunction. As a result, failure of the remaining operational hardware and/or software will result in a loss of corresponding services.
5–6		Alert Description Code: A code point that provides an index to predefined text describing the Alert condition. An Alert receiver has two options for selecting text to display. It can display the English text documented with each code point, or its national language equivalent; or, for a presentation to an operator of a lower skill level, it can choose the following simpler text (shown all capitalized), or its national language equivalent, based only on the first digit of the code point: X'1xxx' HARDWARE X'2xxx' SOFTWARE X'3xxx' COMMUNICATIONS X'4xxx' PERFORMANCE X'5xxx' CONGESTION X'6xxx' MICROCODE X'7xxx' OPERATOR X'8xxx' SPECIFICATION X'9xxx' INTERVENTION REQUIRED X'Axxx' retired X'Bxxx' NOTIFICATION X'Cxxx' SECURITY X'Exxx' RESERVED X'Fxxx' UNDETERMINED Specific defined codes and the corresponding displayed text (shown all capitalized) are listed below. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules). X'1000' EQUIPMENT MALFUNCTION: An internal machine error has occurred X'1001' CONTROL UNIT MALFUNCTION

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content		
			X'1002'	DEVICE ERROR *Note:* This code point is used only if the Alert sender is unable to determine the nature of the affected device.
			X'1003'	CPC HARDWARE FAILURE: A hardware failure has occurred in a central processor complex (CPC).
			X'1004'	TIME OF DAY CLOCK FAILURE: A failure in a mechanism which keeps time.
			X'1005'	BACK-UP RESOURCE FAILURE: A failure on a resource which has been designated as a back-up. The back-up capability has been lost.
			X'1006'	OPTICAL SYSTEM BUS FAILURE
			X'100A'	SERVICE PROCESSOR FAILURE: A failure has occurred in the maintenance, service, and support processor; sometimes called a processor controller.
			X'1010'	ADAPTER ERROR: A hardware error has occurred in an adapter, making it inoperable
		X'1100'	INPUT DEVICE ERROR	
			X'1101'	MICR READER/SORTER ERROR: An error has been detected in a magnetic ink character recognition reader/sorter
		X'1200'	OUTPUT DEVICE ERROR	
			X'1201'	PRINTER ERROR
			X'1202'	PRINTER CASSETTE ERROR
		X'1300'	INPUT/OUTPUT DEVICE ERROR	
			X'1301'	LOCAL CONSOLE ERROR
			X'1302'	REMOTE CONSOLE ERROR *Note:* "Local" and "remote" are defined with respect to the system with which the console communicates.
			X'1311'	DISK FAILURE: A disk unit is no longer usable
			X'1312'	DISK OPERATION ERROR: A disk operation has failed, but the unit may still be usable
			X'1321'	DISKETTE DEVICE FAILURE: A diskette unit is no longer usable
			X'1322'	DISKETTE OPERATION ERROR: A diskette operation has failed, but the unit may still be usable
			X'1331'	TAPE DRIVE FAILURE: A tape drive is no longer usable
			X'1332'	TAPE OPERATION ERROR: A tape operation has failed, but the tape drive may still be usable
			X'1340'	DASD MIRRORING LOST: Access to one or more of the DASD units involved in mirrored (redundant) DASD has been lost. Mirroring protection has been lost and failure of the remaining DASD unit or units will result in an operational failure.

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
		X'1400'	LOSS OF ELECTRICAL POWER: A source of electrical power, internal or external, has been lost
		X'1401'	LOSS OF CHANNEL ADAPTER ELECTRICAL POWER
		X'1402'	LOSS OF LINE ADAPTER ELECTRICAL POWER
		X'1403'	LOSS OF LIC UNIT ELECTRICAL POWER
		X'1404'	LOSS OF MOSS ELECTRICAL POWER
		X'1410'	LOSS OF EXTERNAL ELECTRICAL POWER
		X'1411'	POWER OFF DETECTED: A network component has detected a notification signal announcing that the power of another component was lost or turned off
		X'1412'	LOSS OF ALL SOURCES OF ELECTRICAL POWER
		X'1413'	AN ELECTRICAL POWER SOURCE LOST
		X'1500'	LOSS OF EQUIPMENT COOLING OR HEATING: A loss of equipment cooling or heating has occurred *Note:* If loss of power has not been ruled out as a cause for the loss of heating or cooling, then X'1400' (LOSS OF ELECTRICAL POWER) should be sent instead of this code point.
		X'1501'	LOSS OF EQUIPMENT COOLING
		X'1502'	LOSS OF MOSS EQUIPMENT COOLING
		X'1600'	SUBSYSTEM FAILURE: A failure in a set of components that jointly provide a specified function; typically a subsystem includes a controller, one or more interface adapters, physical connection media, and attached devices
		X'1601'	STORAGE SUBSYSTEM FAILURE: A failure in a subsystem that supports locally-attached storage devices, such as hard disk (DASD), diskette, and tape
		X'1602'	WORKSTATION SUBSYSTEM FAILURE: A failure in a subsystem that supports workstations directly attached to a node, i.e., workstations not attached via telecommunications links
		X'1603'	COMMUNICATIONS SUBSYSTEM FAILURE: A failure in a subsystem that supports communication over telecommunications links; these links may be implemented via leased telephone lines, an X.25 network, a token-ring LAN, or otherwise
		X'1604'	XID NEGOTIATION TERMINATED
		X'1605'	WRONG LINK MODE SETTING COMMAND RECEIVED
		X'1606'	TOPOLOGY CAPACITY EXCEEDED
		X'1607'	TOPOLOGY PROTOCOL ERROR

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
			X'1608' SERVICE SUBSYSTEM FAILURE: A failure in a subsystem that performs IPL functions, maintenance functions, machine initialization or recovery, and provides problem determination capabilities.
			X'1609' SUBSYSTEM JOB TERMINATED
			X'160A' DASD SUBSYSTEM DATA CHECK
			X'1611' IMPENDING STORAGE SUBSYSTEM FAILURE
			X'1612' IMPENDING WORKSTATION SUBSYSTEM FAILURE
			X'1613' IMPENDING COMM SUBSYSTEM FAILURE
			X'1620' VOICE RESPONSE UNIT SUBSYSTEM FAILURE
		X'2000'	SOFTWARE PROGRAM ABNORMALLY TERMINATED: A software program has abnormally terminated due to some unrecoverable error condition *Note:* See also code point X'6000' (MICROCODE PROGRAM ABNORMALLY TERMINATED).
			X'2001' CPC ENTERED HARD WAIT: A failure has occurred which resulted in all central processing units (CPU's) of a central processing complex (CPC) entering into a wait state with interrupts disabled.
			X'2002' SNAPSHOT TABLE PROCESSING FAILURE
			X'2003' REQUESTED SOFTWARE TRAP OCCURRED: A program has stopped execution due to previously setup controls.
		X'2100'	SOFTWARE PROGRAM ERROR: An error has occurred within a software program that has caused incorrect results, but the program has not terminated *Note:* See also code point X'6100' (MICROCODE PROGRAM ERROR).
			X'2101' PROGRAM PROCEDURE IS INCORRECT: A set of instructions which originated in a computer program and are intended to direct the operation of a device are incorrect.
			X'2102' DISTRIBUTED PROCESS FAILED: Some software component for a distributed unit of work failed. This is an internal error in a software component.
			X'2103' REQUIRED SOFTWARE ATTEMPTED TERMINATION
			X'2104' CANNOT LOAD TASK: A processor could not load a task into memory.
			X'2105' PROGRAM PARAMETER IS INCORRECT
			X'2106' FILE ACCESS ERROR
			X'2107' DATABASE UNAVAILABLE
			X'2108' DISK ACCESS ATTEMPT FAILED
			X'2109' NODE TO NODE COMMUNICATIONS NOT POSSIBLE

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
			X'210A' SOFTWARE OPERATION NOT STARTED
			X'210B' SOFTWARE PROGRAM WILL NOT TERMINATE
			X'210C' RESOURCE RECOVERY FAILURE
			X'210D' RECOVERY CANCELLED FOR ERROR DURING IPL
			X'210E' ERRORS CANNOT BE LOGGED
		X'2200'	SOFTWARE OPERATION FAILURE: A hardware component error has caused an operation on a file, volume, or some other unit of stored data, to fail.
			X'2201' POSSIBLE FILE CORRUPTION
			X'2202' FILE SERVER VOLUME INACCESSIBLE
			X'2203' FILE DIRECTORY MAY BE CORRUPTED
			X'2204' FILE ALLOCATION TABLE MAY BE CORRUPT
			X'2205' SYSTEM FILE UNUSABLE
		X'3000'	COMMUNICATION PROTOCOL ERROR: An architecturally defined communication protocol has been violated *Note:* This code point is not used if one that identifies the particular protocol involved is available.
			X'3001' DIRECTORY SERVICES PROTOCOL ERROR
			X'3002' SESSION SERVICES PROTOCOL ERROR
		X'3100'	SNA PROTOCOL ERROR: An SNA protocol has been violated
			X'3110' XID PROTOCOL ERROR: A protocol error related to XID exchange has been detected
			X'3111' INVALID XID RECEIVED: An XID has been received that contains either a format error or a value unacceptable to the receiver
			X'3112' SNA SESSION SETUP FAILURE: Session setup or session termination failed.
			X'3113' CP-CP SESSION FAILURE
			X'3114' MANAGEMENT SERVICES PROTOCOL ERROR: Management Services received a message which cannot be processed because it detected a protocol violation.
			X'3115' LU6.2 RECEIVED NEGATIVE BIND RESPONSE
			X'3116' LU6.2 SENT NEGATIVE BIND RESPONSE
			X'3117' LU6.2 SESSION ACTIVATION REJECTED
			X'3118' LU6.2 UNBIND REQUEST SENT
			X'3119' LU6.2 UNBIND REQUEST RECEIVED
			X'3120' LU6.2 SESSION FAILURE
		X'3200'	LAN ERROR: An error has been detected on a local area network
			X'3203' LOOP ERROR: An error has been detected on a communication loop
			X'3204' LOOP OPEN
			X'3205' LOOP ADAPTER INOPERATIVE

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
		X'3210'	INITIALIZATION FAILURE: A LAN adapter has detected a problem while being initialized
		X'3211'	OPEN FAILURE: A LAN adapter has detected a problem during the insertion process; the insertion process did not complete
		X'3212'	WIRE FAULT: An error condition caused by a break in the wires or a short between the wires (or shield) in a segment of cable has been detected *Note:* The term "wire fault" applies only to failures on the lobes of a token-ring LAN.
		X'3213'	AUTO REMOVAL: A station's adapter has left a LAN token-ring or bus as part of an automatic-recovery process. For token-rings, this process is known as the beacon automatic-recovery process.
		X'3214'	REMOVE ADAPTER COMMAND RECEIVED: The reporting station received a Remove Adapter command from a LAN manager and, as a result, left the LAN.
		X'3215'	TOKEN-RING INOPERATIVE: After the onset of beaconing, a token-ring attempted and failed auto recovery; the token-ring has been beaconing for more than 52 seconds, and is still beaconing.
		X'3216'	TOKEN-RING TEMPORARY ERROR: The token-ring was in a beaconing state for less than 52 seconds and then recovered; the Alert sender has no knowledge whether a station was removed to bypass the fault or the fault was temporary.
		X'3218'	BACK-UP PATH INOPERATIVE: The back-up path of a subsystem has failed leaving only the main path operational.
		X'3219'	MAIN PATH WRAPPED TO BACK-UP PATH: The main path has failed and the back-up path is being used to continue operation.
		X'321A'	LAN LLC PROTOCOL ERROR
		X'321B'	DUPLICATE MAC ADDRESS ON TOKEN-RING
		X'3220'	CSMA/CD BUS INOPERATIVE
		X'3221'	CSMA/CD LAN COMMUNICATIONS LOST: A station is unable to communicate over a CSMA/CD LAN *Note:* The problem may be local to the Alert sender, or it may apply to the entire bus to which the Alert sender is attached.
		X'3230'	LAN MANAGEMENT DATA LOST: A LAN management server has become congested or incapacitated so it cannot handle its data input. As a result, management data from LAN stations has been discarded.
		X'3231'	MONITORED STATION LEFT LAN: A monitored station is one that an operator at the LAN Manager has designated as a critical resource. The station is monitored for its disappearance from the LAN.

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
		X'3238'	FDDI CONNECTION ERROR
		X'3240'	TOKEN-BUS COMMUNICATIONS LOST
		X'3250'	NETWORK ADDRESSING CONFLICT
		X'3251'	LAN COMMUNICATIONS LOST
		X'3252'	LAN SEGMENT NUMBER MISMATCH: The Ring Parameter Server has a segment number for the ring that does not match the segment number that the bridge has for the ring.
		X'3260'	BACK-UP PATH STATUS CHANGE
		X'3261'	PATH WRAP STATUS CHANGE
		X'3262'	FORCE REMOVE IGNORED
		X'3263'	ACCESS UNIT CORRECTED TOKEN-RING ERROR
		X'3270'	UNABLE TO CONTACT DOMAIN CONTROLLER
		X'3300'	LINK ERROR: An error has occurred on a network communication link *Note:* This default code point covers all of the following: Connections between subarea nodes, connections between subarea nodes and peripheral nodes, connections between peripheral nodes, and connections between peripheral nodes and the devices that are hierarchically below them. If the link is implemented by a local area network, one of the X'32xx' code points is used instead. Specific ISDN-related errors should be reported with code points in the X'34xx' range.
		X'3301'	REMOTE SUPPORT FACILITY LINK ERROR: An error has occurred on a communication link with the IBM Remote Support Facility
		X'3302'	UNABLE TO COMMUNICATE WITH DEVICE
		X'3303'	UNABLE TO COMMUNICATE WITH PRINTER
		X'3304'	UNABLE TO COMMUNICATE WITH DISPLAY
		X'3305'	UNABLE TO COMMUNICATE WITH REMOTE NODE
		X'3306'	INBOUND CONNECTION ATTEMPT FAILED: A failure occurred when another node attempted to establish a connection with the reporting node.
		X'3307'	OUTBOUND CONNECTION ATTEMPT FAILED: A failure occurred when the reporting node attempted to establish a connection with another node.
		X'3310'	X.21 ERROR: An error has been detected on a communication link operating according to the X.21 protocols.
		X'3311'	X.21 ERROR – SNA SECONDARY: An error has prevented an SNA secondary link station from establishing an X.21 connection
		X'3312'	X.21 ERROR – SNA PRIMARY: An error has prevented an SNA primary link station from establishing an X.21 connection
		X'3313'	X.21 CONNECTION CLEARED

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
			X'3320' X.25 ERROR: An error has been detected on a communication link operating according to the X.25 protocols
			X'3330' MANAGEMENT SERVER REPORTING LINK ERROR: A LAN manager has detected an error on one of its reporting links with a LAN management server
		X'3400'	ISDN ERROR: An error has occurred on an Integrated Services Digital Network (ISDN) connection
			X'3401' D-CHANNEL ISDN ERROR
			X'3402' B-CHANNEL ISDN ERROR
			X'3403' R-INTERFACE ISDN ERROR
			X'3404' ISDN PHYSICAL LAYER ERROR
			X'3405' ISDN PHYSICAL LAYER ACTIVATION ERROR
			X'3406' ISDN CALL CONTROL ERROR
			X'3407' ISDN CALL TERMINATED UNEXPECTEDLY
		X'3500'	LOCAL CONNECTION ERROR: An error has occurred on a local channel connection
		X'3600'	LINK CONNECTION ERROR *Note:* A *link connection* includes the interface between the DTE and the DCE, any protocol used to communicate between the DTE and the DCE (such as LPDA, the IBM Command Set, the AT Command Set, etc.) and DCE provided information about the link.
			X'3601' NO LPDA RESPONSE RECEIVED
			X'3602' BAD FCS IN LPDA RESPONSE
			X'3603' INTERFACE ERROR DURING LPDA
			X'3604' CONFIGURATION MISMATCH
			X'3605' MODEM CONFIGURATION ERROR
			X'3606' DSU/CSU CONFIGURATION ERROR
			X'3607' MODEM ERROR
			X'3608' DSU/CSU ERROR
			X'3609' EQUIPMENT INCOMPATIBILITY
			X'360A' MODEM REINITIALIZED
			X'360B' DSU/CSU REINITIALIZED
			X'360C' MODEM FAILURE DETECTED
			X'360D' DSU/CSU FAILURE DETECTED
			X'360E' MODEM SPEEDS MISMATCH
			X'360F' TEST IN PROGRESS
			X'3610' STREAMING DETECTED
			X'3611' DTR DROPPED
			X'3612' EXTERNAL CLOCK NOT RUNNING
			X'3613' BAD LINE QUALITY
			X'3614' RLSD OFF DETECTED
			X'3615' EXCESSIVE IMPULSE HITS DETECTED
			X'3616' EXCESSIVE BIPOLAR CODE ERRORS
			X'3617' DCE INTERFACE ERROR

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
		X'3618'	UNEXPECTED RECEIVED CARRIER DETECTED
		X'3619'	NO LINE SIGNAL
		X'361A'	OUT OF FRAME RECEIVED BY LOCAL DSU/CSU
		X'361B'	OUT OF FRAME RECEIVED BY REMOTE DSU/CSU
		X'361C'	OUT OF SERVICE RECEIVED BY LOCAL DSU/CSU
		X'361D'	OUT OF SERVICE RECEIVD BY REMOTE DSU/CSU
		X'361E'	DDS LOOP BACK DETECTED BY LOCAL DSU/CSU
		X'361F'	DDS LOOP BACK DETECTED BY REMOTE DSU/CSU
		X'3620'	FORBIDDEN CALL – CALL REJECTED
		X'3621'	DELAYED CALL – CALL REJECTED
		X'3622'	LOCAL MODEM AUTO-CALL TIME-OUT
		X'3623'	LOCAL DTE AUTO-CALL TIME-OUT
		X'3624'	CALL FAILURE – CALLED NUMBER BUSY
		X'3625'	CALL FAILURE – NO ANSWER
		X'3626'	CALL FAILURE – ANSWER TONE NOT DETECTED
		X'3627'	CALL COLLISION
		X'3628'	INVALID/UNSUPPORTED MODEM COMMAND
		X'3629'	NETWORK PHYSICAL LAYER ERROR
		X'4000'	PERFORMANCE DEGRADED: Service or response time exceeds what is considered an acceptable level
		X'4001'	EXCESSIVE TOKEN-RING ERRORS: Soft errors are occurring on a token ring at an excessive rate *Note:* The token-ring LAN term "soft error" is defined as an intermittent error on a network that causes data to have to be transmitted more than once to be received. The condition identified by this code point is detected by Ring Error Monitor (REM); REM also provides a fault domain to indicate the location of most of the soft errors.
		X'4003'	EXCESSIVE CONTROL UNIT ERRORS
		X'4010'	ERROR TO TRAFFIC RATIO EXCEEDED: A computed ratio of errors to total traffic has exceeded a specified threshold
		X'4011'	THRESHOLD HAS BEEN EXCEEDED
		X'4012'	THRESHOLD HAS BEEN REACHED
		X'4021'	EXCESSIVE STORAGE SUBSYSTEM ERRORS
		X'4022'	EXCESSIVE WORKSTATION SUBSYSTEM ERRORS
		X'4023'	EXCESSIVE COMMUNICATIONS SUBSYST ERRORS
		X'5000'	CONGESTION: A system or network component has either reached its capacity or is approaching it
		X'5001'	NETWORK CONGESTION: There is excessive traffic in the network

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content		
			X'5002'	RESOURCE NEARING CAPACITY: A resource is approaching its capacity; it is still usable, but it threatens to become unusable unless corrective action is taken
			X'5003'	CAPACITY EXCEEDED: A request has been received by a component that, if granted, would require more resources than the component has available to it
			X'5004'	OUT OF RESOURCES: A component has no more resources available; it is no longer able to function
			X'5005'	WORKSTATION LIMIT EXCEEDED: More workstations than the workstation subsystem supports being powered on have attempted to power on simultaneously
			X'5006'	ALMOST OUT OF RESOURCES
			X'500C'	VIRTUAL ROUTE HELD TIME LIMIT REACHED: A congested virtual route has been in the held state (i.e. a virtual route pacing response has not been received) for longer than a defined time limit.
			X'500D'	VR PACING WITHHELD TIME LIMIT REACHED: A congested virtual route endpoint has not sent a virtual route pacing response for longer than a defined time limit.
			X'500E'	HELD VR DEACTIVATION TIME LIMIT REACHED: A congested virtual route has been deactivated since it was in the held state (i.e. a virtual route pacing response had not been received) for longer than a defined time limit.
			X'500F'	VIRTUAL ROUTE TRANSMIT QUEUE OVERRUN: The size of a virtual route transmit queue has exceeded a defined threshold for longer than a defined time limit.
			X'5010'	COMMUNICATIONS UNDERRUN: A link station element is unable to write data to an adapter rapidly enough
			X'5011'	COMMUNICATIONS OVERRUN: A MAC service user is unable to read data from an adapter rapidly enough
			X'5012'	RECEIVE QUEUE OVERRUN: A receive queue in a node is unable to receive data from a link station in the node rapidly enough
			X'5013'	SLOWDOWN: A device has exhausted its supply of available buffers and has stopped accepting inbound data until it can handle all outbound requests
			X'5020'	FILE NEEDS REORGANIZATION A file is approaching its capacity, and will soon be unusable unless it is reorganized
		X'5100'	CONFIGURABLE CAPACITY LIMIT REACHED	
			X'5101'	FILE DIRECTORY ENTRY LIMIT EXCEEDED
			X'5102'	FILE LOCK THRESHOLD REACHED
			X'5103'	RECORD LOCK THRESHOLD REACHED
			X'5104'	MEMORY THRESHOLD REACHED
			X'5105'	MEMORY USAGE THRESHOLD REACHED
			X'5106'	DIRECTORY SEARCH THRESHOLD REACHED

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
		X'5107'	ERROR LOG LIMIT REACHED
		X'5108'	AUDIT LOG LIMIT REACHED
		X'5109'	ERROR LOG FILE ALMOST FULL
		X'510A'	AUDIT LOG FILE ALMOST FULL
		X'510B'	ACCOUNT LIMIT REACHED
		X'510C'	FILE SERVER ERROR LIMIT REACHED
		X'510D'	NETWORK I/O ERROR LIMIT REACHED
		X'510E'	SYSTEM RESOURCE LIMIT REACHED
		X'510F'	REQUESTER RESOURCE LIMIT REACHED
		X'5110'	MAXIMUM STORAGE LIMIT EXCEEDED
		X'5111'	NETWORK CONTROL BLOCK LIMIT REACHED
		X'5112'	THREAD LIMIT REACHED
		X'6000'	MICROCODE PROGRAM ABNORMALLY TERMINATED: A microcode program has abnormally terminated due to some unrecoverable error condition *Note:* See also code point X'2000' (SOFTWARE PROGRAM ABNORMALLY TERMINATED).
		X'6100'	MICROCODE PROGRAM ERROR: An error has occurred within a microcode program that has caused incorrect results, but the program was not terminated *Note:* See also code point X'2100' (SOFTWARE PROGRAM ERROR).
		X'7000'	OPERATOR PROCEDURAL ERROR: An operator has attempted to initiate an incorrect procedure, or has initiated a procedure incorrectly
		X'7001'	RESOURCES NOT ACTIVE: An operator has deactivated, or failed to activate, resources required for a requested operation
		X'8000'	CONFIGURATION OR CUSTOMIZATION ERROR: A system or device generation or customization parameter has been specified incorrectly, or is inconsistent with the actual configuration
		X'8001'	CUSTOMIZATION IMAGE WARNING: A customization image parameter is incorrect and has been replaced by a valid value.
		X'8002'	PASSWORD ENCRYPTION ERROR
		X'8003'	TELEPHONE NUMBER NOT STORED
		X'8004'	SESSION SET-UP PARAMETER MISMATCH
		X'8005'	REQUIRED SOFTWARE FEATURE NOT INSTALLED

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content
		X'9000' OPERATOR INTERVENTION REQUIRED: A condition has occurred indicating that operator intervention is required, and an operator has not responded *Note:* The X'90xx' code points are used only for conditions that (1) require *on-site* intervention, and (2) can be resolved by personnel that do not possess a high level of technical skill.
		X'9001' PRINTER RIBBON JAM
		X'9002' PAPER JAM
		X'9003' BILL/DOCUMENT JAM
		X'9004' COIN JAM
		X'9005' FILM/VIDEOTAPE NOT MOVING
		X'9010' DEVICE NOT READY: A device has indicated that it is not ready for use, due to an unspecified intervention-required condition
		X'9011' PRINTER NOT READY: A printer has indicated that it is not ready for use, due to an unspecified intervention-required condition
		X'9030' OUT OF FOCUS
		X'9031' SERVICE DOOR OPENED *Note:* Security and/or safety considerations may preclude normal operation until the door is closed.
		X'9032' DEVICE NOT CALIBRATED
		X'9100' STOCK LOW: The stock of some required material (e.g., paper, ink, coins) is low, but is not yet exhausted
		X'9101' LOW ON INK
		X'9102' LOW ON PAPER
		X'9103' LOW ON BILLS/DOCUMENTS
		X'9104' LOW ON COINS
		X'9105' LOW ON FILM/VIDEOTAPE
		X'9106' LOW ON TONER
		X'9107' LOW ON FUSER OIL
		X'9108' LOW ON STAPLES
		X'9109' DISKETTE FILE NEARLY FULL: An output file being written to a diskette is almost full. Continued operation may result in the file becoming full, which may result in abnormal operation of the device.
		X'910A' LOW ON ENVELOPES
		X'9200' STOCK EXHAUSTED: The stock of some required material (e.g., paper, ink, coins) has been exhausted
		X'9201' OUT OF INK
		X'9202' OUT OF PAPER
		X'9203' OUT OF BILLS/DOCUMENTS
		X'9204' OUT OF COINS
		X'9205' OUT OF FILM/VIDEOTAPE

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content
		X'9206' OUT OF TONER
		X'9207' OUT OF FUSER OIL
		X'9208' OUT OF STAPLES
		X'920A' OUT OF ENVELOPES
		X'9300' DEPOSITORY FULL: A depository has become full, and thus cannot receive any more deposits
		X'9301' DEPOSITORY APPROACHING CAPACITY: A depository is nearing its capacity; if it is not emptied shortly, it will become completely filled
		X'A000'–X'AFFF' retired
		X'B000' OPERATOR NOTIFICATION: Problem-related information is being conveyed to a network operator *Note:* A X'Bxxx' code point is used only if no more specific one is available.
		X'B001' MAINTENANCE PROCEDURE: A resource has been taken offline for maintenance *Note:* This code point is used to notify a network operator about a disruptive maintenance procedure that was invoked locally; otherwise, there would be an unexplained loss of a resource.
		X'B002' OPERATOR TOOK PRINTER OFFLINE
		X'B003' LAN BRIDGE TAKEN OFFLINE
		X'B004' RESOURCES REQUIRE ACTIVATION: Some resources are not active. The operator must activate these resources to make the system fully operational.
		X'B005' SERVICE SUBSYSTEM TAKEN OFFLINE
		X'B006' LINE ADAPTER DISCONNECTED
		X'B007' TOKEN RING ADAPTER DISCONNECTED
		X'B008' HIGH SPEED LINE ADAPTER DISCONNECTED
		X'B009' CHANNEL ADAPTER MAINTENANCE PROCEDURE: A channel adapter has been taken offline for maintenance.
		X'B00A' TIMED IPL TO OCCUR SOON: An automatic initial program load (IPL) of a machine has been scheduled at a date and time that will occur soon (e.g. in 30 or 60 minutes).
		X'B00B' CSMA/CD ADAPTER DISCONNECTED
		X'B00C' SNMP RESOURCE PROBLEM: A problem has occurred with an SNMP resource. This code point is used when the problem could not be specifically identified.
		X'C000' SECURITY EVENT: An event indicative of a possible security exposure has been detected
		X'C001' INVALID REPORTING LINK PASSWORD
		X'C002' UNAUTHORIZED LAN INSERTION ATTEMPTED

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content	
		X'C003'	UNAUTHORIZED NETWORK CHANGE ATTEMPTED: An end node CP, without authorization, has attempted to delete a resource.
		X'C004'	UNAUTHORIZED ACCESS ATTEMPTED: BIND received from an end node that this network node does not serve.
		X'C005'	UNKNOWN OSI MANAGEMENT SERVICES REQUEST: An OSI system is attempting to solicit management services from another system without being properly identified.
		X'C006'	INVALID PASSWORD
		X'C007'	UNAUTHORIZED ACCESS ATTEMPTED
		X'C008'	ACCESS TO DOMAIN CONTROLLER DENIED
		X'C009'	SYSTEM NOT SECURE
		X'E000'–X'EFFF'	Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.
		X'FE00'	UNDETERMINED ERROR: An error condition has occurred that cannot be related to a more specific error category
		X'FE01'	RESOURCE UNAVAILABLE: A resource has become unavailable, but the Alert sender has no indication of why this has happened *Note:* This code point should be used only if the Alert sender cannot determine, with any degree of certainty, that another Alert description code is applicable to the event being reported.

Generic Alert Data (X'92') Alert MS Subvector

Byte	Bit	Content
7 – 10(=p)		Alert ID number: A 4-byte hexadecimal value computed as follows: Stage 1: Assemble (in order) the following input from the Alert major vector: • Alert Type • Alert Description Code code point • All Probable Causes code points, in order • The delimiter X'FFFF' • All User Causes code points, in order, if any are present • The delimiter X'FFFF' • All Install Causes code points, in order, if any are present • The delimiter X'FFFF' • All Failure Causes code points, in order, if any are present Stage 2: Apply to this input the 32-bit CRC algorithm: $$\frac{x^{32}I(x) + x^{k}L(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$ where: L(x) $\sum_{i=0}^{31} x^i$ G(x) $\sum_{i} x^i$ for i = 32, 26, 23, 22, 16, 12, 11, 10, 8, 7, 5, 4, 2, 1, 0 I(x) The polynomial represented by the input to the CRC algorithm (with the convention that the first bit of the input represents the coefficient of this polynomial's *highest*-order term) k number of bits in the input polynomial I(x) The Alert ID number is the *complement* of the remainder polynomial R(x) (sometimes represented as *Alert ID* = $\overline{R(x)}$). The reader should remember that all arithmetic is modulo 2, and that the degree of the remainder polynomial, R(x), is less than 32.

Probable Causes (X'93') Alert MS Subvector

This subvector contains one or more code points denoting probable causes of the Alert condition. The probable causes appear in order of decreasing probability.

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Probable Causes subvector
1		Key: (X'93')

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
2−p		One or more two-byte probable cause code points, defined below. Each code point provides an index to predefined text denoting the probable cause. An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point (not indented) above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are listed below. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules). X'0000' PROCESSOR: The equipment used to interpret and process programmed instructions. These instructions may be programmed in either software or microcode X'0001' MOSS (Maintenance and Operation Subsystem): A service processor for a communication controller X'0002' VECTOR PROCESSOR: The vector processing element associated with a central processing unit (CPU) X'0003' PROCESSOR SWITCH: A component within a hardware product used to switch buses and the resources attached to them among processors X'0004' CONTROL PANEL X'0005' SYSTEM I/O BUS X'000A' SERVICE PROCESSOR: The maintenance, service, and support processor; sometimes called a processor controller. X'0010' LAN MANAGER: A network component responsible for managing a local area network X'0011' PRINTER SERVER: A network component that controls the operation of a printer *Note:* In the current implementation, the printer server is a PC that stands between a printer and the host applications that communicate with it. X'0012' FILE SERVER X'0013' OPTICAL SYSTEM BUS CONTROLLER X'0030' SYSTEM MICROCODE: The specific microcode was not identified. X'0031' SYSTEM STORAGE MICROCODE *Note:* See also code point X'0421' (STORAGE CONTROLLER MICROCODE) X'0032' SYSTEM DISPLAY MICROCODE *Note:* See also code point X'0422' (WORKSTATION CONTROLLER MICROCODE)

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'0033'	SYSTEM COMMUNICATION MICROCODE *Note:* See also code point X'0423' (COMM SUB-SYSTEM CONTROLLER MICROCODE)
			X'0034'	SYSTEM PRINTER MICROCODE *Note:* See also code point X'0422' (WORKSTATION CONTROLLER MICROCODE)
			X'0040'	INITIAL PROGRAM LOAD
			X'0050'	TERMINAL EMULATOR SUBSYSTEM MICRO-CODE
		X'0100'	STORAGE: The random access memory (RAM) or read only memory (ROM) accessible by a processor and by peripheral devices	
			X'0101'	MAIN STORAGE: Storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing
			X'0102'	AUXILIARY STORAGE: Storage that can not be directly addressed by a processor, such as external or secondary storage
			X'0103'	NON-VOLATILE STORAGE
			X'0104'	EXPANDED STORAGE: A specific type of auxiliary storage used for data and program paging
			X'0105'	CRITICAL SYSTEM STORAGE: A specific portion of main storage used only by the machine
			X'0106'	MEMORY
			X'0107'	TRANSMIT/RECEIVE BUFFERS
			X'0108'	DASD CACHE
		X'0200'	POWER SUBSYSTEM: The subsystem within a hardware product that provides electrical power to the different components within the product that require it	
			X'0201'	INTERNAL POWER UNIT: An element of the power subsystem providing electrical power to a specific component
			X'0202'	INTERNAL POWER CONTROL UNIT: An element of the power subsystem that controls the internal power units
			X'0203'	POWER CABLE
			X'0204'	POWER CORD
			X'0205'	POWER SUBSYSTEM PROCESSOR: A processor within the power subsystem responsible for its operation
			X'0210'	BATTERY
			X'0211'	MOSS BATTERY
			X'0212'	UNINTERRUPTIBLE POWER SUPPLY
			X'0220'	MAIN AC POWER SUPPLY
			X'0240'	INTERNAL CLOCK: A mechanism which keeps time.

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
		X'0300' COOLING OR HEATING SUBSYSTEM: The subsystems within a hardware product responsible for maintaining a temperature at which the product can operate
		X'0301' COOLING FAN
		X'0302' AIR FILTER
		X'0310' AIR FLOW DETECTOR
		X'0311' THERMAL DETECTOR
		X'0400' SUBSYSTEM CONTROLLER: A unit within a subsystem that interfaces between a processor and the devices in the subsystem *Note:* See Alert Description X'1600', SUBSYSTEM FAILURE, for descriptions of the particular subsystems mentioned here
		X'0401' STORAGE CONTROLLER *Note:* This code point is contrasted with X'3131', DASD CONTROL UNIT and X'3132', TAPE CONTROL UNIT. A storage controller is typically a component within a larger node that provides for the node's communication with a variety of storage devices; a DASD or tape control unit is typically a separate device providing communication with storage devices.
		X'0402' WORKSTATION CONTROLLER
		X'0403' COMMUNICATIONS SUBSYSTEM CONTROLLER *Note:* This code point should be contrasted with X'3111', COMMUNICATION CONTROLLER. A communication controller is typically a stand-alone node within a network, for example, a 3725; a communications subsystem controller is typically a component within a larger node that provides for the node's communication with nodes remote from it.
		X'0421' STORAGE CONTROLLER MICROCODE
		X'0422' WORKSTATION CONTROLLER MICROCODE
		X'0423' COMM SUBSYSTEM CONTROLLER MICROCODE
		X'0441' STORAGE CONTROLLER INTERFACE: The interface between a storage controller and the main processor in its node
		X'0442' WORKSTATION CONTROLLER INTERFACE: The interface between a workstation controller and the main processor in its node
		X'0443' COMM SUBSYSTEM CONTROLLER INTERFACE: The interface between a communications subsystem controller and the main processor in its node

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
		X'0500' SUBSYSTEM: A set of components that jointly provide a specified function; typically a subsystem includes a controller, one or more interface adapters, physical connection media, and attached devices *Note:* See Alert Description X'1600', SUBSYSTEM FAILURE, for descriptions of the particular subsystems mentioned here
		X'0501' STORAGE SUBSYSTEM
		X'0502' WORKSTATION SUBSYSTEM
		X'0503' COMMUNICATIONS SUBSYSTEM
		X'0504' LOGICAL X.25 DCE: A communications subsystem which is configured as a logical DCE, as opposed to a network DCE.
		X'0505' X.25 DTE: A communications subsystem which is configured as a DTE.
		X'0506' CHANNEL SUBSYSTEM: A subsystem that processes channel operations, routes I/O interruptions and moves data between main storage and an I/O interface
		X'0507' CALLER PROCESSING SUBSYSTEM: The telephone interface subsystem which processes touch-tone input and voice input. It resides between the telephone system and an application processor.
		X'0509' COMMUNICATIONS SYSTEM SERVICES
		X'1000' SOFTWARE PROGRAM: A program implemented in software, as distinguished from one implemented in microcode *Note:* For this code point, and for the replacement code points under it, an Alert receiver has two options: It may display the English text (or its national language equivalent) documented with the code points, or it may display the software product common name from the first software Product Identifier (X'11') subvector within the indicated resource Product Set ID.
		X'1001' APPLICATION PROGRAM: A program written for or by a user that applies to the user's work. A program used to connect and communicate with devices in a network, enabling users to perform application-orientated activities
		X'1002' LOADABLE SOFTWARE MODULE: Software that may be loaded or unloaded as a containable unit.
		X'1003' SYNCHRONIZATION POINT MANAGER
		X'1004' ENCRYPTION PROGRAM
		X'1005' VOICE RESPONSE UNIT PROGRAM
		X'1006' LAN OVER WAN COMMUNICATIONS PROGRAM
		X'1010' HOST PROGRAM: A program running in a host processor that is a primary or controlling program in a system
		X'1011' PRINTER SERVER PROGRAM: A program running in a printer server that controls a printer *Note:* See also Probable Cause X'0011' (PRINTER SERVER).

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
		X'1012'	SOFTWARE DEVICE DRIVER: A program designed to control a device.
		X'1020'	CONTROL PROGRAM: A computer program designed to schedule and supervise the execution of programs in a computer system
		X'1021'	COMMUNICATION CONTROLLER CONTROL PROGRAM: A software program designed to schedule and supervise the execution of programs in a communication controller
		X'1022'	COMMUNICATIONS PROGRAM: A software program designed to provide direct assistance to a node in communicating with other nodes
		X'1023'	COMMUNICATIONS PROGRAM IN REMOTE NODE
		X'1024'	COMMUNICATION ACCESS METHOD
		X'1025'	COMMUNICATIONS PROGRAM IN LOCAL NODE
		X'1030'	LAN MANAGER PROGRAM: The software program in a LAN manager
		X'1031'	LAN MANAGEMENT SERVER: A data collection and distribution point for a single LAN segment token-ring or bus. A LAN management server forwards data received from stations on its token-ring or bus and possibly results from preliminary analysis performed by the server (on that data) to the LAN manager. LAN management servers also send data to stations on their token-rings or busses. *Note:* The LAN management servers that are currently defined are: Ring Error Monitor (REM), Configuration Report Server (CRS), Ring Parameter Server (RPS), LAN Bridge Server(LBS), and LAN Reporting Mechanism (LRM).
		X'1040'	I/O ACCESS METHOD
		X'1050'	AGENT PROGRAM: An agent has experienced a permanent error condition. This is an internal error in a software component.
		X'1051'	COMMAND NOT RECOGNIZED: An internal command encountered at a software component was not recognized. This is an internal error in a software component.
		X'1052'	CONVERSATION PROTOCOL: A conversation protocol error was encountered by a software component. This is an internal error in a software component.
		X'1053'	DATA DESCRIPTOR: The description of the data defined in a distributed unit of work was in error. This is an internal error in a software component.
		X'1054'	INVALID DATA STRUCTURE: The data structure encountered by a software component was in error. This is an internal error in a software component.

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'1055'	INVALID CURSOR STATE: The cursor state during a query operation against a relational data base was invalid. This is an internal error in a software component.
			X'1056'	RELATIONAL DATABASE ACCESS: An error was encountered while accessing a relational database. This is an internal error in a software component.
			X'1057'	RESOURCE LIMIT REACHED
			X'1058'	OBJECT ACCESS AUTHORIZATION
			X'1059'	OBJECT DEFINITION DROPPED: An example of the use of this code point is that the definition of a database object was expected to exist. It did not, so a process operating on the object failed.
			X'105A'	SYNCHRONIZATION PROCESSING
			X'105B'	COMMAND NOT AUTHORIZED
			X'105C'	PARTNER COLD START
			X'105D'	WARM START RECONNECTION FAILURE
			X'1060'	QUEUE ALLOCATION FAILURE
			X'1061'	QUEUE OPERATION FAILURE
			X'1062'	FILE CREATE FAILURE
			X'1063'	FILE ACCESS FAILURE
			X'1064'	FILE PROCESSING FAILURE
		X'1100'	OPERATING SYSTEM: Software that controls the execution of programs. An Operating system may provide services such as resource allocation, scheduling, I/O control, and data management.	
		X'2000'	COMMUNICATIONS: The facility used to permit data flow from one location to another *Note:* This code point, and the replacement code points under it, is used only when a more appropriate probable cause cannot be determined.	
			X'2001'	START-STOP COMMUNICATIONS: Asynchronous transmission in which a group of signals representing a character is preceded by a start element and is followed by a stop element; for example, ASCII
			X'2002'	BINARY SYNCHRONOUS COMMUNICATIONS: Synchronous transmission of binary-coded data between stations, using a standard set of control characters and control character sequences
			X'2003'	SNA COMMUNICATIONS: Communication according to the Systems Network Architecture formats, protocols, and operational sequences
			X'2004'	SDLC COMMUNICATIONS: (Synchronous Data Link Control)—synchronous, code-transparent, serial-by-bit information transfer over a link connection

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
		X'2005'	X.21 NETWORK: A network implementing the X.21 protocols. These protocols define an interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for synchronous operation on circuit switched public data networks
		X'2006'	X.25 NETWORK: A packet switching data network implemented according to the recommendation developed by the CCITT which provides a standard interface for the connection of processing equipment
		X'2007'	LAN LLC COMMUNICATIONS: (Logical Link Control)—error-free, in sequence information transfer over a LAN
		X'2008'	X.25 COMMUNICATIONS: Communications according to CCITT recommendation X.25 for a packet switching data network. *Note:* Use code point X'2006' (X.25 NETWORK) if the problem is known to be in X.25 network.
		X'2009'	X.21 COMMUNICATIONS: Communications according to CCITT recommendation X.21 for a circuit switching data network. *Note:* Use code point X'2005' (X.21 NETWORK) if the problem is known to be in X.21 network.
		X'200A'	ISDN NETWORK: A network implementing the Integrated Services Digital Network protocols
		X'200B'	OSI COMMUNICATIONS: Communications according to OSI and CCITT standards
		X'200C'	NETBIOS COMMUNICATIONS: A protocol error occurred in the NETBIOS interface
		X'200D'	FRAME RELAY COMMUNICATIONS: Communications according to the Frame Relay protocols for a packet switching data network.
		X'2010'	DDS NETWORK: A network implementing the Digital Data Service, e.g., DATAPHONE[1] Digital Service (DDS). [1] DATAPHONE is the Registered Service Mark of AT&T Company.
		X'2011'	SWITCHED NETWORK: This could be (but is not limited to) a public switched telephone network
		X'2012'	SERIAL NETWORK
		X'2013'	DS1 NETWORK
		X'2021'	BANKING LOOP: A network configuration, specifically designed for the finance industry, in which there is a single path between all devices and the path is a closed circuit terminating in a controller
		X'2022'	STORE LOOP: A network configuration, specifically designed for the retail industry, in which there is a single path between all devices and the path is a closed circuit terminating in a controller

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'2031'	LINE: The telephone line or transmission link connecting two or more components in the network *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.
			X'2033'	LINE/REMOTE MODEM: A line or the modem on it remote from the Alert sender
			X'2034'	LINE/REMOTE LDM: A line or the limited distance modem on it remote from the Alert sender
			X'2035'	LINE/REMOTE DIGITAL DATA DEVICE: A line or the digital data device (DDD) on it remote from the Alert sender
			X'2036'	LINE/REMOTE DCE A line or the Data Circuit-Terminating Equipment (DCE) on it remote from the Alert sender *Note:* This code point is used only if the Alert sender is unable to determine whether the DCE is a modem or a DDD; see code points X'2033' and X'2035'.
			X'2037'	DCE-DSE CONNECTION: The telephone line connecting the calling DCE to its local DSE
			X'2040'	INTER-EXCHANGE NETWORK: A network providing services between two local exchange areas
			X'2041'	PRIVATE NETWORK REACHED: The private network containing the called DTE
			X'2050'	PACKET LAYER CONTROL
			X'2051'	LINK ACCESS PROTOCOL BALANCED
			X'2052'	LOGICAL LINK CONTROL
			X'2058'	SERIAL LINK
			X'2080'	HOST COMMUNICATIONS *Note:* If the Alert sender is aware of the protocol being used for communication with the host, it uses a code point identifying that protocol.
			X'20A7'	OUTBOUND LINE: The equipment that connects the transmit circuits of the local DCE (i.e., the DCE local to the node sending error notification) to the receive circuits of the remote DCE.
			X'20A8'	INBOUND LINE: The equipment that connects the receive circuits of the local DCE (i.e., the DCE local to the node sending the error notification) to the transmit circuits of the remote DCE.
		X'2100'		COMMUNICATIONS/REMOTE NODE: Either a communications facility denoted by a X'20xx' code point or a remote node denoted by a X'22xx' code point *Note:* This code point is used only when a more specific probable cause cannot be determined.
			X'2101'	START-STOP COMMUNICATIONS/REMOTE NODE

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
			X'2102' BSC COMMUNICATIONS/REMOTE NODE
			X'2104' SDLC COMMUNICATIONS/REMOTE NODE
			X'2105' X.21 COMMUNICATIONS/CALLED DTE
			X'2106' X.25 COMMUNICATIONS/REMOTE NODE
			X'2107' LAN LLC COMMUNICATIONS/REMOTE NODE
			X'210A' ISDN COMMUNICATIONS/REMOTE NODE
			X'2130' LINE/REMOTE NODE
			X'2131' COMMUNICATIONS PROGRAM IN ADJACENT NODE
		X'2200'	REMOTE NODE: The node at the remote end of a link connection *Note:* "Remote"file defined from the point of view of the node detecting the Alert condition.
			X'2201' CALLED DTE: On a switched telephone connection, the data terminal equipment (DTE) to which the telephone call to establish the connection was placed
			X'2204' OTHER REMOTE NODE: On a multipoint link, the remote node interfering with the link activity but not part of the logical connection for which the error was detected
		X'2300'	CONNECTION NOT ESTABLISHED: A telephone connection required for the requested operation has not been established
			X'2301' CALLED NUMBER BUSY: The telephone number dialed for a teleprocessing connection was busy
			X'2302' CALLED NUMBER DID NOT ANSWER: The telephone number dialed for a teleprocessing connection did not answer
			X'2303' CALLED NUMBER OUT OF ORDER: The telephone number dialed for a teleprocessing connection is inoperative
			X'2304' INCORRECT NUMBER CALLED: The telephone number dialed for a teleprocessing connection was incorrect
			X'2305' MANUAL DIAL REQUIRED: The operator must establish a manual dial connection to a remote device before normal operation can continue
			X'2306' CHANGED NUMBER: The called DTE has recently been assigned a new number (unique X.21 status provides this information)
			X'2307' INVALID REQUEST: In the course of attempting to set up a telephone connection, the caller has made an invalid request
			X'2308' ACCESS BARRED: The calling DTE is not allowed to connect to the called DTE
			X'2309' LINK AND/OR AUTO-CALL UNIT IN USE: An auto-call attempt failed because either the link or the attached auto-call unit was in use.
			X'230A' CALL COLLISION: An outgoing call was not completed because it collided with an incoming call on the same link.
			X'230B' LINK SET UP FAILURE

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
		X'230C' SERVICE NOT AVAILABLE OR NOT SUPPORTED
		X'2600' ELECTRICAL INTERFERENCE: An electrical disturbance in a communication system that interferes with or prevents reception of a signal or of information
		X'3000' CHANNEL: The equipment that is used to direct data to and from input/output devices and locally-attached control units *Note:* This code point applies only to the channel itself. If the channel interface cable is intended, code point X'3411' (CHANNEL INTERFACE CABLE) is used instead.
		X'3100' CONTROLLER: A communication device that controls other devices and the flow of information to and from them *Note:* For this code point, and for the replacement code points under it, an Alert receiver has two options: It may display the English text (or its national language equivalent) documented with the code points; or it may display the machine type, or, if one is present, the hardware product common name, from the first hardware Product Identifier (X'11') subvector within the indicated resource Product Set ID.
		X'310F' COMMUNICATION CONTROLLER RECOVERY: A process which recovers resources from a back-up processor in a communication controller. *Note:* This code point is used to notify the network operator about a maintenance procedure that was invoked locally or initiated automatically which results in the availability of additional resources.
		X'3110' COMMUNICATION CONTROLLER BACK-UP: A process which switches resources from one processor to a back-up processor in a communication controller. *Note:* This code point is used to notify the network operator about a maintenance procedure that was invoked locally or initiated automatically which results in the availability of additional resources.
		X'3111' COMMUNICATION CONTROLLER: A communication device that controls the transmission of data over links in a network *Note:* In SNA, a communication controller is a type 4 node.
		X'3112' SENDING NODE: The node detecting the error and sending the error notification for it.
		X'3113' SENDING NODE AND MODEMS CONFIGURATION
		X'3114' SENDING NODE AND DSU/CSU'S CONFIGURATION
		X'3115' SENDING NODE/TAILED-CIRCUIT CABLE: The error notification sender configuration is incorrect or the tailed-circuit attachment cable is not connected or present

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
		X'3121'	TERMINAL CONTROL UNIT: A communication device that controls the transmission of data to and from terminals *Note:* In SNA, a terminal control unit is a type 2.0 or 2.1 node.
		X'3122'	FINANCE CONTROLLER: A terminal control unit specifically designed for the banking industry
		X'3123'	STORE CONTROLLER: A terminal control unit specifically designed for the retail industry
		X'3131'	DASD CONTROL UNIT: A device that controls the transfer of data to and from a direct access storage device such as disk or drum
		X'3132'	TAPE CONTROL UNIT: A device that controls the transfer of data to and from tape drives
		X'3200'	COMMUNICATIONS INTERFACE: The equipment connecting a node to the component in a link connection with which it exchanges physical control signals *Note:* This code point covers (1) the receivers and drivers in the node, (2) the cable, and (3) the component in the link connection that responds to the physical control signals from the node (e.g., a modem). This code point is used only when a more specific probable cause cannot be determined.
		X'3220'	LOCAL TOKEN-RING ADAPTER INTERFACE: The programming interface for the local token-ring adapter
		X'3221'	CSMA/CD ADAPTER INTERFACE: The programming interface for the local CSMA/CD adapter
		X'3222'	ISDN ADAPTER INTERFACE: The programming interface for the local ISDN adapter
		X'3223'	TOKEN-RING ADAPTER INTERFACE: The programming interface for a token-ring adapter
		X'3224'	LOCAL AUTO-CALL UNIT INTERFACE
		X'3225'	ISDN-R INTERFACE
		X'3226'	FDDI ADAPTER INTERFACE: The programming interface for a FDDI adapter
		X'32D1'	LOCAL DCE COMMUNICATIONS INTERFACE: The communications interface between the Alert sender and the local Data Circuit-Terminating Equipment (DCE)
		X'32D2'	REMOTE DCE COMMUNICATIONS INTERFACE: The communications interface between the Data Circuit-Terminating Equipment (DCE) remote from the Alert sender and the remote node
		X'32D3'	DCE EMULATION INTERFACE: The communications interface between the Alert sender and the DCE emulation cable that attaches it to a device's DCE interface cable

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
		X'3300' ADAPTER: The part of a device that interfaces between a processor in the device and one or more attached devices *Note:* The processor referred to here could be either the main processor in the node containing the adapter or a processor in, e.g., a communication subsystem controller.
		X'3301' CHANNEL ADAPTER
		X'3302' COMMUNICATIONS ADAPTER
		X'3309' LINE ADAPTER *Note:* A line adapter in a communication controller is often referred to as a scanner.
		X'330F' HPTSS ADAPTER: A high-speed processor transmission subsystem adapter in a communication controller
		X'3310' LOCAL ISDN ADAPTER: An adapter that attaches the Alert sender to an ISDN network *Note:* See also code point X'3532' LOCAL ISDN TERMINAL ADAPTER. A terminal adapter is distinguished from an ISDN adapter by the presence of a defined interface (e.g., RS-232C) between itself and the node that it serves; an ISDN adapter is typically integrated within its node.
		X'3311' REMOTE ISDN ADAPTER: An adapter that attaches to an ISDN network a node with which the Alert sender has a logical connection using the network *Note:* See also code point X'3533' REMOTE ISDN TERMINAL ADAPTER.
		X'3312' LOCAL DS1 ADAPTER
		X'3320' LOCAL TOKEN-RING ADAPTER: An adapter that attaches the Alert sender to a token-ring LAN
		X'3321' REMOTE TOKEN-RING ADAPTER: An adapter that attaches a node other than the Alert sender to a token-ring node
		X'3322' LOCAL CSMA/CD ADAPTER: An adapter that attaches the Alert sender to a CSMA/CD LAN
		X'3323' REMOTE CSMA/CD ADAPTER: An adapter that attaches a node other than the Alert sender to a CSMA/CD LAN
		X'3325' CSMA/CD ADAPTER
		X'3326' TOKEN BUS ADAPTER
		X'3328' 3270 ADAPTER
		X'3330' ADAPTER HARDWARE: The hardware comprising an adapter
		X'3331' ADAPTER MICROCODE: The microcode executing in an adapter
		X'3340' LOCAL LAN ADAPTER
		X'3350' PCNET ADAPTER
		X'3351' FDDI ADAPTER
		X'3352' T1 ADAPTER

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'3353'	T3 ADAPTER
			X'3354'	HSSI ADAPTER: A high-speed serial interface adapter
			X'3355'	E1 ADAPTER
			X'3356'	J1 ADAPTER
			X'3360'	PARALLEL CHANNEL ADAPTER
			X'3361'	SERIAL OPTICAL CHANNEL ADAPTER
			X'336F'	CONTROLLER BUS ADAPTER
			X'3370'	OPTICAL SYSTEM BUS ADAPTER
			X'3380'	ROTARY GROUP: A number of ports on a device that are all reached via the same telephone number; a rotary group is sometimes referred to as MLSA (multiple lines at same address)
			X'3381'	X.21 ROTARY GROUP
			X'33C1'	LINE ADAPTER HARDWARE
			X'33C2'	LINE ADAPTER MICROCODE
			X'33C3'	LINE INTERFACE COUPLER (LIC)
		X'3400'	CABLE: A cable or its connectors used to electrically connect devices together	
			X'3401'	LOCAL DCE INTERFACE CABLE: The cable, or its connectors, between the Alert sender and the local Data Circuit-Terminating Equipment (DCE)
			X'3403'	REMOTE DCE INTERFACE CABLE: The cable, or its connectors, between the Alert sender's remote DCE and the device attached to it. (The device could be another DCE, e.g., the local DCE on a second link segment.) *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.
			X'3404'	DCE EMULATION CABLE: The cable, or its connectors, between the Alert sender and a DCE interface cable attached to a device *Note:* The end of the DCE emulation cable remote from the Alert sender plugs directly into the DCE interface cable attached to the device.
			X'3411'	CHANNEL INTERFACE CABLE: The cable or cables, or their connectors, between a channel and a locally attached device
			X'3426'	CSMA/CD LAN CABLES: The cables of a CSMA/CD LAN. These include the cable attaching the Alert sender to the CSMA/CD bus and the bus itself.
			X'3436'	LOCAL CSMA/CD ADAPTER CABLE: The cable attaching the Alert sender to the CSMA/CD bus
			X'3441'	LOOP CABLE: A cable connecting the nodes attached to a communication loop

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'3451'	DEVICE CABLE: A cable connecting a device directly to a communication controller or a control unit *Note:* This code point also covers any passive distribution assembly that, externally, is indistinguishable from the cable itself.
			X'3452'	STORAGE DEVICE CABLE: A cable directly connecting a local storage device to its adapter/controller
			X'3460'	INTERNAL CABLE
			X'3461'	CABLE TERMINATOR
			X'3462'	LOCAL DCE LOOP: the DCE loop local to the error notification sender. *Note:* A DCE loop is the equipment comprised of cables, converters, etc., that connect the DCE with the nearest central office exchange; this equipment does not include the customer premises wiring.
			X'3463'	REMOTE DCE LOOP: The DCE loop remote from the error notification sender.
			X'3464'	TELECOMMUNICATION CABLE CONNECTION: The connection of the telecommunication cable with the local DCE or with the telephone connecting block provided by the telecommunications facility.
			X'3465'	PREMISES WIRING
			X'3470'	OPTICAL SYSTEM BUS CABLE
		X'3500'	COMMUNICATION EQUIPMENT: External equipment used to connect devices or other system components *Note:* If the attaching equipment is known to be a modem, then a modem code point (X'36xx') is sent instead of this code point. *Note:* LAN components are not reported with X'35xx' code points; see the X'3700' code point for a discussion of how they are reported.	
			X'3501'	PROTOCOL CONVERTER: A device that converts one protocol data stream to another
			X'3502'	TERMINAL MULTIPLEXER: The equipment used to connect multiple devices to a single cable
			X'3503'	LINE SWITCH: A device that on demand allows Data Circuit-terminating Equipment (DCE) to be attached to different Data Terminal Equipment (DTE) ports. The device supports both digital switching for the DCE-DTE interface and also the switching of the analog interface between the DCE and the communication facility (transmission medium).

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
		X'3504'	TIME DIVISION MULTIPLEXER: A device that combines digital data streams from different tributary channels into one data stream on a common channel; a separate periodic time interval is allocated to each tributary channel in the common channel. It also performs the reverse process of demultiplexing the composite data stream from the common channel into its constituent component data streams for the tributary channels
		X'3505'	STATISTICAL MULTIPLEXER: A device that combines digital data streams from different tributary channels into one data stream for the common channel; it takes advantage of the bursty nature of information on the tributary channels to interleave information from these channels onto the common channel. It also performs the reverse process of demultiplexing the composite data stream into its constituent component data streams
		X'3506'	LOCAL DIGITAL DATA DEVICE: The digital data device (DDD) connected to the Alert sender
		X'3507'	REMOTE DIGITAL DATA DEVICE: The digital data device (DDD) remote from the Alert sender
		X'3508'	LOCAL AUTO-CALL UNIT
		X'3509'	SERIAL NODE
		X'350A'	NETWORK COMPONENT
		X'3510'	CALLED DCE *Note:* See also code point X'3542' REMOTE DCE. X'3510' is used when reporting a problem encountered during an attempt to establish a switched connection. X'3542' is used when the problem is not related to the establishment of a switched connection.
		X'351F'	DS1 NETWORK COMPONENT
		X'352F'	ISDN TE-NT CONNECTION
		X'3530'	ISDN NETWORK COMPONENT
		X'3531'	ISDN NETWORK TERMINATION EQUIPMENT (NT1): A device, normally residing on the user's premises, that provides conversion, for basic-rate ISDN service, between the 4-wire interface seen by the user and the 2-wire interface seen by the ISDN service provider
		X'3532'	LOCAL ISDN TERMINAL ADAPTER: The terminal adapter local to the Alert sender *Note:* See also code point X'3310' LOCAL ISDN ADAPTER. A terminal adapter is distinguished from an ISDN adapter by the presence of a defined interface (e.g., RS-232C) between itself and the node that it serves; an ISDN adapter is typically integrated within its node.

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
		X'3533'	REMOTE ISDN TERMINAL ADAPTER: The terminal adapter that attaches to an ISDN network a node with which the Alert sender has a logical connection utilizing the network *Note:* See also code point X'3311' REMOTE ISDN ADAPTER.
		X'3534'	LOCAL DSU/CSU: The DSU/CSU local to the error notification sender *Note:* DSU/CSU is a signal converter which implements the function of a Data Service Unit (DSU) and Channel Service Unit (CSU) to provide the DTE interface and the line interface, respectively, with a Digital Data Service (DDS). *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.
		X'3535'	REMOTE DSU/CSU: The DSU/CSU remote from the error notification sender. *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.
		X'3536'	LOCAL AND REMOTE DSU/CSU'S *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.
		X'3541'	LOCAL DCE: The Data Circuit-Terminating Equipment (DCE) connected to the Alert sender *Note:* This code point is used only if the Alert sender is unable to determine whether the DCE is a modem or a DDD; see code points X'3506' and X'3601'.
		X'3542'	REMOTE DCE: The Data Circuit-Terminating Equipment (DCE) remote from the Alert sender *Note:* This code point is used only if the Alert sender is unable to determine whether the DCE is a modem or a DDD; see code points X'3507' and X'3603'. See also code point X'3510' CALLED DCE.
	X'3600'	MODEM: A device or functional unit that modulates and demodulates signals transmitted over data communication facilities	
		X'3601'	LOCAL MODEM: On a particular link segment, the modem nearer to the Alert sender *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
		X'3602'	LOCAL LINK DIAGNOSTIC UNIT: A device that connects to both sides of a local modem and provides Link Problem Determination Aid (LPDA) data for digital and analog links with non-intelligent IBM or non-IBM modems
		X'3603'	REMOTE MODEM: On a particular link segment, the modem farther from the Alert sender *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.
		X'3604'	REMOTE LINK DIAGNOSTIC UNIT: A device that connects to both sides of a remote modem and provides Link Problem Determination Aid (LPDA) data for digital and analog links with non-intelligent IBM or non-IBM modems
		X'3605'	LOCAL AND REMOTE MODEMS *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with a Failure Cause.
		X'3611'	LOCAL LDM: The limited distance modem nearer to the Alert sender
		X'3613'	REMOTE LDM: The limited distance modem farther from the Alert sender
		X'3621'	LOCAL ENHANCED MODEM: The enhanced modem connected to the Alert sender *Note:* An enhanced modem is a modem that can provide functions other than modulation/demodulation, such as establishing switched connections and storing dial digits.
		X'3700'	LAN COMPONENT: A component of a local area network. On a token-ring LAN, the LAN components include the adapters, bridges, access units, repeaters, repeater/amplifiers, and the LAN cable. On a CSMA/CD LAN, the LAN components include the adapters, bridges, cables, taps, splitters, amplifiers, and translator units. On a FDDI LAN, the LAN components include the adapters, bridges, concentrators, and the LAN (FDDI) cable. *Note:* This default code point is used to indicate that some unspecified LAN component is a probable cause. Individual LAN components are denoted by replacement code points under X'3700', with the exception of the LAN adapters, which fall under ADAPTER (X'3300'), and the CSMA/CD LAN cables, which fall under CABLE (X'3400').
		X'3701'	TOKEN-RING LAN COMPONENT
		X'3702'	TOKEN-RING LOBE: An adapter, the lobe cables connecting it to its access unit, and a portion of the access unit

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'3703'	TOKEN-RING FAULT DOMAIN: An adapter, its nearest active upstream neighbor, and the token-ring media between them; the token-ring media consists of the lobe cables, portions of one or more access units, and possibly a portion of the LAN cable
			X'3704'	TOKEN-RING DUPLICATE STATION ADDRESS
			X'3705'	TOKEN-RING REMOVE COMMAND RECEIVED: An adapter received a Remove Ring Station MAC frame
			X'3706'	OPTICAL FIBER CONVERTER: A device which converts electrical signals into optical signals and vice-versa
			X'3707'	TOKEN-RING LAN CABLES
			X'3708'	DOMAIN CONTROLLER
			X'3709'	LAN REMOVE COMMAND RECEIVED
			X'370A'	TOKEN-BUS LAN
			X'370B'	TOKEN-BUS DUPLICATE MAC ADDRESS: MAC sublayer fault indicated when the MAC sublayer has detected that there is another MAC entity on the network which has the same MAC address as the current value of the variable in this station.
			X'3714'	REMOTE TOKEN-RING LOBE: A lobe attaching a node other than the Alert sender to the token-ring
			X'3721'	CSMA/CD LAN COMPONENT
			X'3724'	CSMA/CD DUPLICATE STATION ADDRESS
			X'3725'	CSMA/CD REMOVE COMMAND RECEIVED
			X'3730'	FDDI LAN COMPONENT
			X'3731'	REMOTE FDDI LAN COMPONENT: A FDDI LAN component in a node other than the Alert sender
			X'3732'	DUPLICATE FDDI ADDRESS
			X'3733'	FDDI FAULT DOMAIN: An adapter, its nearest active upstream neighbor, and the FDDI media between them.
			X'3734'	FDDI LAN PORT
			X'3740'	LAN BRIDGE: A network component that interconnects, at the medium access sublayer (of the DLC layer, two token-rings, two busses, or a token-ring and a bus *Note:* The busses involved may use either the CSMA/CD protocol or the token bus protocol
			X'3750'	TOKEN-RING CAU ATTACHMENT MODULE: An interface to the wire lobes connecting workstations to a token-ring network.
			X'3751'	TOKEN-RING ACCESS UNIT
		X'4000'	PERFORMANCE DEGRADED	
			X'4001'	STORAGE SUBSYSTEM OVERLOADED
			X'4002'	WORK STATION SUBSYSTEM OVERLOADED
			X'4003'	COMMUNICATIONS SUBSYSTEM OVERLOADED

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
		X'5000' MEDIA: A tape, disk, diskette, or paper (or other data medium) that is required to read data from or write data on
		X'5001' DASD MEDIA: The media used in a direct access storage device; it may be either removable or non-removable
		X'5002' DISKETTE: A thin, flexible magnetic disk in a semi-rigid protective jacket, in which the disk is permanently enclosed; also termed a floppy diskette
		X'5003' TAPE: A recording medium in the form of a ribbon that has one or more tracks along its length; magnetic recordings can be made on either one or both sides
		X'5004' OPTICAL DISK: A DASD medium on which data is encoded optically
		X'5005' ID RECORDING SURFACE: The recording media on an Identification Card Reader (ICR) card is defective, missing or the reading device has failed.
		X'6000' DEVICE: An input, output, or input/output device (e.g., a terminal or disk drive) *Note:* An Alert sender may be unable to distinguish a directly-attached device from an attached protocol converter or media conversion unit by which devices are attached to it. Thus this code point covers these additional components as well.
		X'6100' INPUT DEVICE: A device that is used to enter data into a system
		X'6110' KEYBOARD: An arrangement of alphanumeric, special character and function keys laid out in a specified manner and used to enter information into a terminal, and thereby into a system
		X'6111' KEYPAD: A specialized keyboard with an arrangement of a limited number of alphanumeric, special character and/or function keys
		X'6112' SELECTOR PEN: A light sensitive pen used in display operations
		X'6113' MICR READER/SORTER: A magnetic ink character recognition reader/sorter
		X'6114' MAGNETIC STRIPE READER: A device that reads data from, and in some cases writes data to, a card containing a magnetic stripe
		X'6115' ID CARD READER: An Identification Card Reader (ICR) is a device which can read data from or write data to a magnetic stripe or an electronic chip on a consumer's identification card.
		X'6200' OUTPUT DEVICE: A device that receives data from a system

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'6210'	PRINTER: An output device that produces durable and optically viewable output in the form of characters (and optionally graphics) by a means other than by drawing with one or more pens *Note:* Contrast with code point X'6213' PLOTTER.
			X'6211'	COPIER: An output device that produces one or more copies of an original without affecting the original
			X'6212'	CAMERA: An output device that combines electronic data with a visual image on a single visual medium
			X'6213'	PLOTTER: An output device that produces graphic and/or character output by means of one or more pens that draw on the surface of the output medium *Note:* Contrast with code point X'6210' PRINTER.
			X'6220'	PRINTER CASSETTE: A removable container for inputting paper to a printer
		X'6300'	INPUT/OUTPUT DEVICE: A device whose parts can be performing an input process and output process at the same time, such as a card reader/punch	
			X'6301'	DISPLAY/PRINTER: A device that has either of the characteristics of a display or printer or both. This code point is used only when the Alert sender cannot determine whether the attached device is a display or printer
			X'6302'	DISPLAY: A workstation that requires a host connection in order to function; typically a display includes both a monitor and a keyboard
			X'6310'	DASD DEVICE: A device in which the access time is effectively independent of the location of the data. The device may use either removable or non-removable media
			X'6311'	DISKETTE DEVICE: A direct access storage device that uses a diskette as the storage medium
			X'6312'	OPTICAL DISK DEVICE: A direct access storage device that uses an optical disk as the storage medium. The disk may be either removable or non-removable
			X'6313'	TERMINAL: A device in a system or network at which data can either enter or leave. A terminal is usually equipped with a keyboard and a display device, and is capable of sending and receiving information
			X'6314'	TAPE DRIVE: An input/output device used for moving magnetic tape and controlling its movement
			X'6315'	CONSOLE: A terminal used for communication between an operator and a processor
			X'6316'	TOUCH-SENSITIVE SCREEN
			X'6318'	ENCRYPTION DEVICE
			X'6330'	DISK DRIVE ELECTRONICS: The electronic components of a DASD device
		X'6400'	DEPOSITORY: A device that receives items into a system	

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
			X'6401' ENVELOPE DEPOSITORY: A device that receives into a system items sealed in an envelope. The envelope is not opened, nor are its contents examined by the system; the envelope is stored for human action
			X'6402' CHECK DEPOSITORY: A device that receives checks into a system, then reads and retains them. It may also transfer information to a check and return the check to a user
			X'6403' CARD DEPOSITORY: A device that retains credit, personal banking, or other cards used to access a personal banking machine (PBM)
			X'6404' DOCUMENT DEPOSITORY
		X'6500'	DISPENSER: A device that dispenses items to a user of a system
			X'6501' DOCUMENT DISPENSER: A device that dispenses documents, primarily bills
			X'6502' TICKET DISPENSER
			X'6503' KEY DISPENSER
			X'6504' COIN DISPENSER
			X'6505' ENVELOPE DISPENSER
		X'6600'	SELF-SERVICE TERMINAL: A device that allows a customer of a business to perform a transaction that would otherwise require assistance by personnel of the business
			X'6601' PERSONAL BANKING MACHINE: A self-service terminal for financial transactions
			X'6630' TELLER ASSIST UNIT: A terminal that assists a financial teller in transactions *Note:* This device does not fit the strict definition of a self-service terminal, since it is used by personnel of a financial institution; it is included in this range because it is very close in function to other self-service terminals.
		X'6700'	SECURITY PROBLEM
			X'6701' AUDIBLE ALARM: A device which emits an audible sound.
			X'6702' PROTECTIVE DOOR: An electronically or mechanically operated covering for access to a device.
		X'7000'	PERSONNEL: Action on the part of customer, service, or other personnel
			X'7001' LOCAL SYSTEM OPERATOR: A person (or program) co-located with a system and responsible for the operation of all or part of it, or responsible for performing system orientated procedures

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'7002'	REMOTE SYSTEM OPERATOR: A person (or program) not co-located with a system and responsible for the operation of all or part of it, or responsible for performing system orientated procedures
			X'7003'	NETWORK OPERATOR: A person (or program) responsible for the operation of all or part of the network, or responsible for performing network orientated procedures
			X'7004'	USER: Anyone who requires the services of a computer system, such as an "end user"
			X'7005'	SYSTEM PROGRAMMER
			X'7006'	CUSTOMER PERSONNEL
			X'7007'	SERVICE REPRESENTATIVE
			X'7008'	RESOURCE ADMINISTRATOR
			X'7010'	PRINTER OPERATOR
			X'7011'	TERMINAL CONTROL UNIT OPERATOR
			X'7012'	LAN BRIDGE OPERATOR
			X'7013'	LAN MANAGER OPERATOR
			X'7014'	LAN TRACE TOOL OPERATOR: A person (or program) responsible for the operation of a tool that allows a LAN user to monitor the traffic on the LAN.
			X'7030'	FILE SERVER USER
			X'7031'	LOGGED IN USERS
		X'8000'	CONFIGURATION	
			X'8001'	STORAGE CONFIGURATION
			X'8002'	WORK STATION CONFIGURATION
			X'8003'	COMMUNICATION CONFIGURATION
			X'8004'	CUSTOMIZATION IMAGE: The set of rules which helps direct the operation of a device is suspected of causing the Alert condition.
			X'8005'	SERVER CONFIGURATION
			X'8006'	LAN CONFIGURATION TABLE
			X'8007'	LAN BRIDGE TABLE
			X'8008'	CONFIGURATION OBJECT NOT IN VALID STATE
			X'8010'	FILE DIRECTORY STRUCTURE
			X'8020'	FILE SERVER ACCOUNT SYSTEM FILE
			X'8021'	DOMAIN CONTROLLER ACCOUNT SYSTEM FILE
			X'8050'	INCONSISTENT BRIDGE CONFIGURATION DATA: User-entered data does not match the configuration data held by the LAN Manager.
		X'E000'-X'EFFF'	Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.	
		X'FE00'	UNDETERMINED: No probable cause can be determined for this Alert condition	

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content
		X'FFFF' Reserved

User Causes (X'94') Alert MS Subvector

This subvector transports code points for stored text detailing the probable user causes for the Alert condition and the recommended actions to be taken in connection with these causes. It may also transport additional detailed data, to be inserted into the text indexed by the user cause and/or recommended action code points.

User Causes (X'94') Alert MS Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the User Causes subvector
1		Key: X'94'
2−p		Two or more subfields containing user cause data, as described below for keys X'00'−X'7F' and in "Network Alert (X'0000') Common Subfields" on page 9-131 for keys X'80'−X'FE'. X'01' User Causes X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended *Note:* Subfields X'01' and X'81' are always present and the X'01' always precedes the X'81'. Depending on the code points present in the X'01' and X'81' subfields: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

User Causes (X'01') User Causes Subfield

This subfield contains one or more code points denoting probable user causes of the Alert condition, listed in order of decreasing probability. A user cause is defined to be a condition that an operator can resolve without contacting any service organization.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the User Causes subfield
1		Key: X'01'
2−q		Two-byte user cause code points. Each code point provides an index to predefined text, describing the user cause, that is displayed at the focal point. An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are listed below. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules). The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the X'94' subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. **Note:** The X'82' and X'85' subfields *cannot* both be used in the same subvector. The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content

The third digit of each user cause code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows:

X'xx0x' – X'xx9x': No detailed data qualifiers.

X'xxAx' – X'xxBx': One detailed data qualifier.

X'xxCx': Two detailed data qualifiers.

X'xxDx': Three detailed data qualifiers.

X'xxEx': One X'83' subfield.

X'xxFx': One X'84' subfield.

Defined user cause codes are:

X'0100' STORAGE CAPACITY EXCEEDED: A request has been received requiring more storage than is currently available

- X'0102' INSUFFICIENT STORAGE MEDIA SPACE
- X'0106' INSUFFICIENT MEMORY
- X'0111' THE PORTION OF MAIN STORAGE MADE AVAILABLE BY THE USER FOR A PARTICULAR OPERATION HAS BEEN EXHAUSTED
- X'0112' A USER-SPECIFIED THRESHOLD, INDICATING THAT AVAILABLE AUXILIARY STORAGE IS NEARLY FULL, HAS BEEN REACHED

X'0200' POWER OFF: The equipment is powered off and will require operator action to power on and prepare equipment for use

- X'0201' LOCAL DCE POWER OFF
- X'0202' REMOTE DCE POWER OFF
- X'0203' LOCAL DIGITAL DATA DEVICE POWER OFF
- X'0204' REMOTE DIGITAL DATA DEVICE POWER OFF
- X'0205' LOCAL MODEM POWER OFF
- X'0206' REMOTE MODEM POWER OFF
- X'0207' LOCAL LINK DIAGNOSTIC UNIT POWER OFF
- X'0208' REMOTE LINK DIAGNOSTIC UNIT POWER OFF
- X'0209' REMOTE DEVICE POWER OFF
- X'020A' LOCAL TERMINAL ADAPTER (TA) POWER OFF
- X'020B' REMOTE TERMINAL ADAPTER (TA) POWER OFF
- X'020C' LOCAL CONTROLLER POWER OFF
- X'020D' REMOTE CONTROLLER POWER OFF
- X'020E' PRINTER POWER OFF
- X'020F' COMMUNICATION EQUIPMENT POWER OFF
- X'0210' CALLING DCE POWER OFF
- X'0211' CALLED DCE POWER OFF
- X'0212' CALLED DTE POWER OFF
- X'0213' MODEM POWER OFF

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
		X'0214'	TERMINAL MULTIPLEXER POWER OFF
		X'0215'	LOCAL NT1 POWER OFF
		X'0216'	SERVICE PROCESSOR POWER OFF
		X'0220'	DEVICE POWER OFF
		X'0221'	CONSOLE POWER OFF
		X'0222'	LAN MANAGER POWER OFF
		X'0223'	REMOTE NODE POWER OFF
		X'0224'	LOCAL DSU/CSU POWER OFF
		X'0225'	REMOTE DSU/CSU POWER OFF
		X'0226'	OPTICAL FIBER CONVERTER POWER OFF: A device which converts electrical signals into optical signals and vice-versa.
		X'02A1'	(detailed data qualifier) LOCAL MODEM POWER OFF *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'02A2'	(detailed data qualifier) REMOTE MODEM POWER OFF *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		X'02A3'	(detailed data qualifier) LOCAL DSU/CSU POWER OFF *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'02A4'	(detailed data qualifier) REMOTE DSU/CSU POWER OFF *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
	X'2200'	REMOTE NODE: The node at the remote end of a link connection *Note:* "Remote" is defined from the point of view of the node detecting the Alert condition.	
		X'2201'	CALLED DTE TAKEN OUT OF SERVICE
		X'2210'	REMOTE NODE CONTROL PROGRAM IPL HAS OCCURRED
	X'2300'	CONNECTION NOT ESTABLISHED: A telephone connection required for the requested operation has not been established	
		X'2301'	CALLED NUMBER BUSY
		X'2302'	INCORRECT NETWORK SERVICE ACCESS POINT CALLED
		X'2303'	ACCESS TO CALLED NETWORK SERVICE ACCESS POINT NOT ACTIVATED
		X'2304'	INCORRECT TELEPHONE NUMBER SPECIFIED
		X'2305'	INCORRECT CALL USER DATA
		X'2306'	NEW TELEPHONE NUMBER ASSIGNED TO CALLED DTE
		X'2307'	INCORRECT PARAMETER SPECIFIED

User Causes (X'01') User Causes Subfield

Byte	Bit	Content		
			X'2308'	CALLING DTE DOES NOT SUBSCRIBE TO THIS FACILITY: The calling DTE has requested a service that it does not subscribe to.
			X'230A'	USER CLASSES OF SERVICE INCOMPATIBLE
			X'230D'	SERVICE OR SUBSCRIPTION PARAMETER ERROR
			X'2310'	X.21 CONNECTION INTENTIONALLY CLEARED BY TERMINAL CONTROL UNIT OPERATOR
			X'2313'	INCOMPLETE OR MALFORMED CALL SETUP MESSAGE
			X'2315'	FACILITY PARAMETER INVALID OR NOT SUPPORTED
			X'23A0'	CONNECTION NOT ESTABLISHED — (detailed data qualifier) *Note:* The qualifier indicates the telephone number for the connection that could not be established.
			X'23A1'	INCORRECT TELEPHONE NUMBER — (detailed data qualifier) *Note:* The qualifier indicates the telephone number called
			X'23A2'	INCORRECT MEMORY ADDRESS — (detailed data qualifier) *Note:* The qualifier indicates the memory address at which a telephone number is stored in a modem.
			X'23A3'	INVALID MODEM COMMAND — (detailed data qualifier) *Note:* The qualifier indicates the modem command that is invalid.
			X'23A4'	CALLED NUMBER BUSY — (detailed data qualifier) *Note:* The qualifier indicates the telephone number called.
			X'23A5'	CONNECTION NOT ESTABLISHED FOR (detailed data qualifier) *Note:* The qualifier indicates the type of call.
			X'23A6'	SERVICE CALL CANNOT COMPLETE FOR (detailed data qualifier) *Note:* The qualifier indicates the type of service call.
		X'2400'	BUSY: A requested resource was unavailable because it was in use	
			X'2401'	THE MAXIMUM NUMBER OF WORKSTATIONS SUPPORTABLE BY THE LOCAL WORKSTATION CONTROLLER HAS BEEN EXCEEDED
			X'2402'	THE MAXIMUM NUMBER OF USERS SUPPORTABLE BY THE SERVER HAS BEEN EXCEEDED
			X'2403'	MAXIMUM NUMBER OF LINES ALLOWED ALREADY IN USE
		X'2500'	LINE NOT ENABLED: A communication link has not been prepared for data transmission	
			X'2501'	PORT DEACTIVATED

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
		X'2510'	LINE NOT ENABLED AT CALLED DTE
		X'2511'	PORT DEACTIVATED AT CALLED DTE
		X'2520'	LINE NOT VARIED ON
		X'3300'	ADAPTER NOT READY: An adapter has not been made ready for use
		X'3380'	AN OPERATOR HAS DEACTIVATED ALL PORTS IN A ROTARY GROUP
		X'3381'	AN OPERATOR HAS DEACTIVATED ALL PORTS IN A ROTARY GROUP USED BY AN X.21 SHORT HOLD MODE SESSION
		X'33A1'	ADAPTER (detailed data qualifier) AWAITING DOWN-LOAD *Note:* The qualifier indicates either the adapter card number, the card name, or the card location.
		X'3400'	CABLE NOT CONNECTED: A cable is either loose or disconnected
		X'3401'	CABLING INSTALLED INCORRECTLY
		X'3402'	KEYBOARD UNPLUGGED
		X'3403'	LINE SWITCHED TO INCORRECT POSITION
		X'3451'	DEVICE CABLE NOT CONNECTED
		X'34A0'	CABLE NOT CONNECTED: (detailed data qualifier) *Note:* The qualifier specifies the disconnected cable, by, for example, specifying the number of the port to which it should be attached.
		X'34A1'	CABLE NOT INSTALLED: (detailed data qualifier) *Note:* The qualifier specifies the cable that was not installed.
		X'34A2'	CABLE UNPLUGGED: (detailed data qualifier) *Note:* The qualifier specifies the cable that is unplugged.
		X'3800'	LPDA DCE: A DCE that supports link problem determination aid functions, e.g., IBM LPDA-2 modem and IBM LPDA-2 DSU/CSU.
		X'3801'	SNBU HAS BEEN DISCONNECTED
		X'3802'	TC LEAD ACTIVE ON REMOTE NODE INTER-FACE
		X'3803'	TC LEAD ACTIVE ON OTHER REMOTE NODE INTERFACE
		X'38A1'	SPEED MISMATCH BETWEEN MODEMS ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
		X'38A2'	SPEED MISMATCH BETWEEN DSU/CSU'S ON (detailed data qualifier) *Note:* The qualifier identifies the the link segment level (LSL) on which the DSU/CSUs belong.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
		X'38A3'	(detailed data qualifier) LOCAL MODEM IN TEST MODE *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'38A4'	(detailed data qualifier) LOCAL DSU/CSU IN TEST MODE *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'38A5'	(detailed data qualifier) REMOTE MODEM IN TEST MODE *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		X'38A6'	(detailed data qualifier) REMOTE DSU/CSU IN TEST MODE *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
		X'38A7'	(detailed data qualifier) LOCAL MODEM REINITIALIZED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'38A8'	(detailed data qualifier) LOCAL DSU/CSU REINITIALIZED *Note:* The qualifier identifies the link segment level (LSL) on which the local DSI/CSU belongs.
		X'38A9'	(detailed data qualifier) LOCAL DSU/CSU DETECTED DDS LOOPBACK ACTIVE IN THE LAST 2 MINUTES *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
		X'38AA'	(detailed data qualifier) REMOTE DSU/CSU DETECTED DDS LOOPBACK ACTIVE IN THE LAST 2 MINUTES *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
		X'38AB'	(detailed data qualifier) LOCAL MODEM POWER OFF THEN ON *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'38AC'	(detailed data qualifier) REMOTE MODEM POWER OFF THEN ON *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		X'38AD'	(detailed data qualifier) LOCAL DSU/CSU POWER OFF THEN ON *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
		X'38AE'	(detailed data qualifier) REMOTE DSU/CSU POWER OFF THEN ON *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
		X'38AF'	(detailed data qualifier) REMOTE DSU/CSU IN CONFIGURATION MODE *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
		X'38B0'	(detailed data qualifier) LOCAL DSU/CSU IN CONFIGURATION MODE *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'4000'	PERFORMANCE DEGRADED
		X'4001'	STORAGE SUBSYSTEM OVERLOADED: The number of attached devices is not sufficient to handle the current work load without performance degradation.
		X'4002'	WORK STATION SUBSYSTEM OVERLOADED: The combination of work stations attached and/or the current applications are causing an excessive work load resulting in performance degradation.
		X'4003'	COMMUNICATION SUBSYSTEM OVERLOADED: The number of lines, maximum aggregate data rate, or number of attached devices is in excess of that which can be handled without performance degradation.
		X'5100'	MEDIA DEFECTIVE: The medium (tape, disk, diskette, paper, e.g.) is defective and must be replaced or corrected to continue processing
		X'5101'	IMPROPER DISKETTE INSERTED: There is a usable diskette in the diskette drive, but it is not the required one
		X'5102'	NO DISKETTE OR DEFECTIVE DISKETTE INSERTED: There is no diskette in the diskette drive, or the diskette in the drive is unusable
		X'5110'	NON-DUPLEX PAPER IN CASSETTE
		X'5111'	ID CARD RECORDING SURFACE: The recording media on an Identification Card Reader (ICR) card is defective, missing or the reading device has failed.
		X'5200'	MEDIA JAM: The medium (usually paper, forms, or cards) is jammed in the machine and operator action is required to correct the problem
		X'5201'	CARD JAM
		X'5202'	FORMS JAM
		X'5203'	PAPER JAM
		X'5204'	FILM JAM: There is a jam condition in the media for a camera device.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content
		X'5300' MEDIA SUPPLY EXHAUSTED: The medium (usually paper, forms, or cards) supply has been consumed and operator action is required to replenish the supply and continue operation
		X'5301' OUT OF CARDS
		X'5302' OUT OF FORMS
		X'5303' OUT OF PAPER
		X'5304' OUT OF FILM: The media for a camera device has been exhausted.
		X'5305' OUT OF BILLS OR DOCUMENTS: The media for a document feeding device has been exhausted.
		X'5306' OUT OF ENVELOPES
		X'5313' CASSETTE OUT OF PAPER
		X'5400' OUT OF SUPPLIES: A device is out of supplies required for it to perform its function
		X'5401' END OF RIBBON ENCOUNTERED: A printer has encountered the end of the print ribbon
		X'5402' OUT OF INK
		X'5403' OUT OF TONER
		X'5404' OUT OF FUSER OIL
		X'5405' OUT OF STAPLES
		X'5500' MEDIA SUPPLY LOW: The medium (usually paper, forms, or cards) supply is low and operator action is required to replenish the supply and continue operation
		X'5501' LOW ON CARDS
		X'5502' LOW ON FORMS
		X'5503' LOW ON PAPER
		X'5504' LOW ON FILM: The media for a camera device is nearly exhausted.
		X'5505' LOW ON BILLS OR DOCUMENTS: The media for a document feeding device is nearly exhausted.
		X'5506' LOW ON ENVELOPES
		X'5600' LOW ON SUPPLIES: A device is low on supplies required for it to perform its function
		X'5602' LOW ON INK
		X'5603' LOW ON TONER
		X'5604' LOW ON FUSER OIL
		X'5605' LOW ON STAPLES
		X'6000' DEVICE NOT READY: A device has not been made ready for operation
		X'6001' DEVICE OFFLINE: The device requested has been varied offline by the operator and must be varied online for processing to continue

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
		X'6010'	DASD DEVICE NOT READY
		X'6011'	DISKETTE NOT READY
		X'6012'	TAPE NOT READY
		X'6013'	PRINTER NOT READY
		X'6014'	BIN COVER OPEN
		X'6015'	PRINTER DOOR OPEN
		X'6016'	OUTPUT HOPPER FULL
		X'6017'	TELEPHONE SET NOT IN DATA MODE
		X'6018'	REMOTE NODE OFFLINE
		X'6019'	REMOTE NODE REINITIALIZED
		X'601A'	DOMAIN CONTROLLER STOPPED OR POWERED OFF
		X'6020'	SERVICE DOOR OPEN: The door which provides access to the interior of the machine has been opened.
		X'6021'	MODEM NOT IN DATA MODE
		X'6022'	LOCAL MODEM POWERED OFF THEN ON
		X'6400'	DEPOSITORY: A device that receives items into a system.
		X'6401'	DEPOSITORY NEARLY FULL: A cartridge or other container used to collect items such as checks, envelopes, or documents is approaching the limit of it's capacity.
		X'6402'	DEPOSITORY FULL: A cartridge or other container used to collect items such as checks, envelopes, or documents has reached it's capacity.
		X'7000'	OPERATOR: Operator action is required to return the machine to operational status
		X'7001'	NO CASSETTE IN PRINTER
		X'7002'	CARTRIDGE NOT INSTALLED CORRECTLY: A cartridge used to collect or dispense documents is not installed correctly.
		X'7003'	OUT OF FOCUS: An operator is required to make an adjustment to a camera device.
		X'7004'	USER NEEDS ASSISTANCE: Someone who uses the services of a computer system, such as an "end user," requires assistance in this usage.
		X'7005'	DEVICE IS NOT IN THE PROPER POSITION: A device is not in the correct operating position when an attempt is made to use it.
		X'7006'	NETWORK OPERATOR
		X'7010'	CALL AUTHORIZATION REQUIRED
		X'70A0'	ADDRESS COMPARE DETECTED ON (detailed data qualifier) *Note:* The qualifier identifies the resource.
		X'70A1'	ADDRESS COMPARE STOP ON (detailed data qualifier) *Note:* The qualifier identifies the resource.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
		X'70A2'	SERVICE CALL AUTHORIZATION REQUIRED FOR CALL TYPE (detailed data qualifier) *Note:* The qualifier identifies the type of service call.
		X'7100'	INCORRECT PROCEDURE: An appropriate procedure was not followed
		X'7101'	TOKEN-RING REMOVE ADAPTER COMMAND RECEIVED
		X'7102'	PAPER INSTALLED INCORRECTLY
		X'7103'	LAN MANAGER OPERATOR ENTERED INCORRECT PASSWORD
		X'7104'	UNAUTHORIZED ACCESS TO LAN MANAGEMENT SERVER ATTEMPTED
		X'7105'	UNAUTHORIZED USER ATTEMPTED INSERTION INTO LAN
		X'7106'	ADAPTER ADDRESS NOT ENTERED IN AUTHORIZATION LIST
		X'7107'	CSMA/CD REMOVE ADAPTER COMMAND RECEIVED
		X'7108'	OPERATOR ENTERED INCORRECT PASSWORD
		X'7109'	LAN BRIDGE OPERATOR TOOK BRIDGE OFFLINE *Note:* When this condition occurs, the bridge can no longer forward frames.
		X'710A'	LAN MANAGER OPERATOR TOOK BRIDGE OFFLINE *Note:* When this condition occurs, the bridge can no longer forward frames.
		X'710B'	USER INCAPACITATED LAN MANAGEMENT SERVER PROGRAM: A user has caused the LAN management server program to become inactive, but its processor is still able to process interrupts
		X'710C'	UNAUTHORIZED TRACE TOOL IN LAN: A tool that allows a LAN user to monitor the traffic on the LAN has not been authorized to be used.
		X'710D'	ADAPTER NOT AT EXPECTED ADDRESS
		X'710E'	AUTHORIZATION LIST CONTAINS INCORRECT TIME AND DAY ENTRIES
		X'710F'	INCORRECT ADDRESS ASSIGNED
		X'7110'	LOCAL X.25 PROCEDURE ERROR: An error has occurred at the side of the X.25 network nearer the Alert sender during an attempt by the Alert sender to establish an X.25 connection
		X'7111'	REMOTE X.25 PROCEDURE ERROR: An error has occurred at the side of the X.25 network remote from the Alert sender during an attempt by the Alert sender to establish an X.25 connection
		X'7112'	DESTINATION RESOURCE NOT OPEN

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
		X'7113'	INVALID CONTROL BLOCK TYPE
		X'7114'	MANUFACTURING AUTOMATION PROTOCOL NOT STARTED
		X'7115'	INVALID/INCORRECT SPECIFICATION OF ORIGINATOR RESOURCE The originator resource has been defined incorrectly or was not the expected resource.
		X'7116'	PRINTER CONTROL BLOCK STOPPED AND PURGED
		X'7117'	USER HAS ANOTHER TABLE MEMBER LOCKED
		X'7118'	MAXIMUM NUMBER OF OPERATING SYSTEM FILES ALREADY OPEN
		X'7119'	MAXIMUM NUMBER OF PRINTER CONTROL BLOCKS ALREADY STARTED
		X'711A'	MAXIMUM NUMBER OF RESOURCES ALREADY STARTED
		X'711B'	FILE SERVER PASSWORD CHANGED
		X'711C'	DIFFERENCE BETWEEN SYSTEM CLOCK TIMES UNACCEPTABLE: The difference between the system clock times is either too large or too small.
		X'711D'	PARTNER RESTARTED WITHOUT RECOVERY LOG INFORMATION
		X'711E'	LOG NAME MISMATCH
		X'711F'	INCONSISTENT HEURISTIC DECISION
		X'7120'	INCORRECT TEST TOOL USED: The test tool used for servicing the device is incorrect.
		X'7121'	USER'S LOGON TIME LIMIT EXCEEDED
		X'7130'	MULTIPLE ADAPTERS ATTACHED TO ONE LOBE
		X'7131'	TELEPHONE ANSWERED BY OTHER THAN COMMUNICATIONS EQUIPMENT
		X'7132'	PRINTER ALREADY STARTED UNDER ANOTHER PRINTER CONTROL BLOCK
		X'7133'	TASK ALREADY LOADED: An attempt was made to load a task that was already loaded. The task may be invalid or unavailable, causing an error.
		X'7140'	OPERATOR DEACTIVATED THE SESSION
		X'7141'	OPERATOR TOOK PROGRAM OFFLINE
		X'7142'	NETWORK INTERFACE NOT VARIED ON
		X'7143'	CONTROLLER DESCRIPTION OBJECT NOT VARIED ON
		X'7144'	RESOURCE NOT AVAILABLE
		X'7150'	LOCAL ISDN PROCEDURE ERROR: A procedure error occurred at the local (Alert sender) end of an attempted ISDN connection.
		X'7151'	REMOTE ISDN PROCEDURE ERROR: A procedure error occurred at the end of an attempted ISDN connection that is remote from the Alert sender.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
			X'7152' LOCAL PROCEDURE ERROR: A procedure error occurred at the local (Alert sender) end of an attempted communication connection. *Note:* Use this code point only when a more specific code point is not available (e.g. X'7110' or X'7150').
			X'7153' REMOTE PROCEDURE ERROR: A procedure error occurred at the end of an attempted communication connection that is remote from the Alert sender. *Note:* Use this code point only when a more specific code point is not available (e.g. X'7111' or X'7151').
			X'7199' UNAUTHORIZED ACCESS ATTEMPTED
			X'71A0' UNAUTHORIZED ACCESS ATTEMPTED ON (detailed data qualifier) *Note:* The qualifier identifies the account.
			X'71A1' RESOURCE (detailed data qualifier) NOT STARTED IN REMOTE NODE *Note:* The qualifier identifies the resource.
			X'71A2' (detailed data qualifier) NOT STARTED *Note:* The qualifier identifies the resource.
			X'71A3' SYSTEM RESOURCE (detailed data qualifier) LOCKED: A required resource is locked preventing continuation of a function or application. *Note:* The qualifier identifies the locked resource.
			X'71A4' (detailed data qualifier) REACHED ACCOUNT LIMITATION *Note:* The qualifier identifies the user.
			X'71A5' NETWORK CONTROL BLOCK (NCB) RESOURCE SHORTAGE ON (detailed data qualifier) *Note:* The qualifier identifies the network.
			X'71C0' (detailed data qualifier) EXCEEDED STORAGE LIMIT ON (detailed data qualifier) *Note:* The qualifiers identify the user and server respectively.
		X'7200'	DUMP REQUESTED: A machine readable copy of processor storage has been obtained at the request of an operator, user, or programmed procedure
			X'7201' MICROCODE DUMP REQUESTED
			X'7202' SOFTWARE DUMP REQUESTED
		X'7300'	FILE FULL: A requested operation cannot be performed because the file to be used for the operation does not have space available to contain the data
			X'7301' DISKETTE OR DIRECTORY FULL: There is no more diskette space or directory space on the diskette.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content	
			X'73A0' FILE FULL (detailed data qualifier): A requested operation cannot be performed because the file to be used for the operation does not have space available to contain the data *Note:* The qualifier specifies the name, or other unique identifier, of the file that is full.
			X'73A1' FILE NEEDS REORGANIZATION (detailed data qualifier): A file is approaching its capacity, and will soon be unusable unless it is reorganized *Note:* The qualifier specifies the name, or other unique identifier, of the file needing reorganization.
			X'73A2' FILE DIRECTORY TABLE FULL ON (detailed data qualifier) *Note:* The qualifier identifies the volume.
			X'73A3' FILE ALMOST FULL (detailed data qualifier) *Note:* The qualifier identifies the file name.
		X'7400'	CONTAMINATION: Dirt or some other contamination is suspected as the cause of the problem. The operator should perform routine cleaning actions required for this equipment
			X'7401' DIRTY READ/WRITE HEAD
			X'74A1' BLOCKED AIR FILTER (detailed data qualifier) *Note:* The qualifier identifies the air filter number.
		X'E000'–X'EFFF'	Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range. *Note:* The following code points specify extended messages, that provide additional information on one or more user causes that have already been specified. An Alert receiver that displays only default text provides no display for these code points.
		X'F000'	(no display): Additional message data
			X'F001' INVALID TRANSIT NETWORK ROUTING SELECTED
			X'F002' CHANNEL REQUEST INVALID
			X'F003' THERE ARE NO DELETED FILES USING STORAGE SPACE
			X'F0A0' INSUFFICIENT STORAGE MEDIA SPACE AVAILABLE FOR (detailed data qualifier) *Note:* The qualifier identifies the volume.
			X'F0A1' (detailed data qualifier) KILOBYTES OF STORAGE IS IN USE BY DELETED FILES THAT CANNOT BE PURGED YET *Note:* The qualifier identifies the amount of storage in kilobytes.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content
		X'F800' (no display): Additional message data *Note:* The X'F8xx' range is used for additional messages that are identical for User, Install, and Failure Causes. X'F8A0' PROBLEM DETECTED BY (detailed data qualifier) *Note:* The qualifier identifies the file server. X'F8C0' FAILING COMPONENT IS IDENTIFIED BY (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the failing component by means of its logical location, e.g., its port number and device address. X'F8C1' REQUEST ORIGINATED FROM (detailed data qualifier) ON (detailed data qualifier) *Note:* The qualifiers identify the node and network. X'F8D0' PROBLEM IS RELATED TO THE CONTROLLER LOCATED AT (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the controller location as follows: Q1 = RACK Q2 = UNIT (within a rack) Q3 = CARD SLOT (within a unit) X'F8E0' FAILING COMPONENT IS IDENTIFIED BY (sf83 product text) X'FFFF' Reserved

Install Causes (X'95') Alert MS Subvector

This subvector transports code points for stored text detailing the probable install causes for the Alert condition and the recommended actions to be taken in connection with these causes. It may also transport additional detailed data, to be inserted into the text indexed by the install cause and/or recommended action code points.

Install Causes (X'95') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Install Causes subvector
1		Key: X'95'

Install Causes (X'95') Alert MS Subvector

Byte	Bit	Content
2−p		Two or more subfields containing install cause data, as described below for keys X'00'−X'7F' and in "Network Alert (X'0000') Common Subfields" on page 9-131 for keys X'80'−X'FE'. X'01' Install Causes X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended *Note:* Subfields X'01' and X'81' are always present and the X'01' always precedes the X'81'. Depending on the code points present in the X'01' and X'81' subfields: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

Install Causes (X'01') Install Causes Subfield

This subfield contains one or more code points denoting probable install causes of the Alert condition, listed in order of decreasing probability. An install cause is defined to be a condition that resulted from the initial installation or set-up of some equipment.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Install Causes subfield
1		Key: X'01'
2−q		2-byte install cause code points. Each code point provides an index to predefined text, describing the install cause, that is displayed at the Alert receiver. An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content

Specific defined codes and the corresponding displayed text (shown all capitalized) are listed below. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules).

The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the X'95' subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. **Note:** The X'82' and X'85' subfields *cannot* both be used in the same subvector.

The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

The third digit of each install cause code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows:

X'xx0x' – X'xx9x':	No detailed data qualifiers.
X'xxAx' – X'xxBx':	One detailed data qualifier.
X'xxCx':	Two detailed data qualifiers.
X'xxDx':	Three detailed data qualifiers.
X'xxEx':	One X'83' subfield.
X'xxFx':	One X'84' subfield.

Defined install cause codes are:

X'1200' INCORRECT HARDWARE CONFIGURATION: The hardware has been installed incorrectly and the requested function cannot be performed

X'1201' OPTICAL FIBER CONVERTER CONFIGURATION: A device which converts electrical signals into optical signals and vice-versa, is not configured correctly.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			X'1202'	LOCAL MODEM: The modem connected to the Alert sender
			X'1203'	REMOTE MODEM: The modem remote from the Alert sender
			X'1204'	LOCAL DIGITAL DATA DEVICE: The digital data device (DDD) connected to the Alert sender
			X'1205'	REMOTE DIGITAL DATA DEVICE: The digital data device (DDD) remote from the Alert sender
			X'1206'	LOCAL DCE: The Data Circuit-Terminating Equipment (DCE) connected to the Alert sender *Note:* This code point is used only if the Alert sender is unable to determine whether the DCE is a modem or a DDD; see code points X'1202' and X'1204'.
			X'1207'	REMOTE DCE: The Data Circuit-Terminating Equipment (DCE) remote from the Alert sender *Note:* This code point is used only if the Alert sender is unable to determine whether the DCE is a modem or a DDD; see code points X'1203' and X'1205'.
			X'1208'	DEVICE NOT CALIBRATED
			X'1209'	DISK STORAGE FULL: The disk(s) in question cannot store the normally required amount of data.
		X'1300'	INCORRECT SOFTWARE GENERATION: A program has been installed incorrectly and the requested function cannot be performed	
			X'1301'	COMMUNICATIONS PROGRAM: Required communication program cannot be started.
			X'1302'	REQUIRED SOFTWARE FEATURES NOT INSTALLED
			X'13E1'	INCORRECT SOFTWARE GENERATION: (sf83 product text)
		X'1400'	MISMATCH BETWEEN HARDWARE AND SOFTWARE: A conflict exists between the hardware configuration and software	
			X'1401'	MISMATCH BETWEEN HARDWARE CONFIGURATION AND SOFTWARE GENERATION
			X'1402'	MISMATCH BETWEEN HARDWARE AND SOFTWARE CONFIGURATIONS: The hardware configuration represented in a software product does not match the actual hardware configuration
			X'1410'	MEMORY IMAGE FILE AND MEMORY OF DEVICE ARE NOT EQUAL
		X'1500'	MISMATCH BETWEEN HARDWARE AND MICROCODE: A conflict exists between the hardware configuration and microcode	
			X'1501'	INCORRECT CUSTOMIZATION PARAMETERS
			X'1502'	INCORRECT MICROCODE FIX

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'1600'	MISMATCH BETWEEN SOFTWARE AND MICROCODE: A conflict exists between a software program and a microcode program
		X'1601'	INCORRECT CUSTOMIZATION IMAGE: The software customization image is incompatible with the actual microcode configuration
		X'1602'	INCORRECT APPLICATION PROGRAM: An application software program is at the wrong level for the actual microcode configuration, or the wrong application software program is attempting to communicate with the microcode
		X'16A1'	INCORRECT SOFTWARE LEVEL (detailed data qualifier) *Note:* The qualifier specifies a generation parameter.
		X'1700'	INCORRECT VALUE SPECIFIED: An incorrect value has been specified for a system operational parameter
		X'1701'	INTERVENTION TIMER VALUE TOO SMALL
		X'1702'	RTS ACTIVATION LIMIT PARAMETER OF THE SENDING NODE IS INCORRECT
		X'1703'	REMOTE NODE TEST TIMEOUT TOO SHORT
		X'1704'	OTHER REMOTE NODE TEST TIMEOUT TOO SHORT
		X'1705'	REMOTE NODE HOLDING RTS ACTIVE
		X'1706'	OTHER REMOTE NODE HOLDING RTS ACTIVE
		X'1707'	MULTIPOINT TRIBUTARIES WITH SAME ADDRESS
		X'1708'	MISMATCH BETWEEN 2-WIRE, HALF DUPLEX COUPLER ON MODEMS AND RTS CONFIGURED FOR FULL DUPLEX BY REMOTE NODE
		X'1709'	SYSTEM FILE LOCK THRESHOLD SET TOO LOW
		X'170A'	SYSTEM RECORD LOCK THRESHOLD SET TOO LOW
		X'170B'	FILE LOCK THRESHOLD FOR WORKSTATIONS SET TOO LOW
		X'170C'	RECORD LOCK THRESHOLD FOR WORKSTATIONS SET TOO LOW
		X'170D'	CONCURRENT DIRECTORY SEARCH THRESHOLD FOR WORKSTATIONS SET TOO LOW
		X'170E'	MEMORY THRESHOLD SET TOO LOW
		X'170F'	SYSTEM TRANSMIT/RECEIVE BUFFER THRESHOLD SET TOO LOW
		X'1710'	MEMORY LOWER LIMIT SET TOO HIGH
		X'1711'	INVALID PROTOCOL PARAMETER IN COMMUNICATION CONTROL BLOCK
		X'1712'	INVALID TASK NUMBER
		X'1713'	SYSTEM TRACE FILE DEFINED WITH INVALID RECORD LENGTH

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			X'1714'	TRANSMISSION PRIORITY PARAMETER IS INCORRECT
			X'1715'	MINIMUM FREE DISK SPACE THRESHOLD SET TOO HIGH
			X'1716'	LOGON VIOLATION THRESHOLD SET TOO LOW
			X'1717'	ATTEMPTED RESOURCE ACCESS VIOLATIONS THRESHOLD SET TOO LOW
			X'1718'	ERROR LOG SIZE SET TOO LOW
			X'1719'	AUDIT LOG SIZE SET TOO LOW
			X'171A'	USER ACCOUNT LIMIT SET TOO LOW
			X'171B'	PREDEFINED RESOURCE THRESHOLD SET TOO LOW
			X'171C'	FILE SERVER ERROR THRESHOLD SET TOO LOW
			X'171D'	NETWORK I/O ERROR THRESHOLD SET TOO LOW
			X'17A0'	DIRECTORY SIZE LIMIT TOO LOW ON (detailed data qualifier) *Note:* The qualifier identifies the volume.
			X'17A1'	COMMUNICATION ACCESS CONTROL BLOCK FOR ADAPTER (detailed data qualifier) IS ALREADY STARTED *Note:* The qualifier identifies the adapter card number.
			X'17A2'	ADAPTER (detailed data qualifier) NOT RECOGNIZED BY COMMUNICATION DRIVER *Note:* The qualifier identifies the adapter card number.
			X'17A3'	NOT ENOUGH BUFFERS FOR ADAPTER (detailed data qualifier) *Note:* The qualifier identifies the adapter card number.
			X'17C0'	THRESHOLD VALUE SET TOO LOW (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the configuration object/record which contains the parameter. The second qualifier identifies the threshold parameter that is set to low.
			X'17C1'	(detailed data qualifier) IS SET AT (detailed data qualifier) *Note:* The first qualifier identifies the configuration parameter and the second qualifier specifies the threshold parameter.
		X'1800'	SERVER/CLIENT MISMATCH	
			X'1801'	CLIENT APPLICATION PROGRAM SENT AN UNENCRYPTED PASSWORD TO A SERVER CONFIGURED FOR ENCRYPTED PASSWORDS ONLY
		X'2600'	SYSTEM OR TRANSMISSION MEDIA INSTALLED NEAR ELECTRICAL INTERFERENCE	
		X'3400'	CABLE INSTALLED INCORRECTLY: A cable has been incorrectly installed	

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'3401'	LOCAL DCE INTERFACE CABLE INSTALLED INCORRECTLY
		X'3402'	LINE ADAPTER MULTIPLEXER CABLE INSTALLED INCORRECTLY
		X'3403'	REMOTE DCE INTERFACE CABLE INSTALLED INCORRECTLY
		X'3404'	DCE EMULATION CABLE INSTALLED INCORRECTLY
		X'3405'	LOCAL TELECOMMUNICATION CABLE NOT PROPERLY CONNECTED
		X'3406'	REMOTE TELECOMMUNICATION CABLE NOT PROPERLY CONNECTED
		X'3407'	PHYSICAL LINE CONNECTIONS
		X'3408'	OPTICAL FIBER CABLE INSTALLED INCORRECTLY
		X'3451'	DEVICE CABLE INSTALLED INCORRECTLY
		X'3460'	VOICE LINE CABLE INSTALLED INCORRECTLY: The cable connecting a Voice Response Unit and a PBX was improperly installed.
		X'34A0'	(detailed data qualifier) LOCAL DCE INTERFACE CABLE NOT PROPERLY CONNECTED *Note:* The qualifier identifies the link segment level (LSL) on which the local DCE belongs.
		X'34A1'	(detailed data qualifier) REMOTE DCE INTERFACE CABLE NOT PROPERLY CONNECTED *Note:* The qualifier identifies the link segment level (LSL) on which the remote DCE belongs.
		X'3500'	COMMUNICATION EQUIPMENT INSTALLED INCORRECTLY: Some communication equipment has been installed incorrectly; the Alert sender cannot determine the precise nature of this equipment
		X'3501'	MULTI-SEGMENT LINK DEFINED AND TAILED-CIRCUIT ATTACHMENT CABLE NOT CONNECTED
		X'35A0'	(detailed data qualifier) LOCAL MODEM EXTERNAL CLOCK NOT PROVIDED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'35A1'	(detailed data qualifier) REMOTE MODEM EXTERNAL CLOCK NOT PROVIDED *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		X'35A2'	2-WIRE, HALF DUPLEX COUPLER ON THE (detailed data qualifier) LOCAL MODEM ON A 4-WIRE, FULL DUPLEX LINE *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			X'35A3'	(detailed data qualifier) MODEMS SPEED MISMATCH *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
			X'35A4'	(detailed data qualifier) DSU/CSU'S SPEED MISMATCH *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSUs belong.
			X'35A5'	(detailed data qualifier) INCOMPATIBLE MODEMS *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
			X'35A6'	SENDING NODE AND (detailed data qualifier) MODEMS CONFIGURATION MISMATCH *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
			X'35A7'	SENDING NODE AND (detailed data qualifier) DSU/CSU'S CONFIGURATION MISMATCH *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSUs belong.
		X'3700'	LAN CONFIGURATION ERROR: A configuration parameter for a local-area network has been specified incorrectly	
			X'3701'	FILE SERVER ASSIGNED DUPLICATE LAN LOGICAL ADDRESS
			X'3702'	FILE SERVER ASSIGNED INCORRECT LAN LOGICAL IDENTIFIER
			X'3704'	TOKEN-RING DUPLICATE STATION ADDRESS ASSIGNED
			X'3724'	CSMA/CD DUPLICATE STATION ADDRESS ASSIGNED
			X'3732'	DUPLICATE FDDI ADDRESS ASSIGNED
		X'3800'	LPDA CONFIGURATION ERROR: A configuration parameter for an LPDA link has been specified incorrectly	
			X'38A0'	(detailed data qualifier) LOCAL MODEM ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
			X'38A1'	(detailed data qualifier) LOCAL DSU/CSU ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
			X'38A2'	(detailed data qualifier) REMOTE MODEM ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
			X'38A3'	(detailed data qualifier) REMOTE DSU/CSU ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'38A4'	(detailed data qualifier) LOCAL MODEM LPDA-2 DISABLED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'38A5'	(detailed data qualifier) LOCAL DSU/CSU LPDA-2 DISABLED *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'38A6'	(detailed data qualifier) LOCAL MODEM NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'38A7'	(detailed data qualifier) REMOTE MODEM NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		X'38A8'	(detailed data qualifier) LOCAL DSU/CSU NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'38A9'	(detailed data qualifier) REMOTE DSU/CSU NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
		X'38AA'	(detailed data qualifier) LOCAL MODEM CONFIGURED AS SECONDARY OR TRIBUTARY *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'38AB'	(detailed data qualifier) LOCAL DSU/CSU CONFIGURED AS SECONDARY OR TRIBUTARY *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'38AC'	(detailed data qualifier) LOCAL MODEM CONFIGURED AS CONTROL *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'38AD'	(detailed data qualifier) LOCAL DSU/CSU CONFIGURED AS CONTROL *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'38C0'	SPEED MISMATCH BETWEEN (detailed data qualifier) AND (detailed data qualifier) *Note:* The qualifiers identify the link segment levels (LSL) where the speed mismatch is.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			X'38C1'	(detailed data qualifier) LOCAL MODEM HAS A 2-WIRE COUPLER AND THE (detailed data qualifier) REMOTE MODEM HAS A 4-WIRE COUPLER *Note:* The qualifiers identify the link segment levels (LSL) where the modems belong.
		X'8000'	CONFIGURATION ERROR: A system or device generation or customization parameter has been specified incorrectly or is inconsistent with the actual configuration.	
			X'8001'	TOPOLOGY STORAGE EXCEEDED: The node table is full.
			X'8002'	INSUFFICIENT STORAGE FOR DIRECTORY SERVICES: Deadlock detected between directory services components in two nodes.
			X'8003'	COMMUNICATIONS SUBSYSTEM DEFINITION: Deadlock detected between directory services components in two nodes.
			X'8004'	REMOTE NODE DEFINITION ERROR: BIND received from an end-node that this network-node does not serve.
			X'8005'	SYSTEM DEFINITION ERROR: BIND received from an end-node that this network-node does not serve.
			X'8006'	LOCAL MODEM CLOCK SIGNAL NOT PROVIDED TO DTE
			X'8011'	SUBNET/SYSTEM CONFIGURATION FILE EMPTY
			X'8012'	BUFFER ALLOCATION ERROR
			X'8013'	BASE LOGGER NODE INFORMATION BLOCK NOT FOUND
			X'8014'	COMMUNICATION NODE INFORMATION BLOCK NOT FOUND
			X'8015'	CONFIGURATION NODE INFORMATION BLOCK NOT FOUND
			X'8016'	COMMUNICATION CONTROL BLOCK NOT DEFINED IN RESOURCE DIRECTORY
			X'8017'	DESTINATION RESOURCE NOT DEFINED IN CONTROL BLOCK FILE
			X'8018'	DATA TRANSPARENCY NOT ENABLED
			X'8019'	DESTINATION RESOURCE NOT DEFINED IN RESOURCE DIRECTORY
			X'801A'	DESTINATION RESOURCE NOT PURGED
			X'801B'	DESTINATION RESOURCE SERVER NOT DEFINED
			X'801C'	DEVICE NAME NOT FOUND IN DEVICE COMMUNICATIONS SERVER TABLE
			X'801D'	FILE NOT FOUND IN PATH SPECIFIED
			X'801E'	INBOUND SERVER NOT DEFINED IN RESOURCE DIRECTORY

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			X'801F'	INCORRECT COMMUNICATIONS DRIVER INSTALLATION
			X'8020'	INVALID OR UNSUPPORTED CODE PAGE IN SYSTEM CONFIGURATION FILE
			X'8021'	INVALID COUNTRY CODE IN SYSTEM CONFIGURATION FILE
			X'8022'	INVALID CONTROL BLOCK TYPE
			X'8023'	LOGICAL DEVICE NOT FOUND
			X'8024'	MESSAGE FILE ERROR
			X'8025'	MESSAGE NOT FOUND
			X'8026'	MAXIMUM NUMBER OF THREADS EXCEEDED
			X'8027'	DATABASE NODE INFORMATION BLOCK NOT LOADED
			X'8028'	NODE INFORMATION BLOCK NOT FOUND
			X'8029'	NOT ENOUGH MEMORY IN CONTROL BLOCK MEMORY POOL
			X'802A'	NOT ENOUGH MEMORY IN MESSAGE MEMORY POOL
			X'802B'	NOT ENOUGH MEMORY IN CO-PROCESSOR CONTROL BLOCK MEMORY POOL
			X'802C'	NOT ENOUGH MEMORY IN CO-PROCESSOR MESSAGE MEMORY POOL
			X'802D'	NOT ENOUGH MEMORY IN CO-PROCESSOR WORK MEMORY POOL
			X'802E'	NOT ENOUGH MEMORY IN WORK MEMORY POOL
			X'802F'	NOT ENOUGH MEMORY ON VIRTUAL DISK
			X'8030'	NOT ENOUGH MEMORY ON VIRTUAL MACHINE
			X'8031'	NO MEMORY AVAILABLE FOR REQUEST
			X'8032'	NETWORK CONTROL NODE INFORMATION BLOCK NOT FOUND
			X'8033'	ORIGINATOR NODE NOT FOUND IN SYSTEM CONFIGURATION FILE
			X'8034'	ORIGINATOR RESOURCE NOT CONNECTED
			X'8035'	ORIGINATOR RESOURCE NOT DEFINED
			X'8036'	OUTBOUND SERVER NOT DEFINED IN RESOURCE DIRECTORY
			X'8037'	ORIGINATOR NODE NOT FOUND IN SUBNET CONFIGURATION FILE
			X'8038'	PHYSICAL RESOURCE NOT DEFINED FOR ALIAS NAME CONTROL BLOCK
			X'8039'	PRINTER CONTROL BLOCK STOPPED AND PURGED
			X'803A'	CO-PROCESSOR DISPATCHER NOT LOADED
			X'803B'	RESOURCE INFORMATION IN RESOURCE DIRECTORY DOES NOT MATCH TABLES AND QUEUES DEFINITION FILE

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'803C'	REMOTE CONTROL BLOCK NOT A REMOTE COMMUNICATION CONTROL BLOCK
		X'803D'	REMOTE CONTROL BLOCK NOT DEFINED IN COMMUNICATION CONTROL BLOCK
		X'803E'	RESOURCE NOT DEFINED IN SYSTEM
		X'803F'	RESOURCE NOT DEFINED IN TABLES AND QUEUES DEFINITION FILE
		X'8040'	RESOURCE NOT STARTED
		X'8041'	SECURITY NOT ENABLED IN NODE INFORMATION BLOCK CONFIGURATION SEGMENT
		X'8042'	SERVER RESOURCE NOT DEFINED IN RESOURCE DIRECTORY
		X'8043'	SYSTEM TEMPLATE FILE ERROR
		X'8044'	TASK NUMBER NOT AVAILABLE
		X'8045'	TASK NUMBER NOT FOUND
		X'8046'	TIMEOUT VALUE SET TOO LOW IN APPLICATION CONTROL BLOCK
		X'8047'	TIMEOUT VALUE SET TOO LOW IN DATABASE CONTROL BLOCK
		X'8048'	TIMEOUT VALUE SET TOO LOW IN LINE CONTROL BLOCK
		X'8049'	TIMEOUT VALUE SET TOO LOW IN REQUEST
		X'804A'	TIMER NOT AVAILABLE
		X'804B'	VIDEO NODE INFORMATION BLOCK NOT FOUND
		X'804C'	DESTINATION RESOURCE NOT DEFINED IN CONTROL BLOCK POOL
		X'804D'	CONFIGURATION PROBLEM WITH DATABASE COMPONENT
		X'804E'	ALTERNATE SYSTEM TRACE FILE CONFIGURATION
		X'8050'	INVALID BRIDGE CUSTOMIZATION DATA
		X'8051'	ERROR ACCESSING SECURED RESOURCE TABLE
		X'8052'	ERROR ACCESSING SECURED VIEW IDENTIFICATION TABLE
		X'8053'	ERROR STARTING MONITOR APPLICATION CONTROL BLOCK
		X'80A0'	(detailed data qualifier) NOT DEFINED AS SYSTEM TRACE FILE *Note:* The qualifier identifies the file name.
		X'80A1'	(detailed data qualifier) NOT DEFINED IN RESOURCE DIRECTORY *Note:* The qualifier identifies the resource name.
		X'80A2'	(detailed data qualifier) NODE INFORMATION BLOCK NOT FOUND *Note:* The qualifier identifies the segment number.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'80A3'	(detailed data qualifier) NOT FOUND IN PATH SPECIFIED *Note:* The qualifier identifies the file name.
		X'80A4'	CONTROL NODE (detailed data qualifier) NOT FOUND IN SYSTEM CONFIGURATION FILE *Note:* The qualifier identifies the node name.
		X'80A5'	COUNTRY CODE INFORMATION IN SYSTEM CONFIGURATION FILE DOES NOT MATCH (detailed data qualifier) *Note:* The qualifier identifies the file name.
		X'80A6'	CODE PAGE INFORMATION IN SYSTEM CONFIGURATION FILE DOES NOT MATCH (detailed data qualifier) *Note:* The qualifier identifies the file name.
		X'80A7'	CONTROL NODE (detailed data qualifier) NOT FOUND IN SUBNET FILE *Note:* The qualifier identifies the node name.
		X'80A8'	DUPLICATE (detailed data qualifier) DEFINED IN CONTROL BLOCK FILE *Note:* The qualifier identifies the resource name.
		X'80A9'	DUPLICATE REGISTRATION NAME (detailed data qualifier) ON DESTINATION COMMUNICATION CONTROL BLOCK *Note:* The qualifier identifies the communication control block name. *Note:* The qualifier identifies the return code.
		X'80AB'	LOGON PANEL (detailed data qualifier) NOT FOUND *Note:* The qualifier identifies the logon panel name.
		X'80AC'	NO MEMORY AVAILABLE FOR (detailed data qualifier) *Note:* The qualifier identifies one of the following: Buffer name, Utility name, or function name. *Note:* The qualifier identifies the node name.
		X'80AE'	NODE (detailed data qualifier) IS IN BOTH SYSTEM AND SUBNET CONFIGURATION FILES *Note:* The qualifier identifies the node name.
		X'80AF'	NODE (detailed data qualifier) NOT FOUND IN CONFIGURATION FILE *Note:* The qualifier identifies the node name.
		X'80B0'	SUBSERVIENT NODE (detailed data qualifier) NOT FOUND IN SUBNET FILE *Note:* The qualifier identifies the node name.
		X'80B1'	SYSTEM NODE (detailed data qualifier) NOT FOUND IN SUBNET FILE *Note:* The qualifier identifies the node name.
		X'80B2'	SYSTEM NODE (detailed data qualifier) NOT FOUND IN SYSTEM FILE *Note:* The qualifier identifies the node name.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'80B3'	SESSION NOT ESTABLISHED WITH (detailed data qualifier) *Note:* The qualifier identifies the node name with which communications could not be established.
		X'80C0'	CONFIGURATION OBJECT DOES NOT EXIST (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the configuration object's name and type, respectively.
		X'80C1'	STORAGE CONFIGURATION ERROR (detailed data qualifier) (detailed data qualifier): The actual device configuration does not match the configuration records. *Note:* The first qualifier identifies the configuration object/record. The second qualifier identifies the incorrect configuration parameter.
		X'80C2'	LOCAL WORK STATION CONFIGURATION ERROR (detailed data qualifier) (detailed data qualifier): The actual controller and/or work station configuration does not match the configuration records. *Note:* The first qualifier identifies the configuration object/record. The second qualifier identifies the incorrect configuration parameter.
		X'80C3'	REMOTE WORK STATION CONFIGURATION ERROR (detailed data qualifier) (detailed data qualifier): The actual controller and/or work station configuration does not match the configuration records. *Note:* The first qualifier identifies the configuration object/record. The second qualifier identifies the incorrect configuration parameter.
		X'80C4'	COMMUNICATION CONFIGURATION ERROR (detailed data qualifier) (detailed data qualifier): The actual communication configuration does not match the configuration records. *Note:* The first qualifier identifies the configuration object/record. The second qualifier identifies the incorrect configuration parameter.
		X'80C5'	(detailed data qualifier) NAME NOT DEFINED FOR (detailed data qualifier) *Note:* The first qualifier identifies the resource type. The second qualifier identifies the resource name.
		X'80C6'	LOCAL COMMUNICATION CONTROL BLOCK (detailed data qualifier) NOT DEFINED FOR COMMUNICATION ACCESS CONTROL BLOCK (detailed data qualifier) *Note:* The first qualifier identifies the local communication control block name. The second qualifier identifies the communication access control block name.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'80C7'	LOCAL COMMUNICATION CONTROL BLOCK (detailed data qualifier) NOT STARTED FOR COMMUNICATION ACCESS CONTROL BLOCK (detailed data qualifier) *Note:* The first qualifier identifies the local communication control block name. The second qualifier identifies the communication access control block name.
		X'80C8'	REMOTE COMMUNICATION CONTROL BLOCK (detailed data qualifier) NOT DEFINED FOR COMMUNICATION ACCESS CONTROL BLOCK (detailed data qualifier) *Note:* The first qualifier identifies the remote communication control block name. The second qualifier identifies the communication access control block name.
		X'80C9'	REMOTE COMMUNICATION CONTROL BLOCK (detailed data qualifier) NOT STARTED FOR COMMUNICATION ACCESS CONTROL BLOCK (detailed data qualifier) *Note:* The first qualifier identifies the remote communication control block name. The second qualifier identifies the communication access control block name.
		X'E000'–X'EFFF'	Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.
		Note: The following code points specify extended messages, that provide additional information on one or more install causes that have already been specified. An Alert receiver that displays only default text provides no display for these code points.	
		X'F000'	(no display): Additional message data
		X'F004'	IDENTITY OF OFFENDING SERVER IS UNKNOWN
		X'F0A3'	(detailed data qualifier) MEMORY ALLOCATION REQUESTS FAILED *Note:* The qualifier identifies the number of failed allocation requests.
		X'F0A4'	(detailed data qualifier) DUPLICATES LOGICAL LAN ADDRESS OF ALERT ORIGINATOR *Note:* The qualifier identifies the file server.
		X'F0A5'	(detailed data qualifier) CURRENTLY USED *Note:* The qualifier identifies a resource.
		X'F0C0'	ERROR PROCESSING (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the system message code and the return code, respectively.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content	
		X'F0D0'	FILE SERVER AT (detailed data qualifier) ASSIGNED (detailed data qualifier) — ALERT ORIGINATOR ASSIGNMENT ON SAME NETWORK IS (detailed data qualifier) *Note:* The qualifiers identify the following: Q1 = NODE Q2 = NETWORK Q3 = NETWORK
		X'F0D1'	ERROR (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the following: Q1 = Database table name Q2 = Node name Q3 = Return code
		X'F800'	(no display): Additional message data *Note:* The X'F8xx' range is used for additional messages that are identical for User, Install, and Failure Causes.
		X'F8A0'	PROBLEM DETECTED BY (detailed data qualifier) *Note:* The qualifier identifies the file server.
		X'F8C0'	FAILING COMPONENT IS IDENTIFIED BY (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the failing component by means of its logical location, e.g., its port number and device address.
		X'F8C1'	REQUEST ORIGINATED FROM (detailed data qualifier) ON (detailed data qualifier) *Note:* The qualifiers identify the node and network.
		X'F8D0'	PROBLEM IS RELATED TO THE CONTROLLER LOCATED AT (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the controller location as follows: Q1 = RACK Q2 = UNIT (within a rack) Q3 = CARD SLOT (within a unit)
		X'F8E0'	FAILING COMPONENT IS IDENTIFIED BY (sf83 product text)
		X'FFFF'	Reserved

Failure Causes (X'96') Alert MS Subvector

This subvector transports code points for stored text detailing the probable failure causes for the Alert condition and the recommended actions to be taken in connection with these causes. It may also transport additional detailed data, to be inserted into the text indexed by the failure cause and/or recommended action code points.

Failure Causes (X'96') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Failure Causes subvector
1		Key: X'96'
2–p		Two or more subfields containing failure cause data, as described below for keys X'00'–X'7F' and in "Network Alert (X'0000') Common Subfields" on page 9-131 for keys X'80'–X'FE'. X'01' Failure Causes X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended *Note:* Subfields X'01' and X'81' are always present and the X'01' always precedes the X'81'. Depending on the code points present in the X'01' and X'81' subfields: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

Failure Causes (X'01') Failure Causes Subfield

This subfield contains one or more code points denoting probable failure causes of the Alert condition, listed in order of decreasing probability. A failure cause is defined to be a condition resulting from the failure of a resource.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Failure Causes subfield
1		Key: X'01'

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content
2−q		Two-byte failure cause code points. Each code point provides an index to predefined text, describing the failure cause, that is displayed at the Alert receiver. An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are listed below. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules). The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the X'96' subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. **Note:** The X'82' and X'85' subfields *cannot* both be used in the same subvector. The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. The third digit of each failure cause code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows: X'xx0x'−X'xx9x': No detailed data qualifiers. X'xxAx'−X'xxBx': One detailed data qualifier. X'xxCx': Two detailed data qualifiers. X'xxDx': Three detailed data qualifiers. X'xxEx': One X'83' subfield. X'xxFx': One X'84' subfield. Defined failure cause codes are:

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'0000'	PROCESSOR: The equipment used to interpret and process programmed instructions
		X'0001'	MOSS HARDWARE: A hardware failure in MOSS (Maintenance and Operation Subsystem), the service processor for a communication controller
		X'0002'	MOSS MICROCODE: A microcode failure in MOSS (Maintenance and Operation Subsystem), the service processor for a communication controller
		X'0003'	PROCESSOR SWITCH: A component within a hardware product used to switch busses and the resources attached to them among processors
		X'0004'	CONTROL PANEL
		X'0005'	SYSTEM I/O BUS
		X'0006'	PROCESSOR MACHINE CHECK: A failure in the processor which precludes it from continuing operation.
		X'0007'	CARD ENCLOSURE AND/OR BOARD: The enclosure and boards used to hold circuit cards and provide power and/or signal connections for the cards.
		X'0008'	VECTOR PROCESSOR: The vector processing element associated with a central processing unit (CPU) has failed and is in a reserved state.
		X'0009'	SYSTEM CHECK STOP
		X'000A'	SERVICE PROCESSOR: A maintenance, service and support processor; sometimes called a process controller
		X'0010'	LAN MANAGER PROCESSOR
		X'0011'	PRINTER SERVER: A network component that controls the operation of a printer *Note:* In the current implementation, the printer server is a PC that stands between a printer and the host applications that communicate with it.
		X'0013'	OPTICAL SYSTEM BUS CONTROLLER
		X'0020'	EXCESSIVE LOAD ON PROCESSOR: The processor is not able to keep up with incoming requests for service. Internal buffers may be filled with queued tasks and not able to accept more requests, which may be asynchronous, and thus, discarded.
		X'0030'	SYSTEM MICROCODE: The specific microcode was not identified.
		X'0031'	SYSTEM STORAGE MICROCODE *Note:* See also code point X'0421' (STORAGE CONTROLLER MICROCODE)
		X'0032'	SYSTEM DISPLAY MICROCODE *Note:* See also code point X'0422' (WORK STATION CONTROLLER MICROCODE)
		X'0033'	SYSTEM COMMUNICATION MICROCODE *Note:* See also code point X'0423' (COMM SUBSYSTEM CONTROLLER MICROCODE)

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'0034'	SYSTEM PRINTER MICROCODE
		X'0050'	TERMINAL EMULATOR SUBSYSTEM MICRO-CODE
		X'00E1'	(sf83 product text) PROCESSOR
		X'0100'	STORAGE: The random access memory (RAM) or read only memory (ROM) accessible by a processor and by peripheral devices
		X'0101'	MAIN STORAGE: Storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing
		X'0102'	AUXILIARY STORAGE: Storage that can not be directly addressed by a processor, such as external or secondary storage
		X'0103'	MAIN STORAGE MACHINE CHECK: A failure in main storage which precludes it from continuing operation.
		X'0104'	EXPANDED STORAGE: A specific type of auxiliary storage used for data and program paging
		X'0108'	DASD CACHE
		X'0110'	STORAGE CONTROL: The component that controls access to storage
		X'0111'	NUMBER OF LAN MANAGEMENT FRAMES RECEIVED EXCEEDS BUFFER CAPACITY: Management frames from stations on a local LAN token-ring or bus are arriving faster than the LAN management server can process them
		X'01E1'	(sf83 product text) MAIN STORAGE
		X'0200'	POWER SUBSYSTEM: The subsystem within a hardware product that provides electrical power to the different components within the product that require it
		X'0201'	INTERNAL POWER UNIT: An element of the power subsystem providing electrical power to a specific component
		X'0202'	INTERNAL POWER CONTROL UNIT: An element of the power subsystem that controls the internal power units
		X'0203'	POWER CABLE
		X'0204'	POWER CORD
		X'0205'	POWER SUBSYSTEM PROCESSOR: A processor within the power subsystem responsible for its operation
		X'0210'	BATTERY
		X'0211'	MOSS BATTERY
		X'0212'	COMMUNICATION WITH UNINTERRUPTIBLE POWER SUPPLY DISRUPTED
		X'0220'	MAIN AC POWER SUPPLY
		X'02C0'	INTERNAL POWER UNIT FOR (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the adapter numbers of the adapters served by the failing internal power unit.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content
		X'0230' POWER DISTRIBUTION UNIT
		X'0231' MOTOR GENERATOR
		X'0240' INTERNAL CLOCK: A mechanism which keeps time.
		X'0300' COOLING OR HEATING SUBSYSTEM: The subsystems within a hardware product responsible for maintaining a temperature at which the product can operate
		X'0301' COOLING FAN
		X'0302' AIR FILTER
		X'0310' AIR FLOW DETECTOR
		X'0311' THERMAL DETECTOR
		X'0320' COOLANT DISTRIBUTION UNIT: A unit that distributes chilled water for cooling purposes, usually circulated in a closed system, has failed.
		X'0321' THERMAL LIMITS EXCEEDED: The acceptable thermal limits for normal operation have been exceeded.
		X'0400' SUBSYSTEM CONTROLLER: A unit within a subsystem that interfaces between a processor and the devices in the subsystem *Note:* See Alert Description X'1600', SUBSYSTEM FAILURE, for descriptions of the particular subsystems mentioned here.
		X'0401' STORAGE CONTROLLER
		X'0402' WORKSTATION CONTROLLER
		X'0403' COMMUNICATIONS SUBSYSTEM CONTROLLER *Note:* Contrast this code point with X'3111', COMMUNICATION CONTROLLER. A communication controller is typically a stand-alone node within a network, for example, a 3725; a communication subsystem controller is typically a component within a larger node that provides for the node's communication with nodes remote from it.
		X'0411' INTERMITTENT STORAGE CONTROLLER ERROR
		X'0412' INTERMITTENT WORKSTATION CONTROLLER ERROR
		X'0413' INTERMITTENT COMMUNICATIONS SUBSYSTEM CONTROLLER ERROR
		X'0421' LOADABLE STORAGE CONTROLLER MICROCODE
		X'0422' LOADABLE WORKSTATION CONTROLLER MICROCODE
		X'0423' LOADABLE COMMUNICATIONS SUBSYSTEM CONTROLLER MICROCODE
		X'0441' STORAGE CONTROLLER INTERFACE: The interface between a storage controller and the main processor in its node
		X'0442' WORKSTATION CONTROLLER INTERFACE: The interface between a workstation controller and the main processor in its node

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'0443'	COMMUNICATIONS SUBSYSTEM CONTROLLER INTERFACE: The interface between a communication subsystem controller and the main processor in its node.
		X'0444'	INTERCONNECT CONTROLLER
		X'0500'	SUBSYSTEM: A set of components that jointly provide a specified function; typically a subsystem includes a controller, one or more interface adapters, physical connection media, and attached devices *Note:* See Alert Description X'1600', SUBSYSTEM FAILURE, for descriptions of the particular subsystems mentioned here
		X'0501'	STORAGE SUBSYSTEM
		X'0502'	WORKSTATION SUBSYSTEM
		X'0503'	COMMUNICATIONS SUBSYSTEM
		X'0506'	CHANNEL SUBSYSTEM: A channel subsystem or a shared element within a channel subsystem has failed. The failing resource consists of more than just a single channel path.
		X'0507'	CALLER PROCESSING SUBSYSTEM: The telephone interface subsystem which processes touch tone input and voice output. It resides between the telephone system and an application processor.
		X'0508'	SPEECH SYNTHESIS SUBSYSTEM: The subsystem which maps host data and programmed phrases to digitized speech for output to the caller.
		X'1000'	SOFTWARE PROGRAM: A program implemented in software, as distinguished from one implemented in microcode
		X'1001'	APPLICATION PROGRAM
		X'1002'	LOADABLE SOFTWARE MODULE
		X'1004'	ENCRYPTION PROGRAM
		X'1005'	VOICE RESPONSE UNIT PROGRAM
		X'1006'	LAN OVER WAN COMMUNICATIONS PROGRAM
		X'100A'	COMMUNICATIONS PROGRAM ABNORMALLY TERMINATED
		X'100F'	PROGRAM CHECK: An error in a program, detected by a processor's circuitry or microcode or by another software program, that would cause erroneous or catastrophic results if allowed to execute uncorrected.
		X'1010'	HOST PROGRAM: A program running in a host processor that is a primary or controlling program in a system
		X'1011'	PRINTER SERVER PROGRAM: A program running in a printer server that controls a printer *Note:* See also Failure Cause X'0011', PRINTER SERVER.
		X'1012'	SOFTWARE DEVICE DRIVER: A program designed to control a device.
		X'1013'	COMMUNICATIONS/LAN DEVICE DRIVER

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'1021'	COMMUNICATION CONTROLLER CONTROL PROGRAM: A software program that is designed to schedule and supervise the execution of programs in a communication controller
		X'1022'	COMMUNICATIONS PROGRAM: A software program designed to provide direct assistance to a node in communicating with other nodes
		X'1023'	COMMUNICATIONS PROGRAM IN REMOTE NODE
		X'1024'	COMMUNICATIONS ACCESS METHOD: A software program in a host that provides access to a telecommunications network
		X'1025'	COMMUNICATIONS PROGRAM IN LOCAL NODE
		X'1030'	LAN MANAGER PROGRAM: The software program in a LAN manager
		X'1031'	LAN MANAGEMENT SERVER: A data collection and distribution point for a single LAN token-ring or bus. A LAN management server forwards data received from stations on its LAN token-ring or bus and possibly results from preliminary analysis performed by the server (on that data) to the LAN manager. LAN management servers also send data to stations on their LAN token-rings or busses *Note:* The LAN management servers that are currently defined are Ring Error Monitor (REM), Configuration Report Server (CRS), Ring Parameter Server (RPS), LAN Bridge Server (LBS), and LAN Reporting Mechanism (LRM).
		X'1032'	RING ERROR MONITOR: The LAN management server responsible for receiving and processing error reports from the stations on its token-ring
		X'1040'	I/O ACCESS METHOD ERROR: An error in a program that provides access to I/O (e.g., DASD, tape, terminals, printer, telecommunications network, etc.).
		X'1050'	AGENT ERROR: An agent has experienced a permanent error condition. This is an internal error in a software component.
		X'1051'	COMMAND NOT RECOGNIZED: An internal command encountered at a software component was not recognized. This is an internal error in a software component.
		X'1052'	CONVERSATION PROTOCOL ERROR: A conversation protocol error was encountered by a software component. This is an internal error in a software component.
		X'1053'	DATA DESCRIPTOR ERROR: The description of the data defined in a distributed unit of work was in error. This is an internal error in a software component.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'1054'	INVALID DATA STRUCTURE ERROR: The data structure encountered by a software component was in error. This is an internal error in a software component.
		X'1055'	INVALID CURSOR STATE: The cursor state during a query operation against a relational data base was invalid. This is an internal error in a software component.
		X'1056'	RELATIONAL DATABASE ACCESS ERROR: An error was encountered while accessing a relational database. This is an internal error in a software component.
		X'1057'	BLOCKING PROTOCOL ERROR: Data blocking protocols were violated. This is an internal error in a software component.
		X'1058'	CHAINING PROTOCOL ERROR: Data chaining protocols were violated. This is an internal error in a software component.
		X'1059'	RELATIONAL DATABASE PACKAGE IS INOPERATIVE
		X'105A'	RELATIONAL DATABASE SNAPSHOT DATA OUT OF DATE
		X'105B'	SNAPSHOT TABLE TIMESTAMP OUT OF SYNC WITH BASE TABLE
		X'105C'	SNAPSHOT TABLE EXISTENCE MISMATCH WITH BASE TABLE
		X'105D'	SNAPSHOT TABLE OPTIONS MISMATCH WITH BASE TABLE
		X'1061'	MESSAGE RECEIVED IS NOT AN ACKNOWLEDGEMENT MESSAGE
		X'1062'	BUFFER ALLOCATION ERROR
		X'1063'	ALL COMPONENT RESOURCE EXECUTION MODULES NOT STOPPED SUCCESSFULLY
		X'1065'	COMMUNICATIONS DRIVER ERROR
		X'1066'	CONTROL BLOCK CHAINING ERROR
		X'1067'	ENCODING/DECODING ERROR
		X'1068'	DESTINATION RESOURCE SERVER STOPPED
		X'1069'	DESTINATION RESOURCE STOPPED
		X'1070'	DEVICE COMMAND ERROR
		X'1071'	DEVICE COMMUNICATIONS SERVER TABLE ERROR
		X'1072'	DEVICE DATA MANAGEMENT SUPPORT SERVER NOT FOUND IN PROCESSOR
		X'1073'	DESTINATION RESOURCE NOT DEFINED IN CONTROL BLOCK POOL
		X'1074'	DESTINATION RESOURCE NOT STOPPED
		X'1075'	DESTINATION RESOURCE PAUSED
		X'1076'	DESTINATION RESOURCE PURGED
		X'1077'	DESTINATION RESOURCE SERVER PAUSED
		X'1078'	DESTINATION RESOURCE SERVER PURGED
		X'1079'	DISTRIBUTION SNAPSHOT FILE ERROR

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'107A'	ERROR ACCESSING FILE
		X'107B'	DEVICE LOCKED BY ANOTHER TASK
		X'107C'	DEVICE LOCKED BY ANOTHER USER
		X'107D'	DISK QUEUE SERVER RESOURCE STOPPED
		X'107E'	DISTRIBUTION FAILED
		X'107F'	DISTRIBUTION SESSION NOT CREATED
		X'1080'	ERROR GENERATING TEMPLATE FOR DATA DEFINITION
		X'1081'	ERROR CONTROL DATA LITERAL FILE
		X'1082'	ERROR OPENING FILE
		X'1083'	ERROR ACCESSING SECURED RESOURCE TABLE
		X'1084'	ERROR CLOSING FILE
		X'1085'	ERROR CLOSING TEMPORARY JOURNAL FILE
		X'1086'	ERROR DELETING TEMPORARY JOURNAL FILE
		X'1087'	ERROR RENAMING TEMPORARY JOURNAL FILE
		X'1088'	ERROR SENDING MAILBOX SERVICES MESSAGE TO DEVICE COMMUNICATION PROTOCOL PROGRAM
		X'1089'	FILE SUPPORT ERROR
		X'108A'	FILE SUPPORT RESOURCE NOT STARTED
		X'108B'	THREAD CREATION ERROR
		X'108C'	FILE CONTROL TABLE FULL
		X'108D'	FILE FULL
		X'108E'	FILE IS CORRUPT
		X'108F'	INBOUND SERVER RESOURCE STOPPED
		X'1090'	INCORRECT ENVIRONMENT STATUS RECEIVED FROM ORIGINATOR
		X'1091'	INPUT/OUTPUT ERROR – JOURNAL FILE
		X'1092'	DEVICE DATA MANAGEMENT SUPPORT PROGRAM ERROR
		X'1093'	INVALID DEVICE NAME
		X'1094'	INVALID LINE NAME
		X'1095'	INVALID LOCK ON RECORDS IN FILE
		X'1096'	INVALID MEMORY ADDRESS
		X'1097'	INVALID MEMORY POOL
		X'1098'	INVALID MEMORY REQUEST SIZE
		X'1099'	JOURNAL LOCKED
		X'109A'	INVALID ACKNOWLEDGEMENT IN REQUEST DATA STRUCTURE
		X'109B'	INVALID COMMAND CODE
		X'109C'	ERROR DURING FILE SEEK
		X'109D'	ERROR DURING BUFFER FLUSH
		X'109E'	INTERCONNECT CONTROLLER PROGRAM
		X'109F'	COMMAND VIOLATION
		X'10A0'	SOFTWARE SUBSYSTEM (detailed data qualifier) *Note:* The qualifier identifies the name of the software subsystem.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'10A1'	UNABLE TO BUILD ALERT REQUESTED BY (detailed data qualifier) *Note:* An Alert builder utility could not complete a request from the program identified by the qualifier.
		X'10A2'	(detailed data qualifier) FILE ERROR *Note:* The qualifier identifies the file name.
		X'10A3'	(detailed data qualifier) IS CORRUPT *Note:* The qualifier identifies the corrupt entity.
		X'10A4'	(detailed data qualifier) NOT FOUND *Note:* The qualifier identifies the entity not found.
		X'10A5'	ERROR SENDING NOTIFY NEW NODE MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10A6'	ERROR SENDING OPEN ACKNOWLEDGEMENT MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10A7'	ERROR SENDING OPEN SESSION MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10A8'	ERROR SENDING QUERY NODE MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10A9'	ERROR SENDING QUERY REPLY MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10AA'	ERROR SENDING REQUEST HEARTBEAT MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10AB'	ERROR SENDING ADD NODE MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10AC'	ERROR SENDING DELETE NODE MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10AD'	ERROR SENDING GET NEW NODE MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10AE'	ERROR SENDING HEARTBEAT MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10AF'	ERROR SENDING LOAD APPLICATIONS MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10B0'	ERROR SENDING NODE ABORT MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'10B1'	ERROR SENDING SUBNET ABORT MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10B2'	ERROR SENDING SWITCH REPLY MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10B3'	ERROR SENDING SWITCH NODE MESSAGE — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10B4'	SUBNET CONTROL NODE (detailed data qualifier) SWITCH FAILED *Note:* The qualifier identifies the node name.
		X'10B5'	SYSTEM CONTROL NODE (detailed data qualifier) SWITCH FAILED *Note:* The qualifier identifies the node name.
		X'10B6'	STRUCTURED QUERY LANGUAGE ERROR (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10B7'	DATA TRANSPARENCY ERROR — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10B8'	DISK ERROR CLOSING (detailed data qualifier) *Note:* The qualifier identifies the file name.
		X'10B9'	DISK ERROR READING (detailed data qualifier) *Note:* The qualifier identifies the file name or drive.
		X'10BA'	CODE PAGE ERROR — (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'10BB'	COMMUNICATIONS DRIVER SENT INCORRECT CARD NUMBER (detailed data qualifier) *Note:* The qualifier identifies the card number.
		X'10BC'	COMMUNICATIONS DRIVER SENT INCORRECT CONTROL BLOCK TYPE (detailed data qualifier) *Note:* The qualifier identifies the control block.
		X'10BD'	CONFIGURATION PACKAGE ACKNOWLEDGE-MENT NOT RECEIVED FROM (detailed data qualifier) *Note:* The qualifier identifies the node name.
		X'10BE'	CONTROL NODE (detailed data qualifier) FAILED TO RESPOND *Note:* The qualifier identifies the node name.
		X'10BF'	SOFTWARE SUBTASK (detailed data qualifier) *Note:* The qualifier identifies the name of the software subtask.
		X'10C1'	(detailed data qualifier) UNABLE TO FORWARD ALERT ORIGINALLY SENT BY (detailed data qualifier) *Note:* The first qualifier identifies the CP that could not forward the Alert originally sent by the CP named in the second qualifier.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'10C2'	COMMUNICATION PROTOCOL PROGRAM ERROR (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the primary return code. The second qualifier identifies the secondary return code.
		X'10C3'	JOURNAL ERROR (detailed data qualifier) CONNECTION NAME (detailed data qualifier) *Note:* The first qualifier identifies the journal resource variable. The second qualifier identifies the connection name.
		X'10C4'	OPERATION ERROR (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the operation type. The second qualifier identifies either a system call or a return code.
		X'10C5'	CANNOT CREATE OPERATOR INTERFACE LOG FILE (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the file name. The second qualifier identifies the return code.
		X'10C6'	CANNOT WRITE TO OPERATOR INTERFACE LOG FILE (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the file name. The second qualifier identifies the return code.
		X'10C7'	ERROR STARTING (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the resource name and return code, respectively.
		X'10C8'	ERROR OPENING (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the resource name and the operating system return code, respectively.
		X'10C9'	ERROR READING (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the member name and the return code, respectively.
		X'10CA'	ERROR SEEKING (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the file name and the operating system return code, respectively.
		X'10CB'	ERROR WAITING FOR REPLY FROM (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the resource name and the return code, respectively.
		X'10CC'	ERROR FREEING BUFFER AT (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the address and the return code, respectively.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
			X'10CD' ERROR SENDING TO (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the resource name and the return code, respectively.
			X'10CE' DEVICE ERROR — (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the device name. The second qualifier identifies either a system call or a return code.
			X'10D0' ERROR ON EVENT SCANNER (detailed data qualifier) CONNECTION NAME (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the event scanner process. The second qualifier identifies connection name, and the third qualifier identifies the return code.
			X'10E1' SOFTWARE PROGRAM (sf83 product text)
		X'1100'	OPERATING SYSTEM: Software that controls the execution of programs. An operating system may provide services such as resource allocation, scheduling, I/O control , and data management.
		X'1200'	SOFTWARE PROGRAM: A program implemented in software, as distinguished from one implemented in microcode.
			X'1201' REQUIRED NODES ARE NOT ACTIVE
			X'1203' COMMUNICATION PROTOCOL PROGRAM SENT AN INVALID COMMAND
			X'1204' ERROR CALLING INDUSTRIAL COMPUTER LANGUAGE EXTENSIONS LIBRARY
			X'1205' MANUFACTURING AUTOMATION PROTOCOL COMMUNICATIONS PROTOCOL
			X'1206' TCP/IP COMMUNICATIONS PROTOCOL
			X'1207' PROTOCOL VIOLATION DURING RESYNCHRONIZATION PROCESSING
			X'1208' PROTOCOL VIOLATION DURING SYNCHRONIZATION POINT PROCESSING
			X'1209' INVALID PATH SPECIFIED
			X'120A' INVALID PATH SPECIFIED IN CONFIGURATION FILE
			X'120C' INVALID PARAMETER IN REQUEST DATA STRUCTURE
			X'120D' INVALID PARAMETER IN REQUEST WORK AREA
			X'120F' LOGICAL DEVICE INTERFACE PROCEDURES (LDIP) PROGRAM SENT AN INVALID COMMAND
			X'1210' INVALID PRIORITY
			X'1211' INVALID PRIORITY IN REQUEST DATA STRUCTURE
			X'1212' INVALID QUEUE NUMBER IN REQUEST
			X'1213' INVALID REQUEST TYPE

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'1214'	INVALID TASK NUMBER
		X'1215'	INVALID TIMEOUT VALUE IN REQUEST DATA STRUCTURE
		X'1216'	INVALID TRANSACTION NUMBER
		X'1217'	MESSAGE SIZE EXCEEDS ALLOCATED BUFFER SIZE
		X'1218'	ERROR ATTEMPTING TO ACCESS MESSAGE FILE
		X'121A'	MANUFACTURING AUTOMATION PROTOCOL APPLICATION PROGRAM INTERFACE ALLOCATION TABLE FULL
		X'121B'	TABLE MEMBER LOCKED BY ANOTHER USER
		X'121C'	TABLE RESOURCE EMPTY
		X'121D'	INVALID NODE INFORMATION BLOCK REQUEST
		X'121E'	MESSAGE TOO LONG FOR COMMUNICATION PROTOCOL PROGRAM TO TRANSMIT
		X'1220'	DATABASE MONITOR TIME-OUT DURING START-UP OR SHUT-DOWN OF DATABASE RESOURCES
		X'1221'	NEW SUBNET CONTROL NODE NOT FOUND IN CONFIGURATION FILE
		X'1222'	NEW SYSTEM CONTROL NODE NOT FOUND IN CONFIGURATION FILE
		X'1223'	ORIGINATOR RESOURCE PURGED
		X'1224'	NOT ENOUGH MEMORY STORAGE IN RESOURCE FOR NEW MEMBER
		X'1225'	PRINT SPOOLER CONTROL FILE ERROR
		X'1226'	PRINT SPOOLER CONTROL FILE FULL
		X'1227'	PROGRAM INITIALIZATION FAILED
		X'1228'	RESOURCE INITIALIZATION FAILED
		X'122A'	REQUESTED DEVICE CONTROL BLOCK FAILED TO START SUCCESSFULLY
		X'122B'	REQUESTED LINE CONTROL BLOCK FAILED TO START SUCCESSFULLY
		X'122C'	REQUESTED VIDEO CONTROL BLOCK FAILED TO START SUCCESSFULLY
		X'122E'	RESOURCE NOT SHUT DOWN
		X'122F'	RESOURCE NOT STARTED
		X'1230'	SERVER RESOURCE INVALID
		X'1236'	TIME-OUT EXPIRED WHILE WAITING FOR RESPONSE FROM DESTINATION RESOURCE
		X'1237'	TIMED OUT WAITING FOR COMMUNICATION PROTOCOL PROGRAM EVENT
		X'1239'	TIMED OUT WAITING FOR HARDWARE I/O PORT ACCESS
		X'123C'	SYSTEM TEMPLATE FILE ERROR
		X'123D'	TABLE RESOURCE LOCKED BY ANOTHER USER
		X'123E'	TABLE RESOURCE NOT AVAILABLE

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'1240'	TASK NUMBER NOT AVAILABLE
		X'1241'	TIME-OUT EXPIRED WHILE WAITING FOR COMPLETION OF PRINT REQUEST
		X'1242'	TIME-OUT EXPIRED WHILE WAITING FOR RESPONSE FROM REMOTE NODE
		X'1243'	TIME-OUT EXPIRED WHILE WAITING FOR COMPLETION OF SPOOL REQUEST
		X'1244'	TIME-OUT EXPIRED WHILE WAITING FOR SUBNET CONFIGURATION PACKAGE
		X'1245'	VM ACCESS COMMAND FAILED
		X'1246'	VM LINK COMMAND FAILED
		X'1247'	VM RELEASE COMMAND FAILED
		X'1248'	DATABASE SESSION NOT DISCONNECTED
		X'1249'	DATABASE SESSION TERMINATED – REQUEST REJECTED
		X'124A'	INVALID PARAMETER
		X'12A0'	ERROR OPENING AUXILIARY DISK (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'12A1'	ERROR OPENING PRINCIPAL DISK (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'12A2'	ERROR OPENING DISTRIBUTION GROUP TABLE (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'12A3'	ERROR SENDING ENVIRONMENT STATUS MESSAGE TO (detailed data qualifier) *Note:* The qualifier identifies the node.
		X'12A4'	ERROR SENDING MESSAGE TO APPLICATION (detailed data qualifier) *Note:* The qualifier identifies the application.
		X'12A5'	ERROR SENDING MESSAGE TO CONTROL NODE (detailed data qualifier) *Note:* The qualifier identifies the node.
		X'12A6'	DISK ERROR WRITING (detailed data qualifier) *Note:* The qualifier identifies the file.
		X'12A7'	TABLES AND QUEUES DEFINITION FILE ERROR (detailed data qualifier) *Note:* The qualifier identifies the operating system return code. The file containing descriptive information about software tables and various system queues contains an error.
		X'12A8'	ERROR DECODING MESSAGE (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'12A9'	ERROR ATTEMPTING TO DELETE (detailed data qualifier) *Note:* The qualifier identifies the file.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'12AA'	ERROR IN SECONDARY INDEX FILE (detailed data qualifier)
		X'12AB'	ERROR CLOSING (detailed data qualifier) *Note:* The qualifier identifies the file.
		X'12AC'	ERROR OPENING (detailed data qualifier) *Note:* The qualifier identifies the file.
		X'12AD'	(detailed data qualifier) RESTORED AND NOT SYNCHRONIZED *Note:* The qualifier identifies the file.
		X'12AF'	INVALID/UNSUPPORTED CODE PAGE SPECIFIED IN (detailed data qualifier) *Note:* The qualifier identifies the file.
		X'12B0'	INVALID/UNSUPPORTED COUNTRY CODE SPECIFIED IN (detailed data qualifier) *Note:* The qualifier identifies the file.
		X'12B1'	ERROR SENDING TO PRINT SERVER (detailed data qualifier) *Note:* The qualifier identifies the resource name.
		X'12B3'	INVALID PATH/MINIDISK (detailed data qualifier) SPECIFICATION *Note:* The qualifier identifies either the path or minidisk.
		X'12B4'	DISTRIBUTION SNAPSHOT FILE ERROR (detailed data qualifier) *Note:* The qualifier identifies the return code. A distribution file contains information that is to be distributed among certain nodes. A snapshot file, or the distribution file's status at some point in time, contains an error.
		X'12B5'	INDUSTRIAL COMPUTER EVENT (detailed data qualifier) OCCURRED *Note:* The qualifier identifies the event.
		X'12B6'	UNSOLICITED MESSAGE RECEIVED FROM (detailed data qualifier) *Note:* The qualifier identifies the device.
		X'12B7'	INITIALIZATION ERROR (detailed data qualifier) *Note:* The qualifier identifies the return code.
		X'12B8'	TIMEOUT OCCURRED WHILE WAITING FOR SESSION (detailed data qualifier) TO STOP *Note:* The qualifiers identify the session.
		X'12B9'	SUBSERVIENT NODE (detailed data qualifier) FAILED TO RESPOND *Note:* The qualifier identifies the node.
		X'12BA'	INVALID PARAMETER (detailed data qualifier) *Note:* The qualifier identifies the parameter.
		X'12BB'	MEMORY ALLOCATION ERROR (detailed data qualifier) *Note:* The qualifier identifies the return code.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
			X'12BC' NETBIOS COMMUNICATION ERROR — ORIGINATING PROGRAM (detailed data qualifier) *Note:* The qualifier identifies the program.
			X'12BD' NO MEMORY AVAILABLE TO RECEIVE MESSAGE FOR (detailed data qualifier) *Note:* The qualifier identifies the device.
			X'12BE' NODE (detailed data qualifier) FAILED TO HEARTBEAT *Note:* The qualifier identifies the node.
			X'12BF' UNEXPECTED MESSAGE (detailed data qualifier) *Note:* The qualifier identifies the command code.
			X'12C0' ERROR ON VARIABLE POLLER (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the variable poller process and the return code, respectively.
			X'12C1' MANUFACTURING AUTOMATION PROTOCOL ERROR (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the function and the return code, respectively.
			X'12CA' ERROR WRITING TO (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the file and the operating system return code, respectively.
			X'12CB' ERROR DETECTED ON CONNECTION TO (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the connection name and the return code, respectively.
			X'12D1' ERROR ON JOURNAL (detailed data qualifier) CONNECTION NAME (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the journal resource name, the connection name, and the return code, respectively. A journal (as used here) is used to record information about device events. The connection is the connection between a logical device and an application.
		X'2000'	COMMUNICATIONS ERROR: An error has occurred on a communication facility
			X'2001' START-STOP COMMUNICATIONS ERROR
			X'2002' BINARY SYNCHRONOUS COMMUNICATIONS ERROR
			X'2003' SNA COMMUNICATIONS ERROR
			X'2004' SDLC COMMUNICATIONS ERROR
			X'2005' X.21 NETWORK
			X'2006' X.25 COMMUNICATIONS ERROR
			X'2007' LAN COMMUNICATIONS ERROR
			X'2008' BANKING LOOP ERROR
			X'2009' STORE LOOP ERROR

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'200A'	ISDN COMMUNICATIONS ERROR
		X'200B'	LINK
		X'200D'	FRAME RELAY COMMUNICATIONS ERROR
		X'200E'	LOCAL DCE LOOP: the DCE loop local to the error notification sender. *Note:* A DCE loop is the equipment comprised of cables, converters, and the like that connect the DCE with the nearest central office exchange; this equipment does not include the customer premises wiring.
		X'200F'	REMOTE DCE LOOP: The DCE loop remote from the error notification sender.
		X'2010'	DDS NETWORK: A network implementing the Digital Data Services, e.g., the DATAPHONE[1] Digital Service (DDS). [1] DATAPHONE is the Registered Service Mark of AT&T Company.
		X'2011'	SWITCHED NETWORK: This could be (but is not limited to) a public switched telephone network
		X'2012'	SERIAL NETWORK
		X'2013'	DS1 NETWORK
		X'2014'	DS1 COMMUNICATIONS ERROR
		X'2040'	INTER-EXCHANGE NETWORK: A network providing services between two local exchange areas
		X'2041'	PRIVATE NETWORK REACHED: The private network containing the called DTE
		X'2049'	DS1 NETWORK HAS INITIATED A TEST LOOP
		X'2050'	X.21 NETWORK HAS INITIATED A TEST LOOP *Note:* The different test loops defined for X.21 networks are documented in the CCITT X.150 Recommendation.
		X'2051'	ISDN NETWORK HAS INITIATED A TEST LOOP
		X'2052'	X.25 NETWORK HAS INITIATED A TEST LOOP
		X'2053'	COMMUNICATIONS PROGRAM ABNORMALLY TERMINATED: Abend of a DLC or path control component.
		X'2054'	INSUFFICIENT STORAGE FOR INTERMEDIATE SESSION SETUP: Session Connector is unable to allocate storage for processing.
		X'2055'	LOGICAL LINK CONTROL
		X'2056'	B-CHANNEL ISDN ERROR
		X'2057'	D-CHANNEL ISDN ERROR
		X'2058'	SERIAL LINK
		X'2060'	REQUEST FOR SESSION CONNECT REJECTED
		X'2061'	REQUEST TIMED OUT
		X'20A0'	NO RESPONSE FROM THE X.21 NETWORK — (detailed data qualifier) EXPIRED *Note:* The qualifier specifies the X.21 timer that has expired.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'20A1'	NO RESPONSE FROM THE ISDN NETWORK — (detailed data qualifier) EXPIRED *Note:* The qualifier specifies the ISDN timer that has expired.
		X'20A2'	OSI PROTOCOL ERROR (detailed data qualifier) *Note:* The qualifier specifies the protocol code that defines the error condition which has occurred.
		X'20A3'	SNA COMMUNICATIONS ERROR (detailed data qualifier)
		X'20A4'	NO RESPONSE FROM THE X.25 NETWORK — (detailed data qualifier) EXPIRED *Note:* The qualifier specifies the X.25 timer that has expired.
		X'20A5'	NO RESPONSE FROM THE X.25 NETWORK — (detailed data qualifier) RETRY COUNT EXPIRED *Note:* The qualifier specifies the X.25 timer for which the retry count has expired.
		X'20A6'	(detailed data qualifier) LINE: The telephone line or transmission link connecting two or more components in the network *Note:* The qualifier identifies the link segment level (LSL) on which the line belongs.
		X'20A7'	(detailed data qualifier) OUTBOUND LINE: The equipment that connects the transmit circuits of the local DCE (i.e., the DCE local to the error notification sending node) to the receive circuits of the remote DCE. *Note:* The qualifier identifies the link segment level (LSL) on which the inbound line belongs.
		X'20A8'	(detailed data qualifier) INBOUND LINE: The equipment that connects the receive circuits of the local DCE (i.e., the DCE local to the error notification sending node) to the transmit circuits of the remote DCE. *Note:* The qualifier identifies the link segment level (LSL) on which the outbound line belongs.
		X'20A9'	COMMUNICATIONS ERROR DURING IPL WITH AUTOMATED REPLY FUNCTION ACTIVE — (detailed data qualifier) *Note:* The qualifier identifies the failing resource.
		X'20B1'	X.25 COMMUNICATIONS ERROR — THE FOLLOWING DIAGNOSTIC PACKET WAS RECEIVED FROM THE NETWORK (detailed data qualifier) *Note:* The qualifier specifies the diagnostic code.
		X'20B2'	X.25 PROTOCOL VIOLATION DETECTED (detailed data qualifier) *Note:* The qualifier specifies the diagnostic code.
		X'20B3'	NO RESPONSE RECEIVED — (detailed data qualifier) EXPIRED *Note:* The qualifier specifies the timer that has expired.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'20C0'	COMMUNICATIONS ERROR OCCURRED WITH UNLIMITED RETRIES SPECIFIED – (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the failing resource and corresponding configuration parameter.
		X'20C1'	X.25 COMMUNICATIONS ERROR – THE FOLLOWING INDICATION PACKET WAS RECEIVED FROM THE NETWORK (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and the cause code. The second qualifier specifies the diagnostic code. This code point is sent when an error is detected after end-to-end LLC communication has been established. Contrast with code point X'23C1'.
		X'20C2'	X.25 COMMUNICATIONS ERROR – THE DTE SENT THE FOLLOWING REQUEST PACKET TO THE NETWORK (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and the cause code. The second qualifier specifies the diagnostic code. This code point is sent when an error is detected after end-to-end LLC communication has been established. Contrast with code point X'23C2'.
		X'20C3'	X.25 COMMUNICATIONS ERROR – THE FOLLOWING DIAGNOSTIC PACKET WAS RECEIVED FROM THE NETWORK (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the diagnostic code and the second qualifier specifies the diagnostic explanation.
		X'20C4'	X.25 COMMUNICATIONS ERROR – THE FOLLOWING INDICATION PACKET WAS SENT BY THE NETWORK (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and cause code and the second qualifier specifies the diagnostic code.
		X'20C5'	X.25 COMMUNICATIONS ERROR – THE NETWORK RECEIVED THE FOLLOWING REQUEST PACKET FROM THE DTE (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and cause code and the second qualifier specifies the diagnostic code.
		X'20C6'	(detailed data qualifier) – (detailed data qualifier) CONNECTION NOT AVAILABLE AT THIS TIME *Note:* The qualifiers indicate the node names at either end of the connection.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'20C7'	APPC COMMUNICATIONS ERROR (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers indicate the primary and secondary return codes, respectively.
		X'20C8'	X.25 COMMUNICATIONS ERROR — THE FOLLOWING INDICATION PACKET WAS RECEIVED (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and the cause code. The second qualifier specifies the diagnostic code.
		X'20C9'	X.25 COMMUNICATIONS ERROR — THE FOLLOWING REQUEST PACKET WAS SENT (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and the cause code. The second qualifier specifies the diagnostic code.
		X'20D1'	NO RESPONSE FROM THE X.25 NETWORK — (detailed data qualifier) EXPIRED (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the timer. The second qualifier specifies the retry count and the third qualifier specifies the timer setting.
		X'20D2'	(detailed data qualifier) (detailed data qualifier) WAS ABOVE (detailed data qualifier) *Note:* The first qualifier identifies the configuration parameter and the second qualifier specifies the parameter value. The third qualifier specifies the threshold parameter.
		X'20D3'	(detailed data qualifier) (detailed data qualifier) WAS BELOW (detailed data qualifier) *Note:* The first qualifier identifies the configuration parameter and the second qualifier specifies the parameter value. The third qualifier specifies the threshold parameter.
	X'2100'		COMMUNICATIONS/REMOTE NODE: Either a communication facility denoted by a X'20xx' code point or a remote node denoted by a X'22xx' code point *Note:* This code point is used only when a more specific probable cause cannot be determined.
		X'2101'	START-STOP COMMUNICATIONS/REMOTE NODE
		X'2102'	BINARY SYNCHRONOUS COMMUNICATIONS/REMOTE NODE
		X'2104'	SDLC COMMUNICATIONS/REMOTE NODE
		X'2107'	LAN LLC COMMUNICATIONS/REMOTE NODE
		X'210A'	ISDN COMMUNICATIONS/REMOTE NODE
	X'2200'		REMOTE NODE: The node at the remote end of a link connection *Note:* "Remote" is defined from the point of view of the node detecting the Alert condition.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'2201'	CALLED DTE
			X'2202'	CALLED DTE SIGNALLING CONTROLLED NOT READY: The called DTE has indicated that it is temporarily unable to accept incoming calls for circuit-switched service *Note:* This condition is unique to X.21.
			X'2203'	CALLED DTE SIGNALLING UNCONTROLLED NOT READY: The called DTE has indicated that it is unable to enter an operational state for accepting an incoming call *Note:* This condition is unique to X.21.
			X'2204'	OTHER REMOTE NODE: On a multidrop link, the remote node interfering with the link activity but not part of the logical connection for which the error was detected.
			X'2205'	TOPOLOGY PROTOCOL ERROR: Format error in a topology database update GDS variable.
			X'2206'	DIRECTORY PROGRAM REMOTE NODE: Protocol violation on a LOCATE request.
			X'2207'	UNAUTHORIZED NETWORK CHANGE ATTEMPTED: An end-node CP without authorization has attempted to delete a resource.
			X'2208'	COMMUNICATIONS PROGRAM IN ADJACENT NODE: Session Connector received an invalid RU.
			X'2209'	SESSION SERVICES PROGRAM IN REMOTE NODE
			X'22A0'	REMOTE NODE (detailed data qualifier)
		X'2300'	CONNECTION NOT ESTABLISHED: A telephone connection required for the requested operation has not been established	
			X'2306'	NEW TELEPHONE NUMBER ASSIGNED TO CALLED DTE
			X'2307'	CALLED NUMBER OUTSIDE OF NUMBERING PLAN OR UNKNOWN BY THE NETWORK
			X'2308'	ACCESS BARRED: The calling DTE is not allowed to connect to the called DTE
			X'2309'	SPEED CLASSES INCOMPATIBLE: The called DTE is operating at a different speed from the calling DTE
			X'230A'	USER CLASSES OF SERVICE INCOMPATIBLE: The user class of service of the called DTE is incompatible with that of the calling DTE
			X'230B'	TERMINAL EQUIPMENT IDENTIFIER (TEI) ASSIGNMENT
			X'230C'	CALL REJECTED BY CALLED DTE
			X'2310'	CALLED DTE NOT RESPONDING
			X'2311'	SERVICE NOT AVAILABLE OR NOT SUPPORTED
			X'2312'	CALL REFERENCE NUMBER ERROR
			X'2314'	ISDN TO NON-ISDN OPERATION ERROR: The interconnection of services between ISDN and non-ISDN networks has failed or is not available.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'2315'	CALL COLLISION: An outgoing call was not completed because it collided with an incoming call on the same link
		X'2316'	LINE OR LINES NOT IN VALID STATE FOR CALL: The line (or all possible lines) is not in a valid state to place or accept a call, i.e., the line is in an error state, unavailable because it is is use, or is not varied on.
		X'2317'	NETWORK INTERFACE NOT IN VALID STATE FOR CALL: The network interface is not in a valid state to be used to place or accept a call, i.e., it is in an error state or is not varied on.
		X'2318'	INCOMING CALL REFUSED – PATH UNAVAILABLE
		X'2319'	OUTGOING CALL REFUSED – LLC TYPE INVALID
		X'23C1'	X.25 COMMUNICATIONS NOT ESTABLISHED – THE FOLLOWING INDICATION PACKET WAS RECEIVED FROM THE NETWORK (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and the cause code. The second qualifier specifies the diagnostic code. This code point is sent when an error is detected before end-to-end LLC communication has been established. Contrast with code point X'20C1'.
		X'23C2'	X.25 COMMUNICATIONS NOT ESTABLISHED – THE DTE SENT THE FOLLOWING REQUEST PACKET TO THE NETWORK (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and the cause code. The second qualifier specifies the diagnostic code. This code point is sent when an error is detected before end-to-end LLC communication has been established. Contrast with code point X'20C2'.
		X'2600'	INTERFERENCE: An electric disturbance in a communication system that interferes with or prevents reception of a signal or of information
		X'3000'	CHANNEL FAILURE: The equipment that is used to direct data to and from input/output devices and locally attached control units has experienced a failure
		X'3100'	CONTROLLER FAILURE: A communication device that controls other devices and the flow of information to and from them has experienced a failure

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'3110'	COMMUNICATION CONTROLLER BACK-UP: A process which switches resources from one processor to a back-up processor in a communication controller. *Note:* This code point is used to notify the network operator about a maintenance procedure that was invoked locally or initiated automatically which results in the availability of additional resources.
			X'3111'	COMMUNICATION CONTROLLER: A communication device that controls the transmission of data over lines in a network *Note:* In SNA a communication controller is a type 4 node.
			X'3121'	TERMINAL CONTROL UNIT: A communication device that controls the transmission of data to and from terminals
			X'3131'	DASD CONTROL UNIT: A device that controls the transfer of data to and from a direct access storage device such as a disk or drum
			X'3133'	STORAGE DEVICE CONTROLLER
		X'3200'		COMMUNICATIONS INTERFACE: The equipment connecting a node to the component in a link connection with which it exchanges physical control signals
			X'3220'	LOCAL TOKEN-RING ADAPTER INTERFACE: The programming interface for the local token-ring adapter
			X'3221'	CSMA/CD ADAPTER INTERFACE: The programming interface for the local CSMA/CD adapter
			X'3225'	ISDN-R INTERFACE
			X'32D1'	LOCAL DCE COMMUNICATIONS INTERFACE (detailed data qualifier) (detailed data qualifier) (detailed data qualifier): The communication interface between the Alert sender and the local Data Circuit-Terminating Equipment (DCE) *Note:* The qualifiers identify the standards, protocols, and other characteristics that characterize the interface, e.g., RS-232C, 1200 BPS, V.24.
			X'32D2'	REMOTE DCE COMMUNICATIONS INTERFACE (detailed data qualifier) (detailed data qualifier) (detailed data qualifier): The communication interface between the Data Circuit-Terminating Equipment (DCE) remote from the Alert sender and the remote node *Note:* The qualifiers identify the standards, protocols, and other characteristics that characterize the interface, e.g., RS-232C, 1200 BPS, V.24.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'32D3'	REMOTE DCE COMMUNICATIONS INTERFACE (detailed data qualifier) (detailed data qualifier) (detailed data qualifier): The communication interface between the Alert sender and the DCE emulation cable that attaches it to a device's DCE interface cable *Note:* The qualifiers identify the standards, protocols, etc. that characterize the interface, e.g., RS-232C, 1200 BPS, V.24.
		X'3300'	ADAPTER: The part of a device that interfaces between a processor in the device and one or more attached devices *Note:* The processor referred to here could be either the main processor in the node containing the adapter or a processor in, e.g., a communication subsystem controller.
		X'3301'	CHANNEL ADAPTER
		X'3302'	COMMUNICATIONS ADAPTER
		X'3303'	DASD ADAPTER
		X'3304'	DISPLAY/PRINTER ADAPTER
		X'3305'	DIRECT-ATTACHED ADAPTER
		X'3306'	DISKETTE ADAPTER
		X'3307'	ENCRYPTION/DECRYPTION ADAPTER
		X'3309'	LINE ADAPTER
		X'330A'	LOOP ADAPTER
		X'330B'	PARALLEL INTERFACE ADAPTER
		X'330C'	SERIAL INTERFACE ADAPTER
		X'330D'	TAPE ADAPTER
		X'330E'	CONSOLE ADAPTER
		X'330F'	HPTSS ADAPTER: A high-speed processor transmission subsystem adapter in a communication controller
		X'3310'	LOCAL ISDN ADAPTER: An adapter that attaches the Alert sender to an ISDN network *Note:* See also code point X'3532' LOCAL ISDN TERMINAL ADAPTER. A terminal adapter is distinguished from an ISDN adapter by the presence of a defined interface (e.g., RS-232C) between itself and the node that it serves; an ISDN adapter is typically integrated within its node.
		X'3311'	REMOTE ISDN ADAPTER: An adapter that attaches to an ISDN network a node with which the Alert sender has a logical connection utilizing the network *Note:* See also code point X'3533' REMOTE ISDN TERMINAL ADAPTER.
		X'3312'	LOCAL DS1 ADAPTER
		X'3320'	LOCAL TOKEN-RING ADAPTER: An adapter that attaches the Alert sender to a token-ring LAN

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'3321'	REMOTE TOKEN-RING ADAPTER: An adapter that attaches a node other than the Alert sender to a token-ring LAN
		X'3322'	LOCAL CSMA/CD ADAPTER: An adapter that attaches the Alert sender to a CSMA/CD LAN
		X'3323'	REMOTE CSMA/CD ADAPTER: An adapter that attaches a node other than the Alert sender to a CSMA/CD LAN
		X'3324'	TOKEN-RING ADAPTER: An adapter that attaches a node to a token-ring LAN
		X'3325'	CSMA/CD ADAPTER: An adapter that attaches a node to a CSMA/CD LAN
		X'3326'	TOKEN BUS ADAPTER
		X'3327'	DEFECTIVE TRANSMITTER
		X'3328'	3270 ADAPTER
		X'3329'	LOCAL FRAME RELAY ADAPTER: An adapter that attaches the Alert sender to a Frame Relay network
		X'3330'	ADAPTER HARDWARE
		X'3331'	ADAPTER MICROCODE
		X'3340'	LOCAL LAN ADAPTER
		X'3341'	STORAGE DEVICE ADAPTER
		X'334F'	REMOTE FDDI ADAPTER: An adapter that attaches a node other than the Alert sender to a FDDI LAN
		X'3351'	FDDI ADAPTER
		X'336F'	CONTROLLER BUS ADAPTER
		X'3370'	OPTICAL SYSTEM BUS ADAPTER
		X'33A0'	LINE ADAPTER MULTIPLEXER (detailed data qualifier): A line adapter (scanner) multiplexer in a communication controller *Note:* The qualifier identifies the line address for the failing multiplexer.
		X'33A5'	LAN ADAPTER — (detailed data qualifier) *Note:* The qualifier identifies the type.
		X'33C0'	LINE ADAPTER (detailed data qualifier) (detailed data qualifier): A line adapter (scanner) in a communication controller *Note:* The qualifiers identify the line adapter number and the line address range for the failing adapter.
		X'33C1'	LINE ADAPTER HARDWARE (detailed data qualifier) (detailed data qualifier): The hardware comprising a line adapter (scanner) in a communication controller *Note:* The qualifiers identify the line adapter number and the line address range for the failing adapter.
		X'33C2'	LINE ADAPTER MICROCODE (detailed data qualifier) (detailed data qualifier): The microcode executing in a line adapter (scanner) in a communication controller *Note:* The qualifiers identify the line adapter number and the line address range for the failing adapter.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'33C3'	LINE INTERFACE COUPLER (LIC) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the line address and the LIC position for the failing LIC.
		X'3400'	CABLE LOOSE OR DEFECTIVE: A cable or its connectors used to electrically connect devices together is loose or defective
		X'3401'	LOCAL DCE INTERFACE CABLE: The cable, or its connectors, between the Alert sender and the local Data Circuit-Terminating Equipment (DCE)
		X'3403'	REMOTE DCE INTERFACE CABLE: The cable, or its connectors, between the Alert sender's remote DCE and the device attached to it.
		X'3404'	DCE EMULATION CABLE: The cable, or its connectors, between the Alert sender and a DCE interface cable attached to a device *Note:* The end of the DCE emulation cable remote from the Alert sender plugs directly into the DCE interface cable attached to the device.
		X'3411'	CHANNEL INTERFACE CABLE: The cable or cables, or their connectors, between a channel and a locally attached device
		X'3426'	CSMA/CD LAN CABLES: The cables in a CSMA/CD LAN. These include the cable attaching the alert sender to the CSMA/CD bus and the bus itself
		X'3430'	FDDI CABLE
		X'3434'	LOCAL LOBE CABLES: The cables between the reporting node and its access unit on a token-ring LAN
		X'3435'	REMOTE LOBE CABLES: The cables between a remote node and its access unit on a token-ring LAN
		X'3436'	LOCAL CSMA/CD ADAPTER CABLE: The cable attaching the Alert sender to the CSMA/CD bus
		X'3441'	LOOP CABLE: A cable connecting the nodes attached to a communication loop
		X'3451'	DEVICE CABLE: A cable connecting a device directly to a communication controller or a control unit
		X'3452'	STORAGE DEVICE CABLE: A cable directly connecting a local storage device to its adapter/controller
		X'3460'	INTERNAL CABLE
		X'3461'	LINE ADAPTER MULTIPLEXER CABLE
		X'3465'	PREMISES WIRING
		X'3470'	OPTICAL SYSTEM BUS CABLE
		X'3480'	TWINAXIAL CABLE DISTRIBUTION ASSEMBLY
		X'34A0'	(detailed data qualifier) LOCAL DCE INTERFACE CABLE: On a particular link segment, the DCE interface cable nearer to the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the DCE belongs.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'34A1'	(detailed data qualifier) REMOTE DCE INTERFACE CABLE On a particular link segment, the DCE interface cable farther from the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the DCE belongs.
		X'3500'	COMMUNICATION EQUIPMENT: External equipment used to connect devices or other system components *Note:* If the attaching equipment is known to be a modem, then a modem code point (X'36xx') is sent instead of this code point.
		X'3501'	PROTOCOL CONVERTER: A device that converts one protocol data stream to another. This code point applies to a protocol converter providing conversion between any two data streams regardless of whether attached via a communications link and/or a local attachment link such as 327X coaxial link or 525X twinaxial link. Protocols involved can include SDLC, BISYNC, ASYNC, 327X and 525X. *Note:* This code point is not to be used for a X.25 Packet Assembler-Disassembler (X.25 Pad).
		X'3502'	TERMINAL MULTIPLEXER: The equipment used to connect multiple devices to a single cable
		X'3503'	LINE SWITCH: A device that on demand allows Data Circuit-Terminating Equipment (DCE) to be attached to different Data Terminal Equipment (DTE) ports. The device supports both digital switching for the DCE-DTE interface and also the switching of the analog interface between the DCE and the communication facility (line).
		X'3504'	AUTO-CALL UNIT: A stand-alone or integrated unit used to establish connection on a switched communication line and connected in parallel with the modem used for data transmission but connected to the DTE via a separate interface (i.e., EIA-366/CCITT V.25).
		X'3506'	LOCAL DIGITAL DATA DEVICE: On a particular link segment, the digital data device (DDD) nearer to the Alert sender *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with Failure Cause X'F0A3', FAILURE OCCURRED ON (detailed data qualifier).
		X'3507'	REMOTE DIGITAL DATA DEVICE: On a particular link segment, the digital data device (DDD) farther from the Alert sender *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with Failure Cause X'F0A3', FAILURE OCCURRED ON (detailed data qualifier).

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'3509'	SERIAL NODE
		X'3510'	CALLED DCE
		X'3511'	LINE: The telephone line or transmission link connecting two or more components in the network *Note:* For a multi-segment link connection, this text does not indicate which segment is involved. This information is typically communicated by means of a qualifier associated with Failure Cause X'F0A3', FAILURE OCCURRED ON (detailed data qualifier).
		X'3512'	THE CONNECTION BETWEEN THE CALLING DCE AND ITS LOCAL DSE
		X'3513'	LOCAL LOOP ASSOCIATED WITH THE CALLED DTE
		X'351F'	DS1 NETWORK COMPONENT
		X'3520'	X.21 NETWORK COMPONENT
		X'3521'	TEMPORARY LACK OF RESOURCES IN THE X.21 NETWORK
		X'3522'	LONG-TERM LACK OF RESOURCES IN THE X.21 NETWORK
		X'352F'	ISDN TE-NT CONNECTION
		X'3530'	ISDN NETWORK COMPONENT
		X'3531'	ISDN NETWORK TERMINATION EQUIPMENT (NT1): A device, normally residing on the user's premises, that provides conversion, for basic-rate ISDN service, between the 4-wire interface seen by the user and the 2-wire interface seen by the ISDN service provider
		X'3532'	LOCAL ISDN TERMINAL ADAPTER: The terminal adapter local to the Alert sender *Note:* See also code point X'3310' LOCAL ISDN ADAPTER. A terminal adapter is distinguished from an ISDN adapter by the presence of a defined interface (e.g., RS-232C) between itself and the node that it serves; an ISDN adapter is typically integrated within its node.
		X'3533'	REMOTE ISDN TERMINAL ADAPTER: The terminal adapter that attaches to an ISDN network a node with which the Alert sender has a logical connection utilizing the network *Note:* See also code point X'3311' REMOTE ISDN ADAPTER.
		X'3534'	TEMPORARY LACK OF RESOURCES IN THE ISDN NETWORK
		X'3535'	LONG-TERM LACK OF RESOURCES IN THE ISDN NETWORK
		X'3536'	ISDN NETWORK TERMINATION DEVICE

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'3541'	LOCAL DCE: The Data Circuit-Terminating Equipment (DCE) nearer to the error notification sender *Note:* This code point is used only if the Alert sender is unable to determine whether the DCE is a modem or a DDD; see code points X'3506' and X'3601'.
			X'3542'	REMOTE DCE: The Data Circuit-Terminating Equipment (DCE) farther from the error notification sender *Note:* This code point is used only if the Alert sender is unable to determine whether the DCE is a modem or a DDD; see code points X'3507' and X'3603'.
			X'3550'	X.25 NETWORK COMPONENT
			X'3551'	SHORT-TERM CONGESTION IN THE X.25 NETWORK
			X'3552'	LONG-TERM CONGESTION IN THE X.25 NETWORK
			X'35A0'	(detailed data qualifier) LOCAL DSU/CSU: On a particular link segment, the DSU/CSU nearer to the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
			X'35A1'	(detailed data qualifier) REMOTE DSU/CSU: On a particular link segment, the DSU/CSU farther from the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
			X'35A2'	(detailed data qualifier) LOCAL DCE: On a particular link segment, the DCE nearer to the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the DCE belongs.
			X'35A3'	(detailed data qualifier) REMOTE DCE: On a particular link segment, the DCE farther from the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the DCE belongs.
		X'3600'	MODEM: A device or functional unit that modulates and demodulates signals transmitted over data communication facilities	
			X'3601'	LOCAL MODEM: The modem connected to the error notification sender
			X'3602'	LOCAL LINK DIAGNOSTIC UNIT: A device that connects to both sides of a local modem and provides Link Problem Determination Aid (LPDA) data for digital and analog links with non-intelligent IBM or non-IBM modems
			X'3603'	REMOTE MODEM: The modem remote from the error notification sender

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
			X'3604' REMOTE LINK DIAGNOSTIC UNIT: A device that connects to both sides of a remote modem and provides Link Problem Determination Aid (LPDA) data for digital and analog links with non-intelligent IBM or non-IBM modems
			X'3621' LOCAL ENHANCED MODEM: The enhanced modem connected to the Alert sender *Note:* An enhanced modem can provide functions in addition to modulation/demodulation, such as establishing switched connections and storing dial digits.
			X'36A0' (detailed data qualifier) LOCAL MODEM: On a particular link segment, the modem nearer to the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the modem belongs.
			X'36A1' (detailed data qualifier) REMOTE MODEM: On a particular link segment, the modem farther from the error notification sender *Note:* The qualifier identifies the link segment level (LSL) on which the modem belongs.
			X'36A2' (detailed data qualifier) LOCAL MODEM FEATURE(S) *Note:* The qualifier identifies the link segment level (LSL) on which the modem belongs.
			X'36A3' (detailed data qualifier) REMOTE MODEM FEATURE(S) *Note:* The qualifier identifies the link segment level (LSL) on which the modem belongs.
		X'3700'	LAN COMPONENT: A component of a local area network. On a token-ring LAN, the LAN components include the adapters, bridges, access units, repeaters, repeater/amplifiers, and the LAN cable. On a CSMA/CD LAN, the LAN components include the adapters, bridges, LAN cables, taps, splitters, amplifiers, and translator units. On a FDDI LAN, the LAN components include the adapters, bridges, concentrators, and the LAN (FDDI) cable. *Note:* This default code point is used to indicate that some unspecified LAN component is a failure cause. Individual LAN components are denoted by replacement code points under X'3700', with the exception of the LAN cable, which falls under CABLE LOOSE OR DEFECTIVE (X'3400'), and the LAN adapters, which fall under ADAPTER (X'3300').
			X'3701' TOKEN-RING LAN COMPONENT
			X'3703' TOKEN-RING FAULT DOMAIN: An adapter, its nearest active upstream neighbor, and the token-ring media between them; the token-ring media consists of the lobe cables, portions of one or more access units, and possibly a portion of the LAN cable

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'3706'	OPTICAL FIBER CONVERTER: A device which converts electrical signals into optical signals and vice-versa
			X'3707'	TOKEN-RING LAN CABLES
			X'3708'	DOMAIN CONTROLLER
			X'370C'	INVALID SYMBOL RECEIVED FROM MAC: The physical layer transmits symbols presented to it by the medium access control (MAC) sublayer entity. This fault is issued when the physical layer cannot encode one of the MAC symbols as specified in the IEEE 802.4 standard.
			X'3711'	LOCAL ACCESS UNIT: The access unit by which the Alert sender is attached to a token-ring LAN *Note:* An access unit is an active or passive wiring concentrator on a token-ring LAN.
			X'3712'	LOCAL TOKEN-RING LOBE: A token-ring lobe attaching the Alert sender to a token-ring
			X'3713'	REMOTE ACCESS UNIT: An access unit by which a node other than the Alert sender is attached to a token-ring LAN
			X'3721'	CSMA/CD LAN COMPONENT
			X'3722'	CSMA/CD LAN TRANSLATOR UNIT: A component at the head end of a CSMA/CD bus, which accepts input at one frequency and transmits the same data at a different frequency
			X'3730'	FDDI LAN COMPONENT
			X'3733'	FDDI FAULT DOMAIN: An adapter, its nearest active upstream neighbor, and the FDDI media between them.
			X'3734'	FDDI LAN PORT
			X'3735'	FDDI LAN MAC
			X'3741'	CONGESTION IN LAN BRIDGE: Frames are arriving at a bridge faster than they can be forwarded by that bridge and, as a result, some frames are discarded
			X'3750'	TOKEN-RING CAU ATTACHMENT MODULE: An interface to the wire lobes connecting workstations to a Token Ring network.
			X'3752'	BASE UNIT INTERNAL ERROR: An error has occurred on the Controlled Access Unit component that directs the operation of the lobe attachment modules and interacts with the Token Ring network.
			X'3760'	FILE SERVER
		X'4000'	PERFORMANCE DEGRADED	
		X'40A0'	(detailed data qualifier) THRESHOLD REACHED *Note:* The qualifier identifies the counter.	
		X'5000'	MEDIA: A tape, disk, diskette, or paper (or other data medium) that is required to read data from or write data on	

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
			X'5001' DASD MEDIA: The medium used in a direct access storage device; it may be either removable or non-removable
			X'5002' DISKETTE: A thin, flexible magnetic disk in a semi-rigid protective jacket, in which the disk is permanently enclosed; also termed a floppy diskette
			X'5003' TAPE: A recording medium in the form of a ribbon that has one or more tracks along its length; magnetic recordings can be made on either one or both sides
		X'5200'	MEDIA JAM: The medium (usually paper, forms or cards) is jammed in the machine and operator action is required to correct the problem.
			X'5201' CARD JAM
			X'5202' FORMS JAM
			X'5203' PAPER JAM
			X'5204' FILM JAM
		X'6000'	DEVICE: An input, output, or input/output device (e.g., a terminal or disk drive)
		X'6100'	INPUT DEVICE: A device that is used to enter data into a system
			X'6110' KEYBOARD: An arrangement of alphanumeric, special character, and function keys laid out in a specified manner and used to enter information into a terminal, and thereby into a system
			X'6111' KEYPAD: A specialized keyboard with an arrangement of a limited number of alphanumeric, special character, and/or function keys
			X'6112' SELECTOR PEN: A light-sensitive pen used in display operations
			X'6113' MICR READER/SORTER: A magnetic ink character recognition reader/sorter
			X'6114' MAGNETIC STRIPE READER (MSR): A device that reads data from a card containing a magnetic stripe
			X'6115' ID CARD READER: An Identification Card Reader (ICR) is a device which can read data from or write data to a magnetic stripe or an electronic chip on a consumer's identification card.
		X'6200'	OUTPUT DEVICE: A device that receives data from a system
			X'6210' PRINTER: An output device that produces durable and optically viewable output in the form of characters (and optionally graphics) by a means other than by drawing with one or more pens *Note:* Contrast with code point X'6213' PLOTTER.
			X'6211' COPIER: An output device that produces one or more copies of an original without affecting the original

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'6212'	CAMERA: An output device that combines electronic data with a visual image on a single visual medium
			X'6213'	PLOTTER: An output device that produces graphic and/or character output by means of one or more pens that draw on the surface of the output medium *Note:* Contrast with code point X'6210' PRINTER.
			X'6220'	PRINTER CASSETTE: A removable container for feeding paper to a printer
		X'6300'	INPUT/OUTPUT DEVICE: A device whose parts can be performing an input process and output process at the same time, such as a card reader/punch	
			X'6301'	DISPLAY/PRINTER: A device that has either of the characteristics of a display or printer or both. This code point is used only when the Alert sender cannot determine whether the attached device is a display or printer
			X'6302'	DISPLAY: A workstation that requires a host connection in order to function; typically a display includes both a monitor and a keyboard
			X'6309'	STORAGE DEVICE: The device cannot be specifically identified as disk, tape, optical, etc.
			X'6310'	DISK DRIVE: The primarily mechanical component of a DASD device, directly involved with transferring data to and from the medium
			X'6311'	DISKETTE DRIVE: The primarily mechanical component of a diskette device, directly involved with transferring data to and from the medium
			X'6312'	OPTICAL DISK DEVICE: A direct access storage device that uses an optical disk as the storage medium. The disk may be either removable or non-removable
			X'6313'	TERMINAL: A device in a system or network at which data can either enter or leave. A terminal is usually equipped with a keyboard and a display device, and is capable of sending and receiving information
			X'6314'	TAPE DRIVE: The primarily mechanical component of a tape drive, directly involved with transferring data to and from the medium
			X'6315'	CONSOLE: A terminal used for communication between an operator and a processor
			X'6316'	TOUCH-SENSITIVE SCREEN
			X'6317'	MAGNETIC STRIPE READER/ENCODER: A device that reads data from, and in some cases writes data to, a card containing a magnetic stripe
			X'6318'	ENCRYPTION DEVICE
			X'6330'	DISK DRIVE ELECTRONICS: The electronic components of a DASD device
			X'6350'	LOCAL CONSOLE

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'6351'	REMOTE CONSOLE *Note:* "Local" and "remote" are defined with respect to the system with which the console communicates.
		X'63A0'	STORAGE DEVICE – (detailed data qualifier) *Note:* The qualifier identifies the type.
		X'6400'	DEPOSITORY: A device that receives items into a system
		X'6401'	ENVELOPE DEPOSITORY: A device that receives into a system items sealed in an envelope. The envelope is not opened, nor are its contents examined by the system; the envelope is stored for human action
		X'6402'	CHECK DEPOSITORY: A device that receives checks into a system, then reads and retains them. It may also transfer information to a check and return the check to a user
		X'6403'	CARD DEPOSITORY: A device that retains credit, personal banking, or other cards used to access a PBM
		X'6500'	DISPENSER: A device that dispenses items to a user of a system
		X'6501'	DOCUMENT DISPENSER: A device that dispenses documents, primarily bills
		X'6502'	TICKET DISPENSER
		X'6503'	KEY DISPENSER
		X'6504'	COIN DISPENSER
		X'6600'	SELF-SERVICE TERMINAL: A device that allows a customer of a business to perform a transaction that would otherwise require assistance by personnel of the business
		X'6601'	PERSONAL BANKING MACHINE: A self-service terminal for financial transactions
		X'6630'	TELLER ASSIST UNIT: A terminal that assists a financial teller in transactions *Note:* This device does not fit the strict definition of a self-service terminal, since it is used by personnel of a financial institution; it is included in this range because it is very close in function to other self-service terminals.
		X'6700'	SECURITY PROBLEM
		X'6701'	AUDIBLE ALARM: A device which emits an audible sound

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'6702'	PROTECTIVE DOOR: An electronically or mechanically operated covering for access to a device.
		X'E000' – X'EFFF'	Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range. *Note:* The following code points specify extended messages, that provide additional information on one or more failure causes that have already been specified. An Alert receiver that displays only default text provides no display for these code points.
		X'F000'	(no display): Additional message data
		X'F001'	UNSOLICITED INTERRUPT RECEIVED
		X'F002'	DATA LOST DURING RESTORE TO DISK
		X'F003'	IPL OCCURRED DUE TO A HARD WAIT
		X'F004'	NODE WILL SHUT DOWN
		X'F005'	NETWORK INTENTIONALLY SENDING CORRUPTED CRC TO THE NT1
		X'F006'	NETWORK REQUESTING THE NT1 TO INTENTIONALLY SEND CORRUPTED CRC
		X'F007'	IN-DOUBT LOGICAL UNIT OF WORK
		X'F008'	RESOURCE RECOVERY CANCELLED TO AVOID INFINITE RETRIES
		X'F009'	SOFTWARE PROBE EXECUTION FREQUENCY THRESHOLD REACHED – PROBE DISABLED
		X'F00A'	RETRY LIMIT REACHED
		X'F00C'	CRC/LRC RETRY LIMIT REACHED
		X'F00D'	IDLE DETECT TIMEOUT RETRY LIMIT REACHED
		X'F00E'	NON-PRODUCTIVE RECEIVE TIMEOUT RETRY LIMIT REACHED
		X'F00F'	RNR RECEIVED THRESHOLD REACHED
		X'F010'	FRAME REJECT RECEIVED: INVALID/UNSUPPORTED COMMAND OR RESPONSE SENT
		X'F011'	FRAME REJECT RECEIVED: I-FIELD SENT WHEN NOT PERMITTED
		X'F012'	FRAME REJECT RECEIVED: INVALID N(R) SENT
		X'F013'	FRAME REJECT RECEIVED: MAXIMUM I-FIELD LENGTH EXCEEDED
		X'F014'	FRAME REJECT RECEIVED: NO REASON SPECIFIED
		X'F015'	SNRM RECEIVED WHILE IN NRM
		X'F016'	SABME RECEIVED WHILE IN ABME
		X'F017'	POLL COUNT EXHAUSTED
		X'F018'	XID POLL COUNT EXHAUSTED
		X'F019'	INACTIVITY TIMER EXPIRED

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F01A'	DM RECEIVED
		X'F01B'	RECEIVE WINDOW SIZE EXCEEDED
		X'F01C'	LLC LEVEL CRC OR CHECKSUM ERROR THRESHOLD REACHED
		X'F01D'	LREJ RECEIVED THRESHOLD REACHED
		X'F01E'	LREJ SENT THRESHOLD REACHED
		X'F01F'	PASSWORD NOT FOUND
		X'F020'	INVALID/UNSUPPORTED COMMAND OR RESPONSE RECEIVED
		X'F021'	I-FIELD RECEIVED WHEN NOT PERMITTED
		X'F022'	INVALID N(R) RECEIVED
		X'F023'	RECEIVED I-FIELD EXCEEDED MAXIMUM LENGTH
		X'F024'	SEGMENTED DATA NOT EXPECTED
		X'F025'	U-FORMAT LPDU MISSING DATA WAS RECEIVED
		X'F026'	RECEIVED I-FIELD TOO SHORT
		X'F027'	REJECT RECEIVED: INVALID N(S) SENT
		X'F028'	DISC RECEIVED
		X'F029'	UNEXPECTED UA OR RR RECEIVED
		X'F02A'	RECEIVED XID OUT OF SEQUENCE
		X'F02B'	FRAME REJECT RECEIVED: S OR U FORMAT FRAME CONTAINING UNEXPECTED DATA RECEIVED
		X'F030'	CTS DROPPED
		X'F031'	CTS FAILED TO DROP
		X'F032'	DSR FAILED TO DROP
		X'F033'	RTS NOT RAISED BUT CTS IS ACTIVE
		X'F034'	CTS FAILED TO RISE
		X'F035'	DSR DROPPED
		X'F036'	DSR IS PRESENT BEFORE DTR IS RAISED
		X'F037'	DSR NOT PRESENT AFTER DTR IS RAISED
		X'F038'	CARRIER DETECT LOST
		X'F039'	DLO INITIALLY ON
		X'F03A'	DLO DID NOT COME ACTIVE DURING CALL REQUEST
		X'F03B'	PND FAILED TO COME ACTIVE IN THE REQUIRED TIME
		X'F03C'	DSC DID NOT COME ACTIVE AFTER A CALL REQUEST WAS COMPLETED
		X'F03D'	DSR DID NOT COME ACTIVE WHILE ATTEMPTING AN AUTO-CALL CONNECTION
		X'F040'	TRANSMISSION UNDERRUN THRESHOLD REACHED
		X'F041'	EXCESSIVE TRANSMIT PROCEDURE TIMEOUTS
		X'F042'	RECEIVE OVERRUN THRESHOLD REACHED
		X'F043'	EXCESSIVE RECEIVE PROCEDURE TIMEOUTS
		X'F044'	RECEIVE QUEUE OVERRUN

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'F050'	DCE NOT READY
			X'F051'	DCE CLEAR INDICATION DURING CALL ESTABLISHMENT
			X'F052'	PERSISTENT DCE CLEAR INDICATION DURING CALL ESTABLISHMENT (T6 TIMER EXPIRED)
			X'F053'	DCE CONTROLLED NOT READY DURING CALL ESTABLISHMENT
			X'F054'	PERSISTENT DCE CONTROLLED NOT READY DURING CALL ESTABLISHMENT (T6 TIMER EXPIRED)
			X'F055'	DCE FAULT CONDITION DURING CALL ESTABLISHMENT
			X'F056'	DCE CLEAR INDICATION RECEIVED DURING DATA PHASE
			X'F057'	PERSISTENT DCE CLEAR INDICATION RECEIVED DURING DATA PHASE (T6 TIMER EXPIRED)
			X'F058'	UNRECOGNIZED CALL PROGRESS SIGNAL RECEIVED FROM THE NETWORK
			X'F059'	INVALID CALL PROGRESS SIGNAL RECEIVED FROM THE NETWORK
			X'F05A'	DSR OR CTS DROPPED
			X'F05B'	FAN-OUT FEATURE IN ERROR
			X'F05C'	TOKEN RING BEGAN OR TERMINATED BEACONING
			X'F05D'	LAN WRAPPED TO CIRCUMVENT FAILING COMPONENT
			X'F060'	DATA BLOCKS IN ERROR THRESHOLD REACHED
			X'F061'	TTD'S TRANSMITTED THRESHOLD REACHED
			X'F062'	WACK'S TRANSMITTED THRESHOLD REACHED
			X'F063'	SYNC TIMEOUT THRESHOLD REACHED
			X'F064'	CONTINUOUS SYNC TIMEOUT RETRY LIMIT REACHED
			X'F065'	NO SYNC RECEIVED TIMEOUT RETRY LIMIT REACHED
			X'F066'	NO DATA RECEIVED TIMEOUT RETRY LIMIT REACHED
			X'F067'	INVALID RESPONSE TO TTD RETRY LIMIT REACHED
			X'F068'	INVALID RESPONSE TO WACK RETRY LIMIT REACHED
			X'F069'	TTD/WACK NO RESPONSE TIMEOUT RETRY LIMIT REACHED
			X'F06A'	TRANSMIT RETRY LIMIT REACHED
			X'F06B'	ENQ RECEIVED TO ACK SENT RETRY LIMIT REACHED
			X'F06C'	UNRECOGNIZABLE DATA RECEIVED RETRY LIMIT REACHED

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F06D'	ISOLATED PACING MESSAGE FAILURE
		X'F06E'	INVALID PIU (PATH INFORMATION UNIT) RECEIVED
		X'F06F'	SYNCHRONIZATION LOST AND NOT RECOVERED
		X'F070'	FILE WRITE PRE-READ ERROR
		X'F071'	FILE WRITE ERROR
		X'F072'	FILE READ ERROR
		X'F073'	FILE NAME INACCESSIBLE
		X'F074'	DETAILED DATA CONTAINS PATH INFORMATION FOR THE FILE – PATH SHOWN MAY BE INCOMPLETE
		X'F075'	SERVER REQUESTED OPERATION
		X'F076'	SYSTEM FILE LOCK THRESHOLD REACHED
		X'F077'	SYSTEM RECORD LOCK THRESHOLD REACHED
		X'F078'	WORKSTATION FILE LOCK THRESHOLD REACHED
		X'F079'	WORKSTATION RECORD LOCK THRESHOLD REACHED
		X'F07A'	ABNORMAL SERVER SHUTDOWN MAY OCCUR
		X'F07B'	WORKSTATION CONCURRENT DIRECTORY SEARCH THRESHOLD REACHED
		X'F07C'	BACK-UP BATTERY CRITICALLY LOW – SERVER WILL BE SHUT DOWN UNLESS COMMERCIAL POWER IS RESTORED
		X'F07D'	TRANSMIT/RECEIVE BUFFER LIMIT REACHED – BUFFERS ARE NOT BEING RELEASED
		X'F07E'	SERVER SOFTWARE COMPONENT MAY HAVE EXCESSIVE MEMORY REQUIREMENTS
		X'F080'	CSMA/CD ADAPTER WOULD NOT COMPLETE DIRECT MEMORY ACCESS
		X'F081'	CSMA/CD ADAPTER WOULD NOT RESET
		X'F082'	SERVICE CONDITION – HARDWARE FAILURE, NO APPLICATIONS AFFECTED
		X'F083'	MODERATE CONDITION – REDUNDANT HARDWARE FAILURE, POSSIBLE PERFORMANCE DEGRADATION
		X'F084'	SERIOUS CONDITION – PRIMARY I/O SUBSYSTEM RESOURCE DISABLED
		X'F085'	ACUTE CONDITION – MAJOR I/O SUBSYSTEM RESOURCE DISABLED
		X'F0A0'	DEFECTIVE DISK SECTOR CIRCUMVENTED ON (detailed data qualifier) *Note:* The qualifier identifies the drive.
		X'F0A1'	ERROR OCCURRED READING FROM FILE (detailed data qualifier)
		X'F0A2'	ERROR OCCURRED WRITING TO FILE (detailed data qualifier)

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F0A3'	FAILURE OCCURRED ON (detailed data qualifier) *Note:* The qualifier identifies the location of the failure being reported. It may identify the processor on which a failure occurred, e.g., the failing communication control unit in a communication controller. It may also identify a particular link segment within a multi-segment link connection.
		X'F0A4'	XID NEGOTIATION FAILED WITH (detailed data qualifier) *Note:* The qualifier specifies the SNA sense data identifying why the negotiation failed.
		X'F0A5'	COMPONENT OF (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the equipment or service belongs.
		X'F0A6'	BAD LINE QUALITY ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the line belongs.
		X'F0A7'	BOTH MODEMS DETECTED IMPULSE HITS ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the impulse hits were detected.
		X'F0A8'	NO LPDA RESPONSE FROM THE LOCAL MODEM ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'F0A9'	NO LPDA RESPONSE FROM THE REMOTE MODEM ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		X'F0AA'	NO LPDA RESPONSE FROM THE LOCAL DSU/CSU ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'F0AB'	NO LPDA RESPONSE FROM THE REMOTE DSU/CSU ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
		X'F0AC'	INCORRECT LPDA RESPONSE FROM THE LOCAL DSU/CSU ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'F0AD'	INCORRECT LPDA RESPONSE FROM THE REMOTE DSU/CSU ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F0AE'	BIPOLAR ERRORS DETECTED BY LOCAL DSU/CSU ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
		X'F0AF'	BIPOLAR ERRORS DETECTED BY REMOTE DSU/CSU ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
		X'F0B0'	IMPULSE HITS DETECTED BY THE LOCAL MODEM ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the hits were detected.
		X'F0B1'	LOCAL DSU/CSU DETECTED REMOTE DSU/CSU FAILURE ALARM ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
		X'F0B2'	LOCAL MODEM DETECTED REMOTE MODEM FAILURE TONE ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
		X'F0B3'	MODEMS ON (detailed data qualifier) IN BACK-UP SPEED *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
		X'F0B4'	(detailed data qualifier) LOCAL DSU/CSU RECEIVED OUT OF FRAME DDS NETWORK CODE *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
		X'F0B5'	(detailed data qualifier) REMOTE DSU/CSU RECEIVED OUT OF FRAME DDS NETWORK CODE *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
		X'F0B6'	(detailed data qualifier) LOCAL DSU/CSU RECEIVED OUT OF SERVICE DDS NETWORK CODE *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
		X'F0B7'	(detailed data qualifier) REMOTE DSU/CSU RECEIVED OUT OF SERVICE DDS NETWORK CODE *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
		X'F0B8'	(detailed data qualifier) LOCAL DSU/CSU DETECTED DDS NETWORK LOOPBACK ACTIVE *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
		X'F0B9'	(detailed data qualifier) REMOTE DSU/CSU DETECTED DDS NETWORK LOOPBACK ACTIVE *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F0BA'	INCORRECT LPDA RESPONSE FROM THE LOCAL MODEM ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
		X'F0BB'	INCORRECT LPDA RESPONSE FROM THE REMOTE MODEM ON (detailed data qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		X'F0BC'	BIND REJECTED WITH (detailed data qualifier) *Note:* The qualifier identifies the SNA sense data that describes the error.
		X'F0BD'	TRANSMISSION SYSTEM (detailed data qualifier) RECEIVED *Note:* The qualifier identifies the alarm signal.
		X'F0BE'	BIND PROCESSING FAILED WITH (detailed data qualifier) *Note:* The qualifier identifies the SNA sense data that describes the error.
		X'F0BF'	ERROR WRITING TO DIRECTORY ON (detailed data qualifier) *Note:* The qualifier identifies the volume.
		X'F0C0'	RESOURCE LIMIT REACHED (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the resource name and the reason code.
		X'F0C1'	(detailed data qualifier) ON (detailed data qualifier) REQUESTED OPERATION *Note:* The qualifiers identify the user and the connection.
		X'F0C2'	UNABLE TO OPEN/CREATE (detailed data qualifier) ON (detailed data qualifier) *Note:* The qualifiers identify the file name and volume.
		X'F0C3'	UNABLE TO WRITE TO (detailed data qualifier) ON (detailed data qualifier) *Note:* The qualifiers identify the file name and volume.
		X'F0C4'	(detailed data qualifier) PERCENT OF (detailed data qualifier) IN USE *Note:* The first qualifier is a number. The second qualifier identifies a resource.
		X'F100'	(no display): Additional message data
		X'F101'	SERVER IS RUNNING ON BATTERY POWER
		X'F1A0'	ERROR WRITING NEWLY ALLOCATED DIRECTORY BLOCK ON (detailed data qualifier) *Note:* The qualifier identifies the volume.
		X'F1A1'	DISK WRITE ERRORS ON (detailed data qualifier) *Note:* The qualifier identifies the volume.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F1A2'	ERROR WRITING FILE ALLOCATION TABLE FOR (detailed data qualifier) *Note:* The qualifier identifies the volume.
		X'F1A3'	(detailed data qualifier) LOCK REQUESTS FAILED *Note:* The qualifier identifies the number of lock violations.
		X'F1A6'	SERVER IS RUNNING ON BATTERY POWER — (detailed data qualifier) MINUTES REMAIN *Note:* The qualifier identifies the number of minutes of remaining battery power.
		X'F1A7'	(detailed data qualifier) HAS BEEN SHUT DOWN *Note:* The qualifier identifies the file server name.
		X'F1A8'	(detailed data qualifier) INACCESSIBLE DUE TO LAN ADAPTER FAILURE *Note:* The qualifier identifies the number of failed memory allocation requests.
		X'F1A9'	(detailed data qualifier) DISMOUNTED DUE TO HARDWARE FAILURE *Note:* The qualifier identifies the volume.
		X'F1AA'	(detailed data qualifier) RECEIVED *Note:* The qualifier identifies the maintenance indicator.
		X'F1AB'	UNSOLICITED RESPONSE FROM (detailed data qualifier) *Note:* The qualifier identifies the resource name.
		X'F1AC'	(detailed data qualifier) NOT IN VALID STATE
		X'F1AD'	(detailed data qualifier) PRIMARY PARTITION NOT FOUND *Note:* The qualifier identifies the drive.
		X'F1AE'	(detailed data qualifier) SECONDARY PARTITION NOT FOUND *Note:* The qualifier identifies the drive.
		X'F1AF'	COMPARE OF MIRRORED PARTITIONS ON (detailed data qualifier) FAILED *Note:* The qualifier identifies the drive.
		X'F1B0'	PARTITION FAILURE AND SHUTDOWN ON (detailed data qualifier) *Note:* The qualifier identifies the drive.
		X'F1B1'	EXCESSIVE DISK FAILURES ON (detailed data qualifier) *Note:* The qualifier identifies the drive.
		X'F1B2'	COMPLETE DISK FAILURE ON (detailed data qualifier) *Note:* The qualifier identifies the drive.
		X'F800'	(no display): Additional message data *Note:* The X'F8xx' range is used for additional messages that are identical for User, Install, and Failure Causes.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F8A0'	PROBLEM DETECTED BY (detailed data qualifier) *Note:* The qualifier identifies the file server name.
		X'F8C0'	FAILING COMPONENT IS IDENTIFIED BY (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the failing component by means of its logical location, e.g., its port number and device address.
		X'F8D0'	PROBLEM IS RELATED TO THE CONTROLLER LOCATED AT (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the controller location as follows: Q1 = RACK Q2 = UNIT (within a rack) Q3 = CARD SLOT (within a unit)
		X'F8D1'	PROBLEM IS RELATED TO DEVICE IDENTIFIED BY (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the following: Q1 = CARD SLOT Q2 = CONTROLLER ID Q3 = DEVICE ID
		X'F8E0'	FAILING COMPONENT IS IDENTIFIED BY (sf83 product text)
	X'FFFF'	Reserved	

Cause Undetermined (X'97') Alert MS Subvector

This subvector transports code points for stored text detailing the recommended actions to be taken when no probable user, install, or failure causes for the Alert condition can be identified. It may also transport additional detailed data, to be inserted into the text indexed by the recommended action code points.

Cause Undetermined (X'97') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Cause Undetermined subvector
1		Key: X'97'

Cause Undetermined (X'97') Alert MS Subvector

Byte	Bit	Content
2–p		One or more subfields containing recommended action data, as described in "Network Alert (X'0000') Common Subfields" on page 9-131. X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended *Note:* Subfield X'81' is always present. Depending on the code points present in the X'81' subfield: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

Detailed Data (X'98') Alert MS Subvector

This subvector transports product specific detailed data.

Detailed Data (X'98') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Detailed Data subvector
1		Key: X'98'
2–p		One or more subfields containing detailed data, as described below for keys X'00'-X'7F' and in "Network Alert (X'0000') Common Subfields" on page 9-131 for keys X'80'-X'FE': X'01' Qualified Message Data X'82' Detailed Data X'85' Detailed Data Extended *Note:* Any number of instances of the X'01' subfield may be present. Any number of instances of either the X'82' or the X'85' subfield may be present (the X'82' and X'85' subfields are mutually exclusive within a subvector). Each X'01' subfield contains a number indicating how many subsequent X'82'/X'85' subfields are associated with it. It is not required that every X'82'/X'85' subfield be associated with a X'01' subfield.

Qualified Message Data (X'01') Detailed Data Subfield

This subfield contains an index to a complete message stored at an Alert receiver, as well as an indication of how many qualifiers are to be inserted into the message.

Qualified Message Data (X'01') Detailed Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Qualified Message Data subfield
1		Key: X'01'
2		Product ID code: The structure of this field is identical to that present in the Product Set ID Index (X'83') subfield.
	0–3	Product ID subvector code: a code point that specifies (1) the type of Product ID subvector being indexed (hardware or software), and (2) the particular data to be extracted from this subvector *Note:* See "Product Identifier (X'11') MS Common Subvector" on page 9-355 for the criteria distinguishing hardware and software Product ID subvectors. X'0'–X'1' reserved X'2' machine type or hardware product common name from a hardware Product ID subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'5' machine type or hardware product common name plus model number from a hardware Product ID subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'9' software product common name from a software Product ID subvector
	4	Product set ID indicator: An indication of which Product Set ID (PSID) contains the Product ID subvector being indexed 0 Alert sender PSID 1 indicated resource PSID
	5–7	Count: A 3-digit binary number that indicates which Product ID subvector, of the type specified by the Product ID Subvector Code, is being indexed within the PSID specified by the Product Set ID Indicator. *Note:* This count applies only to Product ID Subvectors of the type specified by the Product ID Subvector Code. If, for example, the code is X'2' (specifying a hardware Product ID) then only hardware X'11's are counted: a count of X'3' would thus index the third hardware Product ID within the PSID indicated by the Product Set ID Indicator.

Qualified Message Data (X'01') Detailed Data Subfield

Byte	Bit	Content
3		Data ID: A code point indicating the type of the message to be constructed from the data carried in the subfield. The English text associated with each code point, or its national language equivalent, is displayed in conjunction with the message. Defined codes are: X'01' OPERATOR ERROR MESSAGE X'E0'–X'EF' Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.
4		Message code encoding: a code point indicating how the accompanying message code is encoded. This data is included because an Alert receiver has the option of displaying the message code itself in addition to the message that it indexes. Defined codes are: X'00' hexadecimal: the message code is to be displayed as hexadecimal digits X'11' Coded Graphic Character Set 00640–00500 plus: The data is to be decoded using Coded Graphic Character Set 00640–00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," plus three additional code points: X'5B' = "$" (dollar sign); X'7B' = "#" (pound or number sign); X'7C' = "@" (at sign)
5		Qualifier count: a binary number indicating how many qualifiers are associated with this message *Note:* The qualifiers are specified in either X'82' or X'85' subfields following this X'01' subfield, and are substituted into the message in the order in which the X'82' or X'85' subfields are present.
6–q		Message code, encoded as specified in byte 4 above *Note:* This message code is limited to eight bytes.

Detail Qualifier (EBCDIC) (X'A0') Alert MS Subvector

This subvector supplies variables for the Alert function in EBCDIC form that can be inserted on the Alert Detail screens. This subvector and the Detail Qualifier (hexadecimal) subvector (X'A1') are identical in function and format except that this subvector contains EBCDIC codes. *Note:* The Detail Qualifier (X'A0'-X'A1') subvectors are displayed in the order that they appear in the Alert major vector.

Detail Qualifier (EBCDIC) (X'A0') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Detail Qualifier subvector
1		Key: X'A0'
2−p		Detail qualifier: a type-AE symbol-string that qualifies a reference on the Alert Detail screen *Note:* Each qualifier is p-1 bytes in length, but only one qualifier is used per Detail Qualifier subvector. All qualifiers include only codes, numbers, or internationally recognized terms that do not require translation. The coding is not interpreted by the Alert display mechanism.

Detail Qualifier (Hexadecimal) (X'A1') Alert MS Subvector

This subvector supplies variables for the Alert function in hexadecimal form that can be inserted on the Alert Detail screens. This subvector and the Detail Qualifier (EBCDIC) subvector (X'A0') are identical in function and format except that this subvector contains codes in hexadecimal. *Note:* The Detail Qualifier (X'A0' − X'A1') subvectors are displayed in the order that they appear in the Alert major vector.

Detail Qualifier (Hexadecimal) (X'A1') Alert MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Detail Qualifier subvector
1		Key: X'A1'
2−p		Detail qualifier: a type-G symbol-string

Network Alert (X'0000') Common Subfields

The following table shows, by key value, the subfields common to the Network Alert subvectors, and the subvectors in which each can occur.

Key	Subfield	Applicable Network Alert Subvectors
X'81'	Recommended Actions	User Causes subvector, Install Causes subvector, Failure Causes subvector, Cause Undetermined subvector
X'82'	Detailed Data	User Causes subvector, Install Causes subvector, Failure Causes subvector, Cause Undetermined subvector, Detailed Data subvector

X'83'	Product Set ID Index	User Causes subvector, Install Causes subvector, Failure Causes subvector, Cause Undetermined subvector
X'84'	Resource List Index	User Causes subvector, Install Causes subvector, Failure Causes subvector, Cause Undetermined subvector
X'85'	Detailed Data Extended	User Causes subvector, Install Causes subvector, Failure Causes subvector, Cause Undetermined subvector, Detailed Data subvector

Recommended Actions (X'81') Network Alert Common Subfield

This subfield contains code points for stored text describing recommended actions to be taken to rectify an Alert condition.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Recommended Actions subfield
1		Key: X'81'
2−q		2-byte recommended action code points. Each code point provides an index to pre-defined text, describing the recommended action, that is displayed at the Alert receiver. An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are listed below. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules). The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the X'94' - X'97' subvectors, in the order in which they are to be associated with the gaps specified in the X'81' subfield. **Note:** The X'82' and X'85' subfields *cannot* both be used in the same subvector.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content

The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'81' subfield in the same subvector, in the order in which they are to be associated with the gaps specified in the X'81' subfield.

The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'81' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'81' subfield.

The third digit of each recommended action code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows:

X'xx0x' – X'xx9x': No detailed data qualifiers.

X'xxAx' – X'xxBx': One detailed data qualifier.

X'xxCx': Two detailed data qualifiers.

X'xxDx': Three detailed data qualifiers.

X'xxEx': One X'83' subfield.

X'xxFx': One X'84' subfield.

Defined codes are:

X'0000' PERFORM PROBLEM DETERMINATION PROCEDURES: Refer to the problem determination documentation provided for this condition and follow the specified procedures

- X'0001' RUN ONLINE PROBLEM DETERMINATION
- X'0002' INVOKE INTENSIVE MODE RECORDING
- X'0003' DETERMINE THE REASON FOR THE LINE SHUTDOWN
- X'0004' DETERMINE THE REASON FOR THE LOOP SHUTDOWN
- X'0005' PERFORM SNA DATA STREAM PROBLEM DETERMINATION PROCEDURES
- X'0006' USE MICROCODE DUMP AND SYSTEM PROCEDURES TO CREATE AN APAR
- X'0007' PERFORM REMOTE MODEM PROBLEM DETERMINATION
- X'0008' PERFORM REMOTE DSU/CSU PROBLEM DETERMINATION
- X'0009' PERFORM OUTBOUND LINE PROBLEM DETERMINATION

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'000A'	PERFORM INBOUND LINE PROBLEM DETERMINATION
		X'000B'	DETERMINE RESOURCE NAME OF THE OTHER REMOTE NODE
		X'000C'	RUN MODEM SELF TEST WITH WRAP PLUG FROM KEYPAD
		X'000D'	RUN DSU/CSU SELF TEST WITH WRAP PLUG FROM CONTROL PANEL
		X'000E'	RUN MODEM AND LINE STATUS TEST
		X'000F'	RUN DSU/CSU AND LINE STATUS TEST
		X'0010'	RUN LINE TEST
		X'0011'	RUN LINE ANALYSIS TEST
		X'0012'	RUN TRANSMIT/RECEIVE TEST
		X'0013'	RUN REMOTE NODE TEST
		X'0014'	RUN REMOTE NODE-DCE INTERFACE WRAP TEST
		X'0015'	INVESTIGATE INTERFERENCE FROM OTHER PORT ON LOCAL MODEM, IF FAN-OUT INSTALLED
		X'0016'	REVIEW MODEM AND LINE STATUS DATA
		X'0017'	RUN VERIFY COMMAND
		X'0018'	REVIEW USER'S DISK STORAGE USAGE
		X'0019'	REVIEW STATISTICS ASSOCIATED WITH THE LINK
		X'00A0'	PERFORM TRANSMISSION LINE PROBLEM DETERMINATION PROCEDURES ON (detailed data qualifier) *Note:* The qualifier identifies the failing transmission line.
		X'00A1'	REVIEW (detailed data qualifier) *Note:* The qualifier identifies the entity to be reviewed.
		X'00B0'	PERFORM PROBLEM DETERMINATION PROCEDURE FOR (detailed data qualifier) *Note:* The qualifier identifies a value, such as a system reference code, that selects a problem determination procedure to be performed.
		X'00B1'	PERFORM PROBLEM DETERMINATION PROCEDURE AT THE REPORTING LOCATION FOR (detailed data qualifier) *Note:* This code point differs from X'00B0' in that it specifies that the indicated problem determination procedure is one that must be performed locally, at the site of the failure.
		X'00B2'	RUN THE FOLLOWING AT THE REPORTING LOCATION (detailed data qualifier) *Note:* The qualifier identifies the command, program, error recovery procedure, etc.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content		
			X'00B3'	PERFORM (detailed data qualifier) PROBLEM DETERMINATION VIA A REMOTE CONSOLE SESSION: Interactive product error analysis is required. *Note:* The qualifier identifies the scope the problem determination procedure is expected to be.
			X'00C0'	RUN SELF TEST WITH WRAP PLUG ON (detailed data qualifier) REMOTE MODEM AND (detailed data qualifier) LOCAL MODEM FROM MODEM KEYPADS *Note:* The qualifiers identify the link segment level (LSL) on which the modems belong.
			X'00C1'	RUN SELF TEST WITH WRAP PLUG ON (detailed data qualifier) REMOTE DSU/CSU AND (detailed data qualifier) LOCAL DSU/CSU FROM DSU/CSU CONTROL PANELS *Note:* The qualifiers identify the link segment level (LSL) on which the DSU/CSUs belong.
			X'00C2'	RUN SELF TEST WITH WRAP PLUG ON (detailed data qualifier) REMOTE MODEM (FROM KEYPAD) AND (detailed data qualifier) LOCAL DSU/CSU (FROM CONTROL PANEL) *Note:* This code point is used on a multi-segment link with modems on one segment and DSU/CSUs on the other segment.
			X'00C3'	RUN SELF TEST WITH WRAP PLUG ON (detailed data qualifier) REMOTE DSU/CSU (FROM CONTROL PANEL) AND (detailed data qualifier) LOCAL MODEM (FROM KEYPAD) *Note:* The qualifiers identify the link segment level (LSL) on which the equipment belongs. This code point is used on a multi-segment link with modems on one segment and DSU/CSUs on the other segment.
			X'00C4'	RUN DSU/CSU AND LINE STATUS TEST ON (detailed data qualifier) AND (detailed data qualifier) *Note:* The qualifiers identify the link segment level (LSL) on which the lines belong.
			X'00C5'	RUN DCE AND LINE STATUS TEST ON (detailed data qualifier) AND (detailed data qualifier) *Note:* The qualifiers identify the link segment level (LSL) on which the lines belong.
			X'00E1'	PERFORM (sf83 product text) PROBLEM DETERMINATION PROCEDURES
		X'0100'	VERIFY	
			X'0101'	VERIFY X.25 SUBSCRIPTION NUMBER
			X'0102'	VERIFY ADAPTER ADDRESS DEFINITION
			X'0103'	VERIFY TELEPHONE NUMBER
			X'0104'	CHECK FOR CORRECT MICROCODE FIX

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'0105'	REQUEST VERIFICATION OF MANAGEMENT SERVER REPORTING LINKS
		X'0106'	CHECK REMOTE NODE FOR TC ACTIVE
		X'0107'	CHECK OTHER REMOTE NODE FOR TC ACTIVE
		X'0108'	CHECK REMOTE NODE FOR RTS ACTIVE
		X'0109'	CHECK OTHER REMOTE NODE FOR RTS ACTIVE
		X'010A'	CHECK CONFIGURATION OF THE SENDING NODE
		X'010B'	CHECK CONFIGURATION OF THE SENDING NODE AND OF THE MODEMS
		X'010C'	CHECK CONFIGURATION OF THE SENDING NODE AND OF THE DSU/CSU'S
		X'010D'	CHECK CONFIGURATION OF THE REMOTE NODE
		X'010E'	CHECK FOR CONFIGURATION MISMATCH BETWEEN APPLICATION CONTROL BLOCK AND CARD NUMBER
		X'010F'	CHECK CONFIGURATION OF THE OTHER REMOTE NODE
		X'0110'	CHECK CONFIGURATION OF THE REMOTE NODE AND REMOTE MODEM
		X'0111'	CHECK RTS GENERATION PARAMETER
		X'0112'	VERIFY THAT THE FAN-OUT FEATURE IS INSTALLED
		X'0113'	VERIFY THAT REMOTE NODE PROVIDES THE DCE EXTERNAL CLOCK
		X'0114'	CHECK FILE DIRECTORY STRUCTURE
		X'0115'	REVIEW MEMORY USAGE
		X'0116'	VERIFY APPLICATION PROGRAM LOCK USAGE
		X'0117'	CHECK FOR CONFIGURATION MISMATCH BETWEEN APPLICATION CONTROL BLOCK AND PROGRAM
		X'0118'	CHECK FOR CONFIGURATION MISMATCH BETWEEN RESOURCE DIRECTORY AND TABLES AND QUEUES DEFINITION FILE The file containing descriptive information about software tables and various system queues conflicts with the resource directory entries.
		X'0119'	CHECK PROCESSES IN MESSAGE MEMORY POOL
		X'011A'	CHECK PROCESSES IN CO-PROCESSOR MESSAGE MEMORY POOL
		X'011B'	CHECK PROCESSES IN WORK MEMORY POOL
		X'011C'	VERIFY ACCESS AUTHORITY
		X'011D'	VERIFY ADAPTER INSTALLATION
		X'011E'	VERIFY DESTINATION RESOURCE IS DEFINED
		X'011F'	VERIFY DESTINATION RESOURCE IS DEFINED IN RESOURCE DIRECTORY
		X'0120'	VERIFY APPLICATION PROGRAM DIRECTORY SEARCH USAGE

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'0121'	IDENTIFY OFFENDING SOFTWARE COMPONENT
		X'0122'	VERIFY DESTINATION RESOURCE IS VALID
		X'0123'	REVIEW SOFTWARE COMPONENT MEMORY USAGE
		X'0124'	VERIFY DESTINATION RESOURCE NAME
		X'0125'	VERIFY COMMAND
		X'0126'	VERIFY COMMUNICATION ACCESS CONTROL BLOCK CONFIGURATION
		X'0127'	VERIFY COMMUNICATION PATH IS WORKING
		X'0128'	VERIFY COMMUNICATION PROTOCOL PROGRAM IS WORKING
		X'0129'	VERIFY COMMUNICATION PATH DEFINITION
		X'012A'	VERIFY CONTROL BLOCK TYPE
		X'012B'	VERIFY CONTROL NODE IS ACTIVE
		X'012C'	VERIFY CORRECT MEMORY IMAGE FILE SPECIFIED IN COMMAND
		X'012D'	VERIFY DESTINATION NODE IS ACTIVE
		X'012F'	VERIFY FILE IS DEFINED
		X'0130'	VERIFY SIZE OF MEMORY POOLS
		X'0131'	VERIFY FILE PATH
		X'0132'	VERIFY FILE EXISTS AND IS NOT IN USE
		X'0133'	CHECK COMMUNICATION DRIVER CONFIGURATION FILE
		X'0134'	VERIFY FILE WAS CREATED WITH FILE SUPPORT: Verify the file was created as required with the file support function.
		X'0135'	VERIFY FILE IS VALID
		X'0136'	VERIFY FILE SUPPORT IS STARTED
		X'0137'	VERIFY DESTINATION RESOURCE SERVER NAME
		X'0138'	VERIFY DESTINATION SERVER IS DEFINED
		X'0139'	VERIFY DEVICE CONFIGURATION
		X'013A'	VERIFY DEVICE DATA MANAGEMENT SUPPORT IS OPERATING CORRECTLY
		X'013C'	VERIFY DEVICE NAME IS VALID
		X'013D'	VERIFY DIRECTORY REQUEST
		X'013E'	VERIFY ORIGINATOR RESOURCE IS DEFINED IN RESOURCE DIRECTORY
		X'013F'	VERIFY ORIGINATOR RESOURCE IS DEFINED IN SUBNET CONFIGURATION
		X'0140'	VERIFY MEMORY LOCATION AND CORRECT IF NECESSARY
		X'0141'	VERIFY ORIGINATOR RESOURCE IS DEFINED IN SYSTEM CONFIGURATION
		X'0142'	VERIFY ORIGINATOR RESOURCE IS STARTED
		X'0143'	VERIFY ORIGINATOR RESOURCE IS VALID
		X'0144'	VERIFY LINE NAME

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'0145'	VERIFY LOGICAL ADDRESS IS DEFINED
		X'0146'	VERIFY MANUFACTURING AUTOMATION PROTOCOL IS LOADED
		X'0147'	VERIFY MAXIMUM MESSAGE SIZE IN NODE INFORMATION BLOCK
		X'0148'	VERIFY MESSAGE FILE IS DEFINED
		X'0149'	VERIFY MESSAGE IS DEFINED
		X'014A'	VERIFY MINIDISK REQUEST
		X'014B'	VERIFY ORIGINATOR RESOURCE IS DEFINED
		X'014C'	VERIFY DESTINATION RESOURCE IS STOPPED
		X'014D'	VERIFY SECURITY SUBNET CONFIGURATION FOR CONTROL NODE
		X'014E'	VERIFY APPLICATION INITIALIZATION/SHUT-DOWN SEQUENCE
		X'014F'	VERIFY SERVER RESOURCE IS DEFINED
		X'0150'	VERIFY SERVER RESOURCE IS VALID
		X'0151'	VERIFY LOGON PANEL IS DEFINED IN OPERATOR INTERFACE PANEL LIBRARY
		X'0152'	VERIFY ORIGINATOR RESOURCE NAME
		X'0153'	VERIFY PARAMETER
		X'0154'	VERIFY PHYSICAL RESOURCE IS DEFINED
		X'0155'	VERIFY PROGRAM IS LOADED
		X'0156'	VERIFY QUEUE NUMBER IS VALID
		X'0157'	VERIFY REMOTE NODE IS ACTIVE
		X'0159'	VERIFY VIDEO CONTROL BLOCK IS VALID
		X'015A'	VERIFY SUBNET CONFIGURATION IS VALID
		X'015B'	VERIFY SUBSERVIENT NODE IS ACTIVE
		X'015C'	VERIFY CONFIGURATION FILE
		X'015D'	VERIFY SYSTEM CONFIGURATION IS VALID
		X'015E'	VERIFY TABLE DISTRIBUTION GROUP EXISTS ON NODE WHERE FAILURE OCCURRED
		X'015F'	VERIFY TABLE IS DEFINED
		X'0160'	VERIFY TABLES AND QUEUES DEFINITION FILE The file containing descriptive information about software tables and various system queues should be checked for errors.
		X'0161'	VERIFY TASK NUMBER
		X'0162'	VERIFY VIDEO CONTROL BLOCK IS DEFINED
		X'0163'	VERIFY BACK-UP D-CHANNEL OPERATION
		X'0164'	REVIEW SERVER AUDIT TRAIL
		X'0165'	CHECK SIZE AND CHANGE IF SET TOO LOW
		X'0166'	INCREASE USER ACCOUNT LIMITATION
		X'0167'	REVIEW SERVER ERROR LOG
		X'0168'	CHECK DOMAIN CONTROLLER
		X'0169'	CHECK FOR CORRECT LOAD MODULE
		X'0170'	VERIFY IMPENDING EVENT SHOULD OCCUR
		X'0171'	CHECK FOR SYSTEM CONTENTION

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'0172'	CHECK DASD SUBSYSTEM UNIT ENVIRONMENT FOR EXCESSIVE TEMPERATURE
		X'01A0'	CALL (detailed data qualifier) AND CHECK VOICE RECORDING *Note:* The qualifier identifies the telephone number that will be answered by the voice response unit.
		X'01A1'	VERIFY (detailed data qualifier) *Note:* The qualifier identifies the entity to be verified.
		X'01A2'	VERIFY (detailed data qualifier) IN RESOURCE DIRECTORY *Note:* The qualifier identifies the resource name.
		X'01A3'	VERIFY (detailed data qualifier) IS DEFINED *Note:* The qualifier identifies the resource type.
		X'01A4'	VERIFY (detailed data qualifier) IS DEFINED IN SUBNET CONFIGURATION *Note:* The qualifier identifies the node name.
		X'01A5'	VERIFY (detailed data qualifier) IS DEFINED IN SYSTEM CONFIGURATION *Note:* The qualifier identifies the node name.
		X'01A6'	CHECK FOR CONFIGURATION MISMATCH BETWEEN SYSTEM CONFIGURATION FILE AND (detailed data qualifier) *Note:* The qualifier identifies the file name.
		X'01A7'	VERIFY (detailed data qualifier) IS VARIED ON *Note:* The qualifier identifies the entity required to be varied on.
		X'01A8'	VERIFY (detailed data qualifier) WAS CREATED *Note:* The qualifier identifies the entity which should have been created.
		X'01A9'	VERIFY (detailed data qualifier) IS ACCESSIBLE *Note:* The qualifier identifies the entity which should be accessible, for example a file.
		X'01C0'	CHECK FOR CONFIGURATION MISMATCH BETWEEN THE (detailed data qualifier) REMOTE MODEM AND (detailed data qualifier) LOCAL MODEM *Note:* The qualifiers identify the link segment level (LSL) on which the modems belong.
		X'01C1'	CHECK FOR CONFIGURATION MISMATCH BETWEEN THE (detailed data qualifier) REMOTE MODEM AND (detailed data qualifier) LOCAL DSU/CSU *Note:* The qualifiers identify the link segment level (LSL) on which the modem and the DSU/CSU belong. This code point is used on a multi-segment link with modems on one segment and DSU/CSUs on the other segment.
		X'01D0'	VERIFY (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the entities to be verified.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
			X'01E1' VERIFY (sf83 product text) IS RUNNING
		X'0200'	CHECK POWER: Check the electrical power supply for the device
			X'0201' CHECK POWER CONTROL MODE: Check the power control mode for the device, i.e. is the device set to be powered on/off locally from the control panel, remotely from the host, remotely from the network, etc.
		X'0300'	CHECK FOR DAMAGE: Check for damage to the specified resource
			X'0301' CHECK CABLE AND ITS CONNECTIONS X'0302' CHECK CABLES AND THEIR CONNECTIONS X'0303' CHECK PHYSICAL INSTALLATION X'0306' CHECK TAPE PATH TO READ/WRITE HEAD FOR OBSTRUCTION X'0310' CHECK UNINTERRUPTIBLE POWER SUPPLY
		X'0400'	RUN APPROPRIATE TEST: Refer to the appropriate documentation for this condition and run the tests specified for this problem
			X'0401' RUN CONSOLE TEST X'0402' RUN CONSOLE LINK TEST X'0403' RUN MODEM TESTS
		X'0500'	RUN APPROPRIATE TRACE: Refer to the appropriate documentation for this condition and run the traces specified for this problem.
			X'0501' RUN COMMUNICATION LINE DATA TRACE
		X'0600'	OBTAIN DUMP: Perform the required operations to obtain a storage dump (copy to external storage of the processor main storage). The dump will be required by support personnel or service personnel in order to resolve the problem
			X'0601' TRANSFER AND PRINT MOSS DUMP X'0602' TRANSFER AND PRINT CONTROL PROGRAM DUMP X'0603' TRANSFER AND PRINT LINE ADAPTER DUMP X'0604' TRANSFER AND PRINT CHANNEL ADAPTER DUMP X'0605' TRANSFER AND PRINT TOKEN RING COUPLER DUMP X'0610' DUMP CONTROL PROGRAM X'0611' DUMP CHANNEL ADAPTER MICROCODE X'0612' DUMP LINE ADAPTER MICROCODE X'0613' DUMP MOSS MICROCODE X'0614' DUMP TOKEN RING COUPLER
		X'0700'	NO ACTION NECESSARY: For a given cause, no action is necessary, e.g., the problem caused was transitory
			X'0701' IF SNBU JUST DISCONNECTED THEN IGNORE

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
			X'07A0' CONTROL NODE (detailed data qualifier) NOW RESPONDING *Note:* The qualifier identifies the resource name.
			X'07A1' APPLICATION (detailed data qualifier) NOW RESPONDING *Note:* The qualifier identifies the resource name.
			X'07A2' PRINT SERVER (detailed data qualifier) NOW RESPONDING *Note:* The qualifier identifies the resource name.
		X'1000'	PERFORM PROBLEM RECOVERY PROCEDURES: Refer to the problem recovery documentation provided for this condition and follow the specified procedures
			X'1001' REFER TO OPERATOR'S GUIDE FOR CORRECTIVE ACTION
			X'1002' FOLLOW LOCAL BACKUP PROCEDURE
			X'1003' PERFORM LOOP PROBLEM RECOVERY PROCEDURES
			X'1004' PERFORM LAN PROBLEM RECOVERY PROCEDURES
			X'1005' PERFORM DISK FILE ERROR RECOVERY PROCEDURES
			X'1006' FOR SINGLE DRIVE FAILURES, MOVE PACK TO ANOTHER DRIVE AND RERUN SAME JOB
			X'1007' REQUEST RESET OF RING ERROR MONITOR COMPONENT
			X'1008' REQUEST REINITIALIZATION OF LAN MANAGER
			X'1009' ATTEMPT TO REOPEN THE ADAPTER AFTER 30 SECONDS
			X'100A' REORGANIZE THE FILE
			X'100B' LEAVE THE ADDITIONAL WORKSTATION POWERED OFF
			X'100C' POWER OFF ANOTHER WORKSTATION ATTACHED TO THIS CONTROLLER
			X'100D' TRANSFER THE ADDITIONAL WORKSTATION TO ANOTHER CONTROLLER
			X'100E' LOCATE AND ELIMINATE THE SOURCE OF ELECTRICAL INTERFERENCE
			X'100F' CHANGE REGISTRATION NAME OF ONE OF THE COMMUNICATION CONTROL BLOCKS
			X'1010' REDUCE AMOUNT OF AUXILIARY STORAGE USED
			X'1011' INCREASE AMOUNT OF AUXILIARY STORAGE AVAILABLE
			X'1012' INCREASE AMOUNT OF MAIN STORAGE AVAILABLE
			X'1013' POWER OFF REMOTE MODEM
			X'1014' POWER OFF REMOTE DSU/CSU

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'1015'	POWER OFF THEN POWER ON AUTO-CALL UNIT
		X'1016'	REDUCE THE NUMBER OF LINES/DEVICES ATTACHED TO THE SUBSYSTEM: The subsystem is overloaded.
		X'1017'	INSTALL ADDITIONAL MEMORY
		X'1018'	INSTALL ADDITIONAL FILE SERVERS
		X'1019'	PURGE UNUSED FILES
		X'101A'	PRINT THEN CLEAR ERROR LOG
		X'101B'	PRINT THEN CLEAR AUDIT LOG
		X'101C'	DELETE SECOND APPLICATION CONTROL BLOCK FOR MONITOR
		X'101D'	REDUCE TYPES OF AUDIT ENTRIES LOGGED
		X'101E'	CORRECT LOGICAL DEVICE INTERFACE PROCEDURES (LDIP) PROBLEM
		X'101F'	CREATE APPLICATION CONTROL BLOCK FOR INBOUND SERVER
		X'1020'	INCREASE STORAGE MEDIA CAPACITY
		X'1021'	CREATE APPLICATION CONTROL BLOCK FOR OUTBOUND SERVER
		X'1022'	CREATE COMMUNICATION CONTROL BLOCK FOR COMMUNICATION ACCESS CONTROL BLOCK
		X'1023'	CREATE TABLE DISTRIBUTION GROUP TABLE
		X'1024'	DELETE JOBS
		X'1025'	CORRECT CONFIGURATION TO ENABLE DATA TRANSPARENCY IN NODE INFORMATION BLOCK
		X'1026'	CORRECT CONFIGURATION TO ENABLE SECURITY IN NODE INFORMATION BLOCK
		X'1027'	INCREASE NUMBER OF TASK NUMBERS
		X'1028'	INCREASE NUMBER OF TIMERS
		X'1029'	INCREASE SIZE OF CONTROL BLOCK MEMORY POOL
		X'102A'	INCREASE SIZE OF DRIVER BUFFER
		X'102B'	GIVE CONTROL OF ONE COMMUNICATION CONTROL BLOCK TO ANOTHER COMMUNICATION ACCESS CONTROL BLOCK
		X'102C'	HALT DESTINATION RESOURCE To halt, in this sense, means to purge control blocks associated with the resource.
		X'102D'	INCREASE AMOUNT OF VIRTUAL STORAGE AVAILABLE
		X'102E'	INCREASE BUFFER ALLOCATION IN LINE CONTROL BLOCK
		X'102F'	INCREASE TIMEOUT VALUE IN LINE CONTROL BLOCK
		X'1030'	UNLOAD DEPENDENT SOFTWARE PROGRAMS THEN UNLOAD INDICATED PROGRAM

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content		
			X'1031'	UNLOAD OFFENDING SOFTWARE COMPONENT
			X'1032'	INCREASE TIMEOUT VALUE IN REQUEST
			X'1033'	INCREASE SIZE OF MESSAGE MEMORY POOL
			X'1034'	INCREASE SIZE OF CO-PROCESSOR CONTROL BLOCK MEMORY POOL
			X'1035'	INCREASE SIZE OF CO-PROCESSOR MESSAGE MEMORY POOL
			X'1036'	INCREASE SIZE OF CO-PROCESSOR WORK MEMORY POOL
			X'1037'	INCREASE SIZE OF RESOURCE
			X'1038'	INCREASE SIZE OF WORK MEMORY POOL
			X'1039'	TERMINATE USER SESSION
			X'103A'	INCREASE TIMEOUT VALUE IN APPLICATION CONTROL BLOCK
			X'103B'	INCREASE TIMEOUT VALUE IN DATABASE CONTROL BLOCK
			X'103C'	RELEASE ALL LOCKED RECORDS
			X'103D'	SYNCHRONIZE PASSWORDS AT FILE SERVER AND DOMAIN CONTROLLER
			X'103E'	OPEN RESOURCE
			X'103F'	SYNCHRONIZE SYSTEM CLOCK TIMES
			X'1040'	LOG OFF LOW PRIORITY USERS
			X'1041'	RESOLVE IN-DOUBT LOGICAL UNITS OF WORK
			X'1042'	RELEASE LOCKED MEMBER
			X'1043'	PRINT JOBS
			X'1044'	USE DEVICE DATA MANAGEMENT SUPPORT HELP FACILITY FOR MORE INFORMATION ON RETURN CODES
			X'1045'	SELECT VALID MEMORY POOL
			X'1046'	SELECT VALID PARAMETER
			X'1047'	USE ANOTHER DATABASE CONTROL BLOCK
			X'1048'	SELECT VALID REQUEST SIZE
			X'1049'	STOP THEN START LINE CONTROL BLOCK
			X'104A'	STOP PRINTER CONTROL BLOCK
			X'104B'	STOP PRINTER CONTROL BLOCKS NOT IN USE
			X'104C'	STOP THEN START DEVICE CONTROL BLOCK
			X'104D'	DELETE FILE
			X'104E'	RENAME FILE
			X'104F'	LOAD COMMUNICATIONS SUBSYSTEM CONTROLLER
			X'1050'	LOAD CO-PROCESSOR DISPATCHER: Start the co-processor dispatcher resource for the associated co-processor adapter.
			X'1051'	DECREASE REQUEST SIZE
			X'1052'	REDUCE NUMBER OF APPLICATIONS EXECUTING
			X'1053'	RESIZE AND COPY FILE

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'1054'	CLOSE ANY FILES WHICH ARE NOT IN USE
		X'1055'	FREE SPACE ON DESIGNATED DISK DRIVE
		X'1056'	RE-CREATE FILE
		X'1057'	RESTORE OR RESET FILE
		X'1058'	REMOVE PROCESSES FROM SYSTEM
		X'1059'	DELETE MEMBERS FROM TABLE OR QUEUE
		X'105A'	DISCONNECT FROM DATABASE THEN CONNECT TO ANOTHER DATABASE
		X'105B'	DISCONNECT FROM DATABASE THEN CONNECT TO ANOTHER DATABASE
		X'105C'	INCREASE NETWORK CONTROL BLOCKS FOR NETWORK DRIVER
		X'105D'	FORMAT FIXED DISK
		X'105E'	INSTALL FIX FOR CURRENT PROBLEM, IF AVAILABLE
		X'10A0'	RESTORE (detailed data qualifier) FROM BACKUP *Note:* The qualifier identifies the entity to be restored, for example a file.
		X'10A1'	PERFORM (detailed data qualifier) *Note:* The qualifier specifies a problem recovery procedure to be performed.
		X'10A2'	FOLLOW PROBLEM RECOVERY PROCEDURE INDICATED AT PRINTER FOR (detailed data qualifier) *Note:* The qualifier specifies a value that indexes a local problem recovery procedure.
		X'10A3'	FOLLOW PROBLEM RECOVERY PROCEDURE INDICATED AT PRINTER SERVER FOR (detailed data qualifier) *Note:* The qualifier specifies a value that indexes a local problem recovery procedure.
		X'10A4'	FOR CORRECTIVE ACTION REFER TO (detailed data qualifier) *Note:* The qualifier identifies the publication number of a document where corrective actions are described and may only be used when the implementing product will provide a single version of the publication (i.e., the publication will never be translated and therefore the publication number will never be changed).
		X'10A5'	REVIEW (detailed data qualifier) AND UPDATE AS REQUIRED *Note:* The qualifier identifies the file to be reviewed. Any data id describing a type of file may be used in this context.
		X'10A6'	CREATE NEW (detailed data qualifier) *Note:* The qualifier identifies the file. Any data id describing a type of file may be used in this context.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'10A7'	RESTORE (detailed data qualifier) ON DOMAIN CONTROLLER AND FILE SERVER FROM BACKUP FILE *Note:* The qualifier identifies the file. Any data id describing a type of file may be used in this context.
		X'10A8'	STOP (detailed data qualifier) *Note:* The qualifier identifies the node name.
		X'10A9'	FOLLOW PROBLEM HANDLING PROCEDURES FOR (detailed data qualifier) *Note:* The qualifier identifies the resource.
		X'10AA'	POWER OFF THEN POWER ON (detailed data qualifier) *Note:* The qualifier identifies the resource.
		X'10AB'	RESTORE (detailed data qualifier) LICENSED INTERNAL CODE *Note:* The qualifier identifies the resource.
		X'10AC'	SAVE (detailed data qualifier) *Note:* The qualifier identifies the entity to be saved.
		X'10AD'	RUN (detailed data qualifier) AND MIRROR THE DRIVE *Note:* The qualifier identifies a command or program.
		X'10AE'	RUN (detailed data qualifier) AND RECOVER DETACHED SECONDARY *Note:* The qualifier identifies a command or program.
		X'10AF'	RUN (detailed data qualifier) AND CORRECT ALL ERRORS *Note:* The qualifier identifies a command or program.
		X'10C0'	DELETE JOURNAL (detailed data qualifier) CONNECTION NAME (detailed data qualifier) *Note:* The first qualifier identifies the journal resource variable. The second qualifier identifies the connection name variable. A journal (as used here) is used to record information about device events. The connection is the connection between a logical device and an application.
		X'10C1'	INITIALIZE JOURNAL (detailed data qualifier) CONNECTION NAME (detailed data qualifier) *Note:* The first qualifier identifies the journal resource variable. The second qualifier identifies the connection name variable. A journal (as used here) is used to record information about device events. The connection is the connection between a logical device and an application.
		X'10C2'	RE-CREATE JOURNAL (detailed data qualifier) CONNECTION NAME (detailed data qualifier) *Note:* The first qualifier identifies the journal resource variable. The second qualifier identifies the connection name variable. A journal (as used here) is used to record information about device events. The connection is the connection between a logical device and an application.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content		
			X'10C3'	RUN (detailed data qualifier) ON FILE CONTAINED IN (detailed data qualifier) *Note:* The first qualifier identifies a command or program. The second qualifier identifies the disk block number.
			X'10E0'	STOP (sf83 product text)
		X'1100'	VARY OFFLINE	
		X'1200'	RETRY	
			X'1201'	MOVE THE PAGING DATA SETS TO ANOTHER SUBSYSTEM
			X'1202'	MOVE PACK TO ANOTHER DRIVE AND RERUN THE SAME JOB
			X'1203'	RESTART JOB
			X'1204'	ATTEMPT TO REESTABLISH THE CONNECTION
			X'1205'	RERUN THE APPLICATION PROGRAM
			X'1206'	WAIT THEN RETRY
			X'1207'	RETRY – DIAL NUMBER MANUALLY
			X'1208'	RETRY REQUEST
			X'12C0'	RETRY AFTER (detailed data qualifier) (detailed data qualifier) *Note:* The two qualifiers indicate a date and time after which the operation should be retried.
		X'1300'	CORRECT THEN RETRY: The operator should correct the condition referred to and retry the operation	
			X'1301'	READY THE DEVICE THEN RETRY
			X'1302'	RESET PRINT SPOOLER CONTROL FILE
			X'1303'	RESET VALUE WITHIN THE VALID RANGE
			X'1304'	CORRECT NODE INFORMATION BLOCK CONFIGURATION
			X'1305'	REACTIVATE B-CHANNEL RESOURCE
			X'1306'	RELEASE HELD JOBS THEN RETRY
			X'1310'	VERIFY THAT AIR VENTS ARE NOT COVERED
			X'1311'	CHECK FOR DIRTY FILTER
			X'1320'	CHECK CABLE CONNECTION AND RETRY
			X'1330'	ACTIVATE PORT THEN RETRY
			X'1331'	ENABLE LINE THEN RETRY
			X'1332'	REACTIVATE LINE
			X'1340'	CORRECT THEN RETRY OFFENDING SOFTWARE COMPONENT
			X'13A0'	ACTIVATE ONE OR MORE PORTS IN THE ROTARY GROUP ASSOCIATED WITH (detailed data qualifier) *Note:* The qualifier identifies the telephone number associated with the rotary group.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'13A1'	ACTIVATE RESOURCES ATTACHED TO (detailed data qualifier)
		X'13A2'	DEACTIVATE RESOURCES ATTACHED TO (detailed data qualifier)
		X'13A3'	REPEAT CALL WITH (detailed data qualifier) *Note:* The qualifier identifies the new number.
		X'13A4'	RUN (detailed data qualifier) AND VERIFY DRIVE *Note:* The qualifier identifies a command or program.
		X'13A5'	RUN (detailed data qualifier) AND VERIFY MIRRORED DRIVE(S) *Note:* The qualifier identifies a command or program.
		X'1400'	RESTART: Perform the appropriate restart operation on the indicated resource
		X'1401'	RE-IML MOSS: Reload the MOSS microcode
		X'1402'	RE-IPL THE COMMUNICATION CONTROLLER: Reload the system software program in the communication controller
		X'1403'	RE-IPL THE SECONDARY FINANCE CONTROLLER
		X'1404'	RE-IML THE CONTROL UNIT
		X'1405'	REACTIVATE LAN MANAGEMENT SERVER PROGRAM
		X'1406'	FOLLOW ALERT SENDERS PROCEDURES FOR RESOURCE ACTIVATION
		X'1407'	ACTIVATE THE SESSION
		X'1408'	RESTART DEVICE DATA MANAGEMENT SUPPORT
		X'1409'	RESTART DEVICE DATA MANAGEMENT SUPPORT SERVER
		X'140A'	RESTART DISTRIBUTION SERVER: Restart the server responsible for distributing information among nodes.
		X'140B'	RESTART PRINT SPOOLER
		X'140C'	RESTART PRINTER CONTROL BLOCK
		X'140D'	RESTART COMMUNICATION PROTOCOL PROGRAM
		X'140E'	RESTART DESTINATION SERVER
		X'140F'	RESTART PROGRAM
		X'1410'	RESUME OPERATION ON BACKUP PU: Automatic problem bypass has been successful, and a backup PU is now available; operation should be resumed using this PU
		X'1411'	RESTART COMMUNICATIONS SUBSYSTEM CONTROLLER
		X'1412'	RESTART RESOURCE
		X'1413'	RESUME DESTINATION RESOURCE Remove the destination resource from the paused state.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'1414'	RESUME RESOURCE Remove the resource from the paused state.
		X'1415'	START COMMUNICATION CONTROL BLOCK
		X'1416'	START DESTINATION RESOURCE
		X'1417'	START DEVICE DATA MANAGEMENT SUPPORT SERVER
		X'1418'	START DISK QUEUE SERVER
		X'1419'	START FILE SUPPORT
		X'141A'	START INBOUND SERVER
		X'141B'	START ONLY ONE COMMUNICATION CONTROL BLOCK AT A TIME
		X'141C'	START OUTBOUND SERVER
		X'141D'	POWER ON OR RESTART
		X'141E'	STOP / RESTART SERVER LOGON SERVICE
		X'141F'	RESTART SOFTWARE SUBSYSTEM
		X'1420'	RESTART SOFTWARE SUBTASK
		X'1421'	RESTART MESSAGE LOGGING
		X'1422'	RESTART SERVER
		X'14A0'	VARY OR CONNECT (detailed data qualifier) ONLINE: Start the identified element via local target system control program facilities. *Note:* The qualifier identifies the target element.
		X'14A1'	IML THE (detailed data qualifier): Initialize the central processor complex (CPC) or a resource within the CPC. *Note:* The qualifier identifies the IML target element.
		X'14A2'	ACTIVATE (detailed data qualifier): Issue the Activate command to attempt to bring the target system online. *Note:* The qualifier identifies the activation target.
		X'14A4'	START (detailed data qualifier) *Note:* The qualifier identifies the resource name.
		X'14D0'	IPL (detailed data qualifier) FROM (detailed data qualifier) WITH (detailed data qualifier): Load the system control program. *Note:* The first qualifier identifies the IPL target CPU. The second qualifier identifies the IPL device address and the third qualifier identifies an IPL parameter.
		X'14E0'	RESTART (sf83 product text)
		X'1500'	CORRECT INSTALLATION PROBLEM: It will be necessary to correct the installation error before continuing operation
		X'1501'	CORRECT GENERATION PROBLEM
		X'1502'	CORRECT CUSTOMIZATION PARAMETERS
		X'1503'	CORRECT CONFIGURATION
		X'1504'	APPLY CORRECT SOFTWARE LEVEL
		X'1505'	LOAD THE REQUIRED OPTIONAL MODULE
		X'1506'	INCREASE INTERVENTION TIMER VALUE
		X'1507'	CORRECT ADDRESS FROM MODEM KEYPAD

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'1508'	CORRECT ADDRESS FROM DSU/CSU CONTROL PANEL
		X'1509'	ENABLE LPDA-2 FROM MODEM KEYPAD
		X'150A'	ENABLE LPDA-2 FROM DSU/CSU CONTROL PANEL
		X'150B'	CONFIGURE MODEM
		X'150C'	CONFIGURE DSU/CSU
		X'150D'	CONFIGURE LOCAL MODEM AS PRIMARY OR CONTROL
		X'150E'	CONFIGURE LOCAL DSU/CSU AS PRIMARY OR CONTROL
		X'150F'	CHECK THRESHOLD LIMIT AND CHANGE IF SET TOO LOW
		X'1510'	CORRECT THE ADDRESS
		X'1511'	CHECK THRESHOLD LIMIT AND CHANGE IF SET TOO HIGH
		X'1512'	CHANGE CALL SETUP PARAMETERS
		X'1513'	UPDATE CLIENT APPLICATION PROGRAM TO PROVIDE PASSWORD ENCRYPTION
		X'1514'	CHANGE SERVER CONFIGURATION TO ACCEPT UNENCRYPTED PASSWORDS
		X'1515'	CHECK LOWER LIMIT AND CHANGE IF SET TOO HIGH
		X'1516'	INSTALL LATEST MICROCODE LEVEL
		X'1517'	APPLY LATEST SERVICE LEVEL
		X'1520'	INCREASE MAXIMUM NUMBER OF CONNECTIONS ALLOWED ON THIS NODE
		X'1521'	ADD DESTINATION RESOURCE TO RESOURCE DIRECTORY
		X'1522'	ADD ORIGINATOR RESOURCE TO SUBNET DIRECTORY
		X'1523'	ADD ORIGINATOR RESOURCE TO SYSTEM CONFIGURATION
		X'1524'	ADD RESOURCE TO TABLE AND QUEUES DEFINITION FILE Add an entry to the file containing descriptive information about software tables and various system queues.
		X'1525'	ADD DESTINATION RESOURCE TO CONTROL BLOCK FILE
		X'15A0'	ADD (detailed data qualifier) TO RESOURCE DIRECTORY *Note:* The qualifier identifies the resource.
		X'15A1'	ADD (detailed data qualifier) TO SUBNET CONFIGURATION *Note:* The qualifier identifies the node name.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content		
			X'15A2'	ADD (detailed data qualifier) TO SYSTEM CONFIGURATION *Note:* The qualifier identifies the node name.
		X'1600'	REPLACE MEDIA	
			X'1601'	FOR REMOVABLE MEDIA, CHANGE MEDIA AND RETRY
			X'1602'	PLACE BACKUP DISKETTE IN DRIVE
			X'1603'	CHANGE DISKETTE AND RETRY
			X'1604'	PUT CORRECT PAPER IN CASSETTE
			X'1605'	PUT CASSETTE IN PRINTER
			X'1606'	ADD PAPER
		X'1700'	REPLENISH SUPPLIES	
			X'1701'	REPLACE RIBBON
			X'1702'	ADD INK
			X'1703'	ADD TONER
			X'1704'	CHANGE ALL AIR FILTERS
			X'1705'	ADD FUSER OIL
			X'1706'	ADD STAPLES
		X'1800'	REPLACE DEFECTIVE EQUIPMENT	
			X'1801'	REPLACE KEYBOARD
			X'1802'	REPLACE MODULE
			X'1803'	REPLACE CARD
			X'1804'	REPLACE DEVICE
			X'1805'	REPLACE BATTERY
			X'1806'	REPLACE PRINTER
			X'1807'	REPLACE DISPLAY CONTROL MODULE
			X'1808'	REPLACE MSR OR MSRE: Replace the magnetic stripe reader or magnetic stripe reader/encoder
			X'1811'	REPLACE STORAGE CONTROLLER
			X'1812'	REPLACE WORKSTATION CONTROLLER
			X'1813'	REPLACE COMMUNICATIONS SUBSYSTEM CONTROLLER
			X'18A0'	REPLACE THE CARD IDENTIFIED BY (detailed data qualifier) *Note:* The qualifier identifies the card to be replaced, e.g., by its part number.
			X'18A1'	REPLACE CABLE IDENTIFIED BY (detailed data qualifier)

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
			X'18C0' REPLACE THE BATTERY IDENTIFIED BY (detailed data qualifier) (detailed data qualifier) *Note:* The two qualifiers identify the battery to be replaced, e.g., by giving its type and location.
		X'1900'	PERFORM PROBLEM BYPASS PROCEDURES: Refer to the problem bypass documentation provided for this condition and follow the specified procedures
			X'1901' REPLACE MODEM X'1902' REPLACE DSU/CSU X'1903' CHANGE TO BACKUP SPEED X'1904' ACTIVATE SNBU, IF AVAILABLE X'1905' DISCONNECT AND RE-DIAL SNBU LINE X'1906' USE ALTERNATE PORT OR LINE X'1907' BACKUP THE DRIVE AT SERVER X'19A0' QUIESCE AND MOVE THE (detailed data qualifier) WORKLOAD TO ANOTHER SYSTEM: Move work to another equivalent resource prior to imminent shutdown or until the resource is restored. *Note:* The qualifier identifies the system image name. X'19A1' PERFORM MANUAL FALLBACK TO (detailed data qualifier) *Note:* The qualifier identifies the communication control unit (CCU) within the communication controller to which the fallback is to be done. X'19A2' REMOVE (detailed data qualifier) FROM SERVICE AND RESUME OPERATIONS IF POSSIBLE *Note:* The qualifier identifies the resource. X'19A3' BACKUP, REFORMAT, AND RESTORE (detailed data qualifier) *Note:* The qualifier identifies the drive. X'19A4' BACKUP DATA BEFORE POWERING OFF (detailed data qualifier) *Note:* The qualifier identifies the device.
		X'1A00'	REMOVE MEDIA: The medium (usually paper, forms, or cards) supply needs to be removed and operator action is required to remove the medium in order to continue operation.
			X'1A01' REMOVE DOCUMENTS X'1A02' REMOVE CARDS
		X'1B00'	PREPARE: Preparations should be made to handle a particular situation.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
			X'1B11' PREPARE FOR SERVER SHUTDOWN
		X'2000'	(Review detailed data): Refer to the detailed data presentation for additional messages and information *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
			X'2001' (Report detailed data): Report the information that was transported in the Detailed Data subvector *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure. *Note:* An Alert receiver has the option of displaying the data from the Detailed Data (X'82') subvector either in conjunction with this text or in another display that can be reached from the display containing this text.
			X'2002' (Review most recent traffic statistics): Report the information in the statistics subvectors kept for the link stations *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
			X'2010' (Review link detailed data): Review the information that was transported in the X'5x' and/or X'8C' subvectors flowing in this Alert. *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
			X'2011' (Review hexadecimal display of the Alert record): Review the screens providing a hexadecimal display of the entire Alert record *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
			X'2012' (Review associated resources): Review the information that was transported in the Associated Resources (X'11') subfield of the Hierarchy/Resource List subvector in this Alert *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
			X'2013' (Review Supporting Data Correlation): Review the information that was transported in the Supporting Data Correlation (X'48') subvector in this Alert *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
		X'2100'	(Review recent Alerts for this resource) *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
			X'2101' (Review recent statistical records for this resource) *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
			X'2102' (Review correlated Alerts) *Note:* There is no text string defined for this code point; the Alert receiver indicates the action to be taken in terms of its own screen design and command structure.
		X'2200'	REVIEW DATA LOGS: Review the specified records in one or more data logs
			X'2201' REVIEW REMOTE DEVICE LOGS
			X'2202' REVIEW DEVICE STATISTICAL LOG AT ALERT SENDER
			X'2203' REVIEW SUPPORTING DATA AT ALERT SENDER
			X'2204' REVIEW NETWORK LOG AT ALERT FORWARDER
			X'22C0' REVIEW SENDING DEVICE LOG — (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier is the log identification and the second qualifier is the data to be reviewed (i.e., System Message Log).
			X'22C1' REVIEW PARTNER'S RECOVERY LOG — (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers further define the log data to be reviewed. For example the first qualifier could be the log identification and the second the log record number.
		X'3000'	CONTACT APPROPRIATE SERVICE REPRESENTATIVE: This Alert condition has been caused by a hardware or software failure. The operator is directed to contact the person, organization, or vendor responsible to provide service for this product.
			X'3001' CONTACT CONSUMER SERVICE REPRESENTATIVE: Contact the customer representative who is responsible for dealing with consumer users of the device.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content		
			X'3002'	CONTACT SECURITY CONTROL REPRESENTATIVE: Contact the customer representative who is responsible for dealing with security concerns for the device.
			X'3003'	CONTACT SERVICE REPRESENTATIVE FOR LIST OF REMAINING PART NUMBERS
			X'30A0'	DIAL (detailed data qualifier) AND REPORT THE MACHINE INFORMATION: Hardware remote support was unable to contact the service organization. *Note:* The qualifier identifies a telephone number to call for hardware service.
			X'30A1'	CALL (detailed data qualifier) FROM DESIGNATED PHONE: *Note:* The qualifier identifies a service resource to be called and may or may not be a phone number. If the qualifier is not a phone number, installation-specific procedures on contacting the indicated service resource should be followed. If the qualifier is a phone number, the need to perform the manual dial may be a recovery procedure in lieu of an automatic dial function.
			X'30E0'	PROVIDE REMOTE SERVICE CALL AUTHORIZATION FOR (sf83 product text): Hardware remote support requires authorization to proceed with the automatic call for service.
			X'30E1'	CONTACT SERVICE REPRESENTATIVE FOR (sf83 product text)
		X'3100'	CONTACT ADMINISTRATIVE PERSONNEL: Contact personnel with administrative responsibility for one or more network resources	
			X'3101'	CONTACT TOKEN-RING ADMINISTRATOR RESPONSIBLE FOR THIS LAN
			X'3102'	CONTACT CSMA/CD ADMINISTRATOR RESPONSIBLE FOR THIS LAN
			X'3103'	CONTACT LAN ADMINISTRATOR RESPONSIBLE FOR THIS LAN
			X'3104'	CONTACT NETWORK INFORMATION SERVICE FOR PRIVATE NETWORK CALLED
			X'3105'	CONTACT X.21 NETWORK INFORMATION SERVICE
			X'3106'	CONTACT ISDN NETWORK INFORMATION SERVICE
			X'3107'	CONTACT X.25 NETWORK INFORMATION SERVICE
			X'3108'	CONTACT RELATIONAL DATABASE ADMINISTRATOR
			X'3109'	CONTACT PERSONNEL RESPONSIBLE FOR CONNECTION TO ISDN NETWORK
			X'310A'	CONTACT SERVER ADMINISTRATOR
			X'310B'	CONTACT NETWORK ADMINISTRATOR

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'310C'	CONTACT ADMINISTRATOR OF AFFECTED RESOURCES
		X'310E'	CONTACT DS1 NETWORK INFORMATION SERVICE
		X'310F'	CONTACT PERSONNEL RESPONSIBLE FOR CONNECTION TO DS1 NETWORK
		X'3110'	CONTACT COMMUNICATIONS SYSTEMS PROGRAMMER
		X'3111'	CONTACT OPERATOR RESPONSIBLE FOR ALERT SENDER
		X'3112'	CONTACT SYSTEMS PROGRAMMER
		X'3113'	CONTACT PERSONNEL RESPONSIBLE FOR DASD SUPPORT
		X'3120'	CONTACT PRINTER OPERATOR
		X'3121'	CONTACT TERMINAL CONTROL UNIT OPERATOR
		X'3122'	CONTACT CALLED DTE'S OPERATOR
		X'3123'	CONTACT REMOTE DTE'S OPERATOR
		X'3124'	CONTACT PBM NETWORK OPERATOR: Contact the operator who has specific responsibility for controlling the personal banking machine (PBM) network for the reporting device.
		X'3125'	CONTACT REMOTE LINK STATION OPERATOR
		X'3126'	CONTACT CALLING DTE'S OPERATOR
		X'31D0'	IF REQUIRED, QUERY (detailed data qualifier) AT (detailed data qualifier) ABOUT (detailed data qualifier) *Note:* The first qualifier identifies the contact id. The second qualifier identifies the location name and the third qualifier identifies the system name.
		X'31E0'	CONTACT SYSTEMS ADMINISTRATOR FOR (sf83 product text)
		X'3200'	REPORT THE FOLLOWING *Note:* Since replacement code points for reporting one, two, and three (detailed data qualifier)'s are all required, the X'32xx' code points violate the usual rule of defining only one replacement code point, in the range indicating three qualifiers. Three separate replacement code points are defined, and should be used by Alert senders, depending on the number of qualifiers to be passed.
		X'32A0'	REPORT THE FOLLOWING (detailed data qualifier)
		X'32C0'	REPORT THE FOLLOWING (detailed data qualifier) (detailed data qualifier)
		X'32D0'	REPORT THE FOLLOWING (detailed data qualifier) (detailed data qualifier) (detailed data qualifier)

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
			X'32D1' REPORT THE FOLLOWING LOGICAL UNIT OF WORK IDENTIFIER (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The first qualifier identifies the network-qualified LU name. The second qualifier identifies the instance and sequence number and the third qualifier may be used for possible extensions to the logical unit of work identifier.
		X'3300'	IF PROBLEM REOCCURS THEN DO THE FOLLOWING: After performing the previous actions, try the operation again. If you experience another problem, then perform the following actions
			X'3301' IF PROBLEM PERSISTS THEN DO THE FOLLOWING X'3302' IF PROBLEM CONTINUES TO OCCUR REPEATEDLY THEN DO THE FOLLOWING X'3303' IF UNSUCCESSFUL THEN DO THE FOLLOWING
		X'3400'	WAIT FOR ADDITIONAL MESSAGE BEFORE TAKING ACTION: An additional message will be forthcoming, indicating the action to be taken
			X'3401' EXPECT A CALL FROM THE SERVICE ORGANIZATION TO PROVIDE AN ESTIMATED TIME OF ARRIVAL
		X'3500'	REFER TO PRODUCT DOCUMENTATION FOR ADDITIONAL INFORMATION
			X'3502' REFER TO INDUSTRIAL COMPUTER EXTENSIONS LIBRARY FOR ADDITIONAL INFORMATION
		X'3501'	REFER TO MODEM DOCUMENTATION FOR ADDITIONAL INFORMATION
			X'35E0' REFER TO (sf83 product text) PRODUCT DOCUMENTATION FOR ADDITIONAL INFORMATION
		X'E000'–X'EFFF'	Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range. *Note:* The following code points specify extended messages. An Alert receiver that displays only default text provides no display for these code points.
		X'F000'	(no display): Additional message data
			X'F001' MULTIPLE FAILURES INDICATE CHANNEL FAILURE X'F002' MULTIPLE FAILURES INDICATE CONTROLLER FAILURE

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'F003'	MULTIPLE DRIVE FAILURES ON SAME CONTROLLER INDICATE CONTROLLER FAILURE
		X'F004'	MULTIPLE FAILURES INDICATE CONTROL UNIT FAILURE
		X'F005'	MULTIPLE FAILURES INDICATE TERMINAL MULTIPLEXER FAILURE
		X'F006'	REOCCURRENCE OF SAME ERROR INDICATES MEDIA FAILURE
		X'F007'	REOCCURRENCE OF PROBLEM INDICATES DEVICE OR ATTACHMENT ERROR
		X'F008'	REOCCURRENCE INDICATES MEDIA PROBLEM
		X'F009'	NON REOCCURRENCE OF FAILURE INDICATES ORIGINAL DRIVE FAILURE
		X'F00A'	MULTIPLE FAILURES INDICATE LINE ADAPTER MULTIPLEXER FAILURE
		X'F00B'	DO ONE OF THE FOLLOWING: *Note:* This code point is to precede two or more recommended actions. It and the actions following it should be the last actions for a particular cause.
		X'F00C'	NOTIFY USER
		X'F00D'	HIGH ERROR RATE BETWEEN THE FOLLOWING NODES
		X'F00E'	LINK FAILURE BETWEEN THE FOLLOWING NODES
		X'F00F'	LINK CONNECTED TO THE FOLLOWING NODE FAILED
		X'F010'	OTHER NODE IN THE LINK IS ONE OF THE FOLLOWING NODES
		X'F011'	NO FURTHER ACTION REQUIRED UNLESS PROBLEM PERSISTS
		X'F012'	THIS ALERT IDENTIFIES THE CAUSE OF A PREVIOUS ERROR WHICH HAS BEEN RECOVERED
		X'F013'	SERVICE CAN BE SCHEDULED AT A LATER TIME UNLESS REPEATED FAILURES PREVENT NORMAL OPERATION
		X'F014'	RESUME OPERATION
		X'F015'	HEURISTIC DAMAGE
		X'F016'	COORDINATOR COMMITTED
		X'F017'	COORDINATOR BACKED OUT
		X'F018'	AGENT COMMITTED
		X'F019'	AGENT BACKED OUT
		X'F01A'	AGENT IN-DOUBT
		X'F01B'	DASD HDA POWER SEQUENCE FAULT
		X'F01C'	ADDITIONAL ANALYSIS (BY SERVICE PERSONNEL) IS REQUIRED TO DETERMINE REPAIR IMPACT
		X'F01D'	REPAIRING REDUNDANT EQUIPMENT WILL NOT CAUSE A LOSS OF FUNCTION

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'F01E'	REPAIR WILL DISABLE PROTOCOL CONVERTER
		X'F01F'	EXCEPTION ON PROTOCOL CONVERTER
		X'F020'	AUTOMATIC ADAPTER (RE)INITIALIZATION ATTEMPTED
		X'F021'	RESOURCES HAVE BEEN LOST, BUT THEY COULD NOT BE SPECIFICALLY IDENTIFIED
		X'F022'	FORMAT ERROR
		X'F023'	PROTOCOL ERROR
		X'F04F'	NO IML CAPABILITIES
		X'F050'	IPL CAPABILITIES LIMITED
		X'F051'	NO IPL CAPABILITIES
		X'F052'	NORMAL OPERATIONS CAN CONTINUE BUT IF AUXILIARY STORAGE IS EXHAUSTED ON-SITE ACTION WILL BE NECESSARY
		X'F060'	TO RECOVER LOST RESOURCE
		X'F0A0'	FOR (detailed data qualifier)
		X'F0A1'	(detailed data qualifier) HAS BEEN RESTORED FROM BACKUP FILE *Note:* The qualifier identifies the restored file. Any data id describing a type of file may be used in this context.
		X'F0A2'	NO BACKUP AVAILABLE FOR (detailed data qualifier) *Note:* The qualifier identifies the file. Any data id describing a type of file may be used in this context.
		X'F0A3'	NODE (detailed data qualifier) *Note:* The qualifier identifies an attribute of the node, for example the node's name or number.
		X'F0A4'	COORDINATOR (detailed data qualifier) *Note:* The qualifier identifies the coordinator. For example the qualifier could be the relational database name, the logical unit name, etc.
		X'F0A5'	(detailed data qualifier) UNSUCCESSFUL ACCESS ATTEMPTS *Note:* The qualifier specifies a number.
		X'F0A7'	(detailed data qualifier) DEGRADED
		X'F0A8'	(detailed data qualifier) DISABLED
		X'F0A9'	(detailed data qualifier) ENABLED
		X'F0C0'	TRANSMISSION PRIORITY FOR SUBNET (detailed data qualifier) IS (detailed data qualifier) *Note:* The qualifiers identify the subnet and the transmission priority field setting for that subnet, respectively.
		X'F0C1'	DASD CACHE EXCEPTION — (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers specify information related to the failing cache, for example the subsystem id and storage path.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'F0D0'	FAILING COMPONENT LOCATION (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the failing component location in one of two ways: Method 1: Q1 = RACK Q2 = UNIT (within a rack) Q3 = CARD SLOT (within a unit) Method 2: Q1 = RACK/UNIT (with no delimiter between the rack and unit numbers) Q2 = CARD SLOT (within a unit) Q3 = CABLE POSITION (on a card) Method 2 is used only in those cases where cable position on a card is meaningful.
		X'F0D1'	ORIGIN NODE DATA: (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the originator's netid, location name, and class of service, respectively.
		X'F0D2'	DESTINATION NODE DATA: (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the destination netid, location name, and class of service, respectively.
		X'F0D3'	NODE DATA: (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the interface, service action code, and problem data, respectively.
		X'F0D4'	MEDIA EXCEPTION ON (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the entity which had the media exception. For example they could specify the subsystem id, volume, and device address.
		X'F0D5'	ADDITIONAL DATA — (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers provide additional information about or related to the problem being reported. For example, the qualifiers could specify the device address, cylinder, and head for a DASD error.
		X'F0D6'	DASD CONTROL UNIT EXCEPTION ON (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the control unit. For example they could specify the subsystem id, storage cluster, and device path.

Recommended Actions (X'81') Network Alert Common Subfield

Byte	Bit	Content	
		X'F0D7'	REPAIR WILL DISABLE (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the entities which would be disabled by repairs. For example they could specify the subsystem id, storage cluster, and device.
		X'F0D8'	DEVICE EXCEPTION ON (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify the device. For example they could specify the subsystem id, volume, and device address.
		X'F0D9'	AFFECTED RESOURCE IS IDENTIFIED BY (detailed data qualifier) (detailed data qualifier) (detailed data qualifier) *Note:* The qualifiers identify an affected resource.
		X'F0DA'	AND (detailed data qualifier) (detailed data qualifier) (detailed data qualifier)
		X'F0E0'	FOR (sf83 product text)
		X'F0E1'	PREPARE FOR AUTOMATIC SHUTDOWN OF (sf83 product text)
		X'F0E2'	NODE (sf83 product text)
		X'F0F0'	FOR (sf84 resource name)
		X'F0F1'	RESOURCE ASSOCIATED WITH THE ALERT CONDITION (sf84 resource name)
		X'FFFF'	Reserved

Detailed Data (X'82') Network Alert Common Subfield

This subfield contains product specific detailed data to be displayed at an Alert receiver. Since it provides additional function, the Detailed Data Extended (X'85') subfield should be used instead of the Detailed Data (X'82') subfield when possible.

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Detailed Data subfield *Note:* Length = X'02' indicates that the Product ID Code, Data ID, Data Encoding, and Detailed Data fields are not present.
1		Key: X'82'

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content
2		Product ID code: a code indicating what product identification, if any, must be displayed in conjunction with the data type and data. The structure of this field is identical to that present in the Product Set ID Index (X'83') subfield. A value of X'00' in this byte indicates that no product identification data is displayed in conjunction with the data type and detailed data.
	0–3	Product ID subvector code: a code point that specifies (1) the type of Product ID subvector being indexed (hardware or software), and (2) the particular data to be extracted from this subvector *Note:* See "Product Identifier (X'11') MS Common Subvector" on page 9-355 for the criteria distinguishing hardware and software Product ID subvectors. X'0'–X'1' reserved X'2' (machine type or hardware product common name) from a hardware Product ID Subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'5' (machine type or hardware product common name) plus model number from a hardware Product ID Subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'9' software product common name from a software Product ID subvector
	4	Product set ID indicator: an indication of which Product Set ID (PSID) contains the Product ID subvector being indexed 0 Alert sender PSID 1 Indicated resource PSID
	5–7	Count: a 3-digit binary number that indicates which Product ID subvector, of the type specified by the Product ID subvector code, is being indexed within the PSID specified by the Product Set ID Indicator. *Note:* This count applies only to Product ID subvectors of the type specified by the Product ID subvector code. If, for example, the code is X'2' (specifying a hardware Product ID) then only hardware X'11's are counted: a count of X'3' would thus index the third hardware Product ID within the PSID indicated by the Product Set ID Indicator.
3		Data ID: a code point indicating the type of data carried in the subfield. The English text associated with each code point, or its national language equivalent, is displayed in conjunction with the detailed data. Defined codes are: X'00' (no display) X'01' ABEND CODE X'02' ADAPTER CHECK STATUS X'03' ADAPTER RETURN CODE X'04' BOP CODE X'05' PROTOCOL CODE X'06' COMMAND CODE X'07' ERROR CODE X'08' OPERATING SYSTEM RETURN CODE X'09' EVENT CODE X'0A' LLC ERROR CODE

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'0B'	MACHINE CHECK CODE
		X'0C'	MALFUNCTION CODE
		X'0D'	PROGRAM CHECK CODE
		X'0E'	REASON CODE
		X'0F'	RETURN CODE
		X'10'	SENSE CODE
		X'11'	SENSE DATA
		X'12'	SOFTWARE ERROR CODE
		X'13'	STATUS CODE
		X'14'	SYMPTOM CODE
		X'15'	SNA SENSE DATA
		X'16'	BUS STATUS CODE
		X'17'	RING STATUS CODE
		X'18'	CALL PROGRESS SIGNAL: A notification from a network to a DTE, indicating why a connection could not be established
		X'19'	FILE
		X'1A'	X.25 CLEAR PACKET, CAUSE CODE: A code to or from an X.25 network indicating the reason that a CLEAR request or indication packet was sent *Note:* This indicates the reason that a network connection was lost or could not be established.
		X'1B'	X.25 RESET PACKET, CAUSE CODE: A code to or from an X.25 network indicating the reason that a RESET request or indication packet was sent *Note:* This indicates the reason that a network connection was lost or could not be established.
		X'1C'	X.25 RESTART PACKET, CAUSE CODE: A code to or from an X.25 network indicating the reason that a RESTART request or indication packet was sent *Note:* This indicates the reason that a network connection was lost or could not be established.
		X'1D'	X.25 DIAGNOSTIC CODE: A code to or from an X.25 network providing additional information about why a Diagnostic packet or a Clear, Reset, or Restart request or indication packet was sent.
		X'1E'	DIAGNOSTIC EXPLANATION
		X'1F'	DATABASE RETURN CODE
		X'20'	MESSAGE CODE
		X'21'	PANEL ERROR MESSAGE CODE
		X'22'	SYSTEM MESSAGE CODE
		X'23'	MESSAGE SEVERITY
		X'24'	WAIT STATE CODE
		X'25'	PRIMARY RETURN CODE
		X'26'	SECONDARY RETURN CODE
		X'27'	FUNCTION
		X'28'	SOCKET NUMBER: The TCP/IP endpoint for the communication connection
		X'29'	DRIVE

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'2A'	V.25BIS INDICATION
		X'2B'	ICA SENSE CODE
		X'2C'	MAINTENANCE INDICATOR
		X'2D'	ACCESS UNIT ID
		X'2E'	CONTROLLED ACCESS UNIT LOBE NUMBER
		X'2F'	D-CHANNEL NUMBER
		X'30'	REFERENCE CODE
		X'31'	SYSTEM REFERENCE CODE
		X'32'	REPLACEABLE UNIT CODE
		X'33'	COMPONENT ID
		X'34'	COMMUNICATION CONTROL UNIT
		X'35'	TYPE
		X'36'	LOCATION
		X'37'	PART NUMBER
		X'38'	DESTINATION
		X'39'	ORIGINATOR
		X'3A'	RACK/UNIT
		X'3B'	RACK
		X'3C'	UNIT
		X'3D'	CARD SLOT
		X'3E'	CABLE POSITION
		X'3F'	INTERCONNECT CONTROLLER
		X'40'	ERROR RECOVERY PROCEDURE
		X'41'	PDP CODE
		X'42'	BACKUP PATH STATUS
		X'43'	WRAP STATUS
		X'44'	ATTACHMENT MODULE STATUS
		X'45'	CONFIGURATION DATA
		X'46'	LOBES DEACTIVATED
		X'47'	ATTACHMENT MODULE NUMBER
		X'48'	NUMBER OF LINKS LOST
		X'49'	NUMBER OF NETBIOS SESSIONS LOST
		X'4A'	NETID
		X'4B'	LOCATION NAME
		X'4C'	CLASS OF SERVICE
		X'4D'	BACK-UP D-CHANNEL NUMBER
		X'4E'	ROUTE AFFECTED
		X'4F'	ACTIVE ROUTE
		X'50'	CHANNEL UNIT ADDRESS
		X'51'	DEVICE ADDRESS
		X'52'	LINE ADDRESS
		X'53'	LINE ADDRESS RANGE
		X'54'	ADAPTER AT ADDRESS
		X'55'	LINE
		X'56'	DTE ADDRESS CALLED
		X'57'	DTE ADDRESS CALLING

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'58'	LOCAL DTE ADDRESS
		X'59'	CHANNEL
		X'5A'	SERVER
		X'5B'	CARD NUMBER
		X'5C'	CARD NAME
		X'5D'	CARD LOCATION
		X'5E'	LAN BRIDGE ID
		X'5F'	TOKEN-RING ID
		X'60'	PORT NUMBER
		X'61'	ADAPTER NUMBER
		X'62'	CHANNEL ADAPTER NUMBER
		X'63'	LINE ADAPTER NUMBER
		X'64'	LINE INTERFACE COUPLER (LIC) POSITION
		X'65'	BUS NUMBER
		X'66'	TOKEN RING INTERFACE COUPLER NUMBER
		X'67'	LOCALLY-INITIATED LOGICAL CHANNEL
		X'68'	REMOTELY-INITIATED LOGICAL CHANNEL
		X'69'	CONTROLLER ID
		X'6A'	DEVICE ID
		X'6B'	DEVICE
		X'6C'	TRANSMISSION PRIORITY FIELD
		X'6D'	PROGRAM STATUS WORD
		X'6E'	DOMAIN CONTROLLER
		X'6F'	PEER SERVER
		X'70'	GENERATION PARAMETER
		X'71'	THRESHOLD PARAMETER
		X'72'	CONFIGURATION OBJECT/RECORD: Identifies the configuration object or record which contains one or more user settable parameters.
		X'73'	CONFIGURATION PARAMETER
		X'74'	IPL PARAMETER
		X'75'	PARAMETER VALUE
		X'76'	SECURITY DATABASE FILE
		X'77'	REQUESTER
		X'78'	ACCOUNT SYSTEM FILE
		X'79'	SUBSTITUTE PART NUMBER
		X'7A'	CENTRAL PROCESSOR COMPLEX
		X'7B'	CENTRAL PROCESSING UNIT: The CPU includes its associated vector element processor.
		X'7C'	LOGICAL PARTITION NAME
		X'7D'	SUBCHANNEL NUMBER
		X'7E'	CHANNEL PATH ID
		X'7F'	I/O PROCESSING ELEMENT ID
		X'80'	NODE
		X'81'	LINK STATION
		X'82'	CP
		X'83'	PU

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'84'	LU
		X'85'	TRANSACTION PROGRAM
		X'86'	LSL: Link Segment Level of a multi-segment link connection *Note:* In a multi-segment link connection, link segments are numbered in ascending order, from the error notification sender outwards; thus the link segment immediately adjacent to the error notification sender is Link Segment Level 1, the one adjacent to it is Link Segment Level 2, etc.
		X'87'	RELATIONAL DATABASE
		X'88'	STORAGE
		X'89'	FULLY QUALIFIED PCID
		X'8A'	PACKAGE
		X'8B'	TABLE
		X'8C'	VOLUME
		X'8D'	DIRECTORY
		X'8E'	LOGICAL PARTITIONS
		X'8F'	VECTOR PROCESSOR
		X'90'	YEAR/MONTH/DAY
		X'91'	TIME
		X'92'	JULIAN DATE
		X'93'	MINUTES
		X'94'	ACCOUNT
		X'95'	FILE SERVER
		X'96'	DATA STREAM
		X'97'	DATA STREAM OFFSET
		X'98'	USER
		X'99'	CONNECTION
		X'9A'	NETWORK
		X'9B'	SERVICE PROCESSOR
		X'9C'	EXPANDED STORAGE
		X'9D'	CENTRAL PROCESSOR COMPLEX IMAGE
		X'9E'	SERVICE UPDATE
		X'9F'	CUSTOMER PROBLEM REPORT
		X'A0'	BYTE OFFSET
		X'A1'	BIT OFFSET
		X'A2'	DETECTING MODULE
		X'A3'	FAILING MODULE
		X'A4'	MAINTENANCE LEVEL
		X'A5'	COMMAND
		X'A6'	PROGRAM
		X'A7'	RESOURCE
		X'A8'	MAXIMUM NUMBER OF NODE TABLE ENTRIES
		X'A9'	MODE NAME
		X'AA'	UNBIND TYPE
		X'AB'	INBOUND CALL
		X'AC'	SYSTEM ERROR
		X'AD'	REMOTE SUPPORT FACILITY

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content
		X'AE' CENTRAL OPERATIONS
		X'AF' CONTROL PROGRAM
		X'B0' EIA STANDARD
		X'B1' CCITT STANDARD
		X'B2' CHANNEL SUBSYSTEM
		X'B3' MESSAGE LOG
		X'B4' OPERATING SYSTEM LOG
		X'B5' DUMP
		X'B6' RETRY COUNT
		X'B7' TIMER SETTING
		X'B8' LINE SPEED (BITS PER SECOND)
		X'B9' LINE SPEED (KILOBITS PER SECOND)
		X'BA' LINE SPEED (MEGABITS PER SECOND)
		X'BB' OPERATION NUMBER
		X'BC' OPERATION PRIORITY
		X'BD' OPERATION INPUT ARRIVAL TIME
		X'BF' APPLICATION INPUT ARRIVAL TIME
		X'C0' COUNTER
		X'C1' ALARM SIGNAL
		X'C2' JOB NUMBER
		X'C3' ISDN CAUSE CODE
		X'C4' CONTROLLER DESCRIPTION OBJECT
		X'C5' COMMUNICATION LINE DESCRIPTION OBJECT
		X'C6' LOGICAL DEVICE DESCRIPTION OBJECT
		X'C7' NETWORK INTERFACE DESCRIPTION OBJECT
		X'C8' NETWORK INTERFACE NAME
		X'C9' NAME
		X'CA' JOB NAME
		X'CB' LINE NAME
		X'CC' MESSAGE FILE NAME
		X'CD' MESSAGE QUEUE NAME
		X'CE' MESSAGE REFERENCE KEY
		X'CF' REMOTE LOCATION NAME
		X'D0' FILE NAME
		X'D1' LOG RECORD NUMBER
		X'D2' CARTRIDGE: A component that holds items to be dispensed
		X'D3' AIR FILTER NUMBER
		X'D4' TELEPHONE NUMBER
		X'D5' CALLING TELEPHONE NUMBER
		X'D6' TELEPHONE NUMBER CALLED
		X'D7' REPORTING TELEPHONE NUMBER: The telephone number of the Alert sender
		X'D8' TIMER
		X'D9' LOG RECORD TYPE
		X'DA' LOG ID
		X'DB' PUBLICATION NUMBER

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content
		X'DC' NEW NUMBER X'DD' APPLICATION NAME X'DE' NUMBER X'DF' WORKSTATION ID X'E0'–X'EF' reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range. X'F0' PRODUCT ALERT REFERENCE CODE: A code that identifies an Alert in a user-friendly way. The product Alert reference code is used to index documentation provided by the Alert sending product. This documentation can group the Alerts into natural categories and provide extended explanations or diagnostic information. X'F1' QUEUE X'F2' INTERFACE X'F3' SERVICE ACTION CODE X'F4' PROBLEM MANAGEMENT HARDWARE NUMBER X'F5' PROBLEM DATA X'F6' DISK BLOCK NUMBER X'F7' CONTROLLER BUS ADAPTER X'F8' ENTERPRISE X'F9' CONTACT ID X'FA' SNMP GENERIC-TRAP NUMBER X'FB' SNMP SPECIFIC-TRAP NUMBER X'FC' SNMP MIB VARIABLE NAME X'FD' SNMP MIB VARIABLE VALUE X'FE' INTERNET PROTOCOL ADDRESS X'FF' SERVICE POINT APPLICATION
4		Data Encoding: a code point indicating how the accompanying detailed data is encoded, and, thus, how it is to be displayed. Defined code are: X'00' hexadecimal: The data is to be displayed as hexadecimal digits. X'01' binary: The data is the binary representation of an unsigned integer value (8, 16, 24, or 32 bits in length). The decimal equivalent is to be displayed. For example, the value B'11111111' (X'FF') is to be displayed as 255. X'10' hexadecimal code point: The data is a two-byte hexadecimal code point which is used to index a text string stored at the Alert receiver. The text string will be displayed in place of the hexadecimal code point that was transported in the Alert.

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content
		X'11' Coded Graphic Character Set 00640−00500 plus: The data is to be decoded using Coded Graphic Character Set 00640−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," plus three additional code points: X'5B' = "$" (dollar sign) X'7B' = "#" (number sign) X'7C' = "@" (at sign) *Note:* Detailed data encoded in this way is limited to codes, numbers, or internationally recognized terms that do not require translation.
5−q		Detailed data, encoded as specified when byte offset 4 = X'00', X'01', or X'11'.
Or		
5−6		Detailed data, encoded as specified when byte offset 4 = X'10' (hexadecimal code point). The hexadecimal code point has a length of two bytes, and provides an index to predefined text that is displayed at the Alert receiver. An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent. The method chosen must be consistent with the language of the code point text that "imbeds" the X'82' subfield text. These code points are logically grouped, one group per Data ID code point. Within a group, a code point may have only one text string associated with it. But, the same code point may have different text strings associated with it in other groups. The Data ID determines the "table" that is searched by the Alert receiver in order to find the appropriate text string. **Note:** The X'E000'−X'EFFF' range of code points in each table is reserved. Neither IBM products nor non-IBM products may assign a code point in this range. Specific defined codes and the corresponding displayed text are listed below for each group. **Group: Counter Names (Data ID = X'C0')** The code point assignments below are valid when used with the Data ID value of X'C0' (Counter). Defined hexadecimal codes are: X'0001' LOSS OF FRAME ALIGNMENT X'0002' DETECTED ACCESS TRANSMISSION SYSTEM ERRORS (DTSE) IN X'0003' DETECTED ACCESS TRANSMISSION SYSTEM ERRORS (DTSE) OUT X'0004' TE-DETECTED CODE VIOLATIONS X'0005' NT-DETECTED CODE VIOLATIONS X'0006' CRC ERRORS RECEIVED X'0007' CRC ERRORS TRANSMITTED X'0008' ERRORED SECONDS X'0009' SEVERELY ERRORED SECONDS

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'000A'	COLLISION DETECT
		X'000B'	SHORT FRAMES RECEIVED
		X'000C'	TRANSMIT OVERRUN
		X'000D'	RECEIVE UNDERRUN
		X'000E'	TA ASYNCHRONOUS ERRORS
		X'000F'	TA SYNCHRONOUS ERRORS
		X'0010'	ABORTED FRAMES RECEIVED
		X'0011'	ABORTED FRAMES TRANSMITTED
		X'0012'	MISADDRESSED FRAMES RECEIVED
		X'0013'	UNBOUNDED FRAMES RECEIVED
		X'0014'	NON-INTEGRAL FRAMES RECEIVED
		X'0015'	TOTAL FRAMES TRANSMITTED
		X'0016'	TOTAL FRAMES RECEIVED
		X'0017'	PDUS RETRANSMITTED
		X'0018'	RECEIVE SEQUENCE ERRORS
		X'0019'	TOTAL BYTES TRANSMITTED
		X'001A'	TOTAL BYTES RECEIVED
		X'001B'	TOTAL BYTES RETRANSMITTED
		X'001C'	TOTAL INCOMING CALLS
		X'001D'	TOTAL OUTGOING CALLS
		X'001E'	INCOMING CALLS REJECTED
		X'001F'	OUTGOING CALLS REJECTED
		X'0020'	INFORMATION FRAMES TRANSMITTED
		X'0021'	INFORMATION FRAMES RECEIVED
		X'0022'	PDUS DISCARDED
		X'0023'	TOTAL CONNECTIONS
		X'0024'	UNNUMBERED INFORMATION FRAMES TRANSMITTED
		X'0025'	UNNUMBERED INFORMATION FRAMES RECEIVED
		X'0026'	LAN TYPE 3 FRAMES TRANSMITTED
		X'0027'	LAN TYPE 3 FRAMES RECEIVED
		X'0028'	LAN TYPE 3 FRAMES RETRANSMITTED
		X'0029'	LAN TYPE 2 ACKNOWLEDGMENT TIMER TIMEOUTS
		X'002A'	LOCAL BUSY OCCURRENCES
		X'002B'	TOKEN RING MAC LINE ERRORS
		X'002C'	TOKEN RING MAC BURST ERRORS
		X'002D'	TOKEN RING MAC A/C ERRORS
		X'002E'	TOKEN RING MAC INTERNAL ERRORS
		X'002F'	TOKEN RING MAC LOST FRAME ERRORS
		X'0030'	TOKEN RING MAC RECEIVE CONGESTION ERRORS
		X'0031'	TOKEN RING MAC FRAME-COPIED ERRORS
		X'0032'	TOKEN RING MAC TOKEN ERRORS
		X'0033'	TOKEN RING MAC FREQUENCY ERRORS
		X'0034'	UNRECOGNIZED PDUS
		X'0035'	TEST COMMANDS RECEIVED
		X'0036'	TEST RESPONSES TRANSMITTED
		X'0037'	TIMER (MILLISECONDS)

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'0038'	FRAME-LOSS SECONDS NEAR-END
		X'0039'	FRAME-LOSS SECONDS FAR-END
		X'003A'	RESTART MESSAGES RECEIVED
		X'003B'	STATUS ENQUIRY MESSAGES RECEIVED
		X'003C'	ERRORED SECONDS NEAR-END
		X'003D'	ERRORED SECONDS FAR-END
		X'003E'	BURSTY ERRORED SECONDS NEAR-END
		X'003F'	BURSTY ERRORED SECONDS FAR-END
		X'0040'	SEVERELY ERRORED SECONDS NEAR-END
		X'0041'	SEVERELY ERRORED SECONDS FAR-END
		X'0042'	SLIP SECONDS NEAR-END
		X'0043'	SLIP SECONDS FAR-END
		X'0044'	BADLY FORMED FRAMES
		X'0045'	TOTAL POLLS TRANSMITTED
		X'0046'	TOTAL POLLS RECEIVED
		X'0047'	TOTAL UNNUMBERED FRAMES TRANSMITTED
		X'0048'	TOTAL UNNUMBERED FRAMES RECEIVED
		X'0049'	TOTAL SUPERVISORY FRAMES TRANSMITTED
		X'004A'	TOTAL SUPERVISORY FRAMES RECEIVED
		X'004B'	CLASS 0 CAUSE CODE
		X'004C'	CLASS 1 CAUSE CODE
		X'004D'	CLASS 2 CAUSE CODE
		X'004E'	CLASS 3 CAUSE CODE
		X'004F'	CLASS 4 CAUSE CODE
		X'0050'	CLASS 5 CAUSE CODE
		X'0051'	CLASS 6 CAUSE CODE
		X'0052'	CLASS 7 CAUSE CODE
		X'0053'	RAI SECONDS RECEIVED
		X'0054'	RAI SECONDS TRANSMITTED
		X'0055'	AIS SECONDS RECEIVED

Product Set ID Index (X'83') Network Alert Common Subfield

This subfield contains a code point and a count that jointly specify a particular Product ID (X'11') subvector within a particular Product Set ID (X'10') subvector in the Alert major vector.

Product Set ID Index (X'83') Network Alert Common Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Product Set ID Index subfield
1		Key: X'83'

Product Set ID Index (X'83') Network Alert Common Subfield

Byte	Bit	Content
2(=q)		Product ID code
	0–3	Product ID subvector code: a code point that specifies (1) the type of Product ID subvector being indexed (hardware or software), and (2) the particular data to be extracted from this subvector *Note:* See "Product Identifier (X'11') MS Common Subvector" on page 9-355 for the criteria distinguishing hardware and software Product ID subvectors. X'0'–X'1' reserved X'2' (machine type or hardware product common name) from a hardware Product ID subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'3' serial number or repair ID number, whichever is present, from a hardware Product ID subvector X'4' (machine type or hardware product common name) plus serial number or repair ID number, whichever is present, from a hardware Product ID subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'5' (machine type or hardware product common name) plus model number from a hardware Product ID subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'6' (machine type or hardware product common name) plus model number plus serial number or repair ID number, whichever is present, from a hardware Product ID subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'9' software product common name from a software Product ID subvector
	4	Product set ID indicator: An indication of which Product Set ID contains the Product ID subvector being indexed 0 Alert sender PSID 1 indicated resource PSID
	5–7	Count: a three-digit binary number that indicates which Product ID subvector, of the type specified by the Product ID Subvector Code, is being indexed within the PSID specified by the Product Set ID Indicator. *Note:* This count applies only to Product ID subvectors of the type specified by the Product ID Subvector Code. If, for example, the code is X'2' (specifying a hardware Product ID) then only hardware X'11's are counted: a count of X'3' would thus index the third hardware Product ID within the PSID indicated by the Product Set ID Indicator.

Resource List Index (X'84') Network Alert Common Subfield

This subfield contains data identifying a single resource within a resource list subvector in the Alert major vector.

Resource List Index (X'84') Network Alert Common Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource List Index subfield
1		Key: X'84'
2		Resource list subvector key: The key of the subvector containing the identification of the resource being indexed
3		Subfield key: Within the subvector identified in byte 2, the key of the subfield containing the identification of the resource being indexed
4		Entry count: Within the subfield identified in byte 3, the count of the resource entry being indexed. Only entries having their Resource List Indexing Flag set are counted. (In the Hierarchy Name List (X'10') and Associated Resources (X'11') subfields of the Hierarchy/Resource List (X'05') subvector, this flag is bit 2 of byte r + 1 of the entry.)
5(= q)		Requested data: For the resource entry identified in byte 4, a code point indicating what data is to be displayed: X'00' resource name X'01' resource type and name X'02' resource type

Detailed Data Extended (X'85') Network Alert Common Subfield

This subfield contains product specific detailed data to be displayed at an Alert receiver. Since it provides additional function (namely additional Data ID encodings), the Detailed Data Extended (X'85') subfield should be used instead of the Detailed Data (X'82') subfield when possible.

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Detailed Data Extended subfield *Note:* Length = X'02' indicates that the Product ID Code, Reserved, Data ID, Data Encoding, and Detailed Data fields are not present.
1		Key: X'85'

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content
2		Product ID code: a code indicating what product identification, if any, must be displayed in conjunction with the data type and data. The structure of this field is identical to that present in the Product Set ID Index (X'83') subfield. A value of X'00' in this byte indicates that no product identification data is displayed in conjunction with the data type and detailed data.
	0–3	Product ID subvector code: a code point that specifies (1) the type of Product ID subvector being indexed (hardware or software), and (2) the particular data to be extracted from this subvector *Note:* See "Product Identifier (X'11') MS Common Subvector" on page 9-355 for the criteria distinguishing hardware and software Product ID subvectors. X'0'–X'1' reserved X'2' (machine type or hardware product common name) from a hardware Product ID Subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'5' (machine type or hardware product common name) plus model number from a hardware Product ID Subvector *Note:* The hardware product common name is used if it is present; otherwise, the machine type is used. X'9' software product common name from a software Product ID subvector
	4	Product set ID indicator: an indication of which Product Set ID (PSID) contains the Product ID subvector being indexed 0 Alert sender PSID 1 Indicated resource PSID
	5–7	Count: a 3-digit binary number that indicates which Product ID subvector, of the type specified by the Product ID subvector code, is being indexed within the PSID specified by the Product Set ID Indicator. *Note:* This count applies only to Product ID subvectors of the type specified by the Product ID subvector code. If, for example, the code is X'2' (specifying a hardware Product ID) then only hardware X'11's are counted: a count of X'3' would thus index the third hardware Product ID within the PSID indicated by the Product Set ID Indicator.
3		Reserved, must be X'00'. *Note:* This field is reserved for future architectural extensions, **not** product specific uses.
4–5		Data ID: a code point indicating the type of data carried in the subfield. The English text associated with each code point, or its national language equivalent, is displayed in conjunction with the detailed data. Defined codes are: X'0000' (no display) X'0001' ABEND CODE X'0002' ADAPTER CHECK STATUS X'0003' ADAPTER RETURN CODE X'0004' BOP CODE X'0005' PROTOCOL CODE X'0006' COMMAND CODE

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content	
		X'0007'	ERROR CODE
		X'0008'	OPERATING SYSTEM RETURN CODE
		X'0009'	EVENT CODE
		X'000A'	LLC ERROR CODE
		X'000B'	MACHINE CHECK CODE
		X'000C'	MALFUNCTION CODE
		X'000D'	PROGRAM CHECK CODE
		X'000E'	REASON CODE
		X'000F'	RETURN CODE
		X'0010'	SENSE CODE
		X'0011'	SENSE DATA
		X'0012'	SOFTWARE ERROR CODE
		X'0013'	STATUS CODE
		X'0014'	SYMPTOM CODE
		X'0015'	SNA SENSE DATA
		X'0016'	BUS STATUS CODE
		X'0017'	RING STATUS CODE
		X'0018'	CALL PROGRESS SIGNAL: A notification from a network to a DTE, indicating why a connection could not be established
		X'0019'	FILE
		X'001A'	X.25 CLEAR PACKET, CAUSE CODE: A code to or from an X.25 network indicating the reason that a CLEAR request or indication packet was sent *Note:* This indicates the reason that a network connection was lost or could not be established.
		X'001B'	X.25 RESET PACKET, CAUSE CODE: A code to or from an X.25 network indicating the reason that a RESET request or indication packet was sent *Note:* This indicates the reason that a network connection was lost or could not be established.
		X'001C'	X.25 RESTART PACKET, CAUSE CODE: A code to or from an X.25 network indicating the reason that a RESTART request or indication packet was sent *Note:* This indicates the reason that a network connection was lost or could not be established.
		X'001D'	X.25 DIAGNOSTIC CODE: A code to or from an X.25 network providing additional information about why a Diagnostic packet or a Clear, Reset, or Restart request or indication packet was sent.
		X'001E'	DIAGNOSTIC EXPLANATION
		X'001F'	DATABASE RETURN CODE
		X'0020'	MESSAGE CODE
		X'0021'	PANEL ERROR MESSAGE CODE
		X'0022'	SYSTEM MESSAGE CODE
		X'0023'	MESSAGE SEVERITY
		X'0024'	WAIT STATE CODE
		X'0025'	PRIMARY RETURN CODE
		X'0026'	SECONDARY RETURN CODE

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content	
		X'0027'	FUNCTION
		X'0028'	SOCKET NUMBER: The TCP/IP endpoint for the communication connection
		X'0029'	DRIVE
		X'002A'	V.25BIS INDICATION
		X'002B'	ICA SENSE CODE
		X'002C'	MAINTENANCE INDICATOR
		X'002D'	ACCESS UNIT ID
		X'002E'	CONTROLLED ACCESS UNIT LOBE NUMBER
		X'002F'	D-CHANNEL NUMBER
		X'0030'	REFERENCE CODE
		X'0031'	SYSTEM REFERENCE CODE
		X'0032'	REPLACEABLE UNIT CODE
		X'0033'	COMPONENT ID
		X'0034'	COMMUNICATION CONTROL UNIT
		X'0035'	TYPE
		X'0036'	LOCATION
		X'0037'	PART NUMBER
		X'0038'	DESTINATION
		X'0039'	ORIGINATOR
		X'003A'	RACK/UNIT
		X'003B'	RACK
		X'003C'	UNIT
		X'003D'	CARD SLOT
		X'003E'	CABLE POSITION
		X'003F'	INTERCONNECT CONTROLLER
		X'0040'	ERROR RECOVERY PROCEDURE
		X'0041'	PDP CODE
		X'0042'	BACKUP PATH STATUS
		X'0043'	WRAP STATUS
		X'0044'	ATTACHMENT MODULE STATUS
		X'0045'	CONFIGURATION DATA
		X'0046'	LOBES DEACTIVATED
		X'0047'	ATTACHMENT MODULE NUMBER
		X'0048'	NUMBER OF LINKS LOST
		X'0049'	NUMBER OF NETBIOS SESSIONS LOST
		X'004A'	NETID
		X'004B'	LOCATION NAME
		X'004C'	CLASS OF SERVICE
		X'004D'	BACK-UP D-CHANNEL NUMBER
		X'004E'	ROUTE AFFECTED
		X'004F'	ACTIVE ROUTE
		X'0050'	CHANNEL UNIT ADDRESS
		X'0051'	DEVICE ADDRESS
		X'0052'	LINE ADDRESS
		X'0053'	LINE ADDRESS RANGE

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content	
		X'0054'	ADAPTER AT ADDRESS
		X'0055'	LINE
		X'0056'	DTE ADDRESS CALLED
		X'0057'	DTE ADDRESS CALLING
		X'0058'	LOCAL DTE ADDRESS
		X'0059'	CHANNEL
		X'005A'	SERVER
		X'005B'	CARD NUMBER
		X'005C'	CARD NAME
		X'005D'	CARD LOCATION
		X'005E'	LAN BRIDGE ID
		X'005F'	TOKEN-RING ID
		X'0060'	PORT NUMBER
		X'0061'	ADAPTER NUMBER
		X'0062'	CHANNEL ADAPTER NUMBER
		X'0063'	LINE ADAPTER NUMBER
		X'0064'	LINE INTERFACE COUPLER (LIC) POSITION
		X'0065'	BUS NUMBER
		X'0066'	TOKEN RING INTERFACE COUPLER NUMBER
		X'0067'	LOCALLY-INITIATED LOGICAL CHANNEL
		X'0068'	REMOTELY-INITIATED LOGICAL CHANNEL
		X'0069'	CONTROLLER ID
		X'006A'	DEVICE ID
		X'006B'	DEVICE
		X'006C'	TRANSMISSION PRIORITY FIELD
		X'006D'	PROGRAM STATUS WORD
		X'006E'	DOMAIN CONTROLLER
		X'006F'	PEER SERVER
		X'0070'	GENERATION PARAMETER
		X'0071'	THRESHOLD PARAMETER
		X'0072'	CONFIGURATION OBJECT/RECORD: Identifies the configuration object or record which contains one or more user settable parameters.
		X'0073'	CONFIGURATION PARAMETER
		X'0074'	IPL PARAMETER
		X'0075'	PARAMETER VALUE
		X'0076'	SECURITY DATABASE FILE
		X'0077'	REQUESTER
		X'0078'	ACCOUNT SYSTEM FILE
		X'0079'	SUBSTITUTE PART NUMBER
		X'007A'	CENTRAL PROCESSOR COMPLEX
		X'007B'	CENTRAL PROCESSING UNIT: The CPU includes its associated vector element processor.
		X'007C'	LOGICAL PARTITION NAME
		X'007D'	SUBCHANNEL NUMBER
		X'007E'	CHANNEL PATH ID
		X'007F'	I/O PROCESSING ELEMENT ID

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content	
		X'0080'	NODE
		X'0081'	LINK STATION
		X'0082'	CP
		X'0083'	PU
		X'0084'	LU
		X'0085'	TRANSACTION PROGRAM
		X'0086'	LSL: Link Segment Level of a multi-segment link connection *Note:* In a multi-segment link connection, link segments are numbered in ascending order, from the error notification sender outwards; thus the link segment immediately adjacent to the error notification sender is Link Segment Level 1, the one adjacent to it is Link Segment Level 2, etc.
		X'0087'	RELATIONAL DATABASE
		X'0088'	STORAGE
		X'0089'	FULLY QUALIFIED PCID
		X'008A'	PACKAGE
		X'008B'	TABLE
		X'008C'	VOLUME
		X'008D'	DIRECTORY
		X'008E'	LOGICAL PARTITIONS
		X'008F'	VECTOR PROCESSOR
		X'0090'	YEAR/MONTH/DAY
		X'0091'	TIME
		X'0092'	JULIAN DATE
		X'0093'	MINUTES
		X'0094'	ACCOUNT
		X'0095'	FILE SERVER
		X'0096'	DATA STREAM
		X'0097'	DATA STREAM OFFSET
		X'0098'	USER
		X'0099'	CONNECTION
		X'009A'	NETWORK
		X'009B'	SERVICE PROCESSOR
		X'009C'	EXPANDED STORAGE
		X'009D'	CENTRAL PROCESSOR COMPLEX IMAGE
		X'009E'	SERVICE UPDATE
		X'009F'	CUSTOMER PROBLEM REPORT
		X'00A0'	BYTE OFFSET
		X'00A1'	BIT OFFSET
		X'00A2'	DETECTING MODULE
		X'00A3'	FAILING MODULE
		X'00A4'	MAINTENANCE LEVEL
		X'00A5'	COMMAND
		X'00A6'	PROGRAM
		X'00A7'	RESOURCE
		X'00A8'	MAXIMUM NUMBER OF NODE TABLE ENTRIES

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content	
		X'00A9'	MODE NAME
		X'00AA'	UNBIND TYPE
		X'00AB'	INBOUND CALL
		X'00AC'	SYSTEM ERROR
		X'00AD'	REMOTE SUPPORT FACILITY
		X'00AE'	CENTRAL OPERATIONS
		X'00AF'	CONTROL PROGRAM
		X'00B0'	EIA STANDARD
		X'00B1'	CCITT STANDARD
		X'00B2'	CHANNEL SUBSYSTEM
		X'00B3'	MESSAGE LOG
		X'00B4'	OPERATING SYSTEM LOG
		X'00B5'	DUMP
		X'00B6'	RETRY COUNT
		X'00B7'	TIMER SETTING
		X'00B8'	LINE SPEED (BITS PER SECOND)
		X'00B9'	LINE SPEED (KILOBITS PER SECOND)
		X'00BA'	LINE SPEED (MEGABITS PER SECOND)
		X'00BB'	OPERATION NUMBER
		X'00BC'	OPERATION PRIORITY
		X'00BD'	OPERATION INPUT ARRIVAL TIME
		X'00BF'	APPLICATION INPUT ARRIVAL TIME
		X'00C0'	COUNTER
		X'00C1'	ALARM SIGNAL
		X'00C2'	JOB NUMBER
		X'00C3'	ISDN CAUSE CODE
		X'00C4'	CONTROLLER DESCRIPTION OBJECT
		X'00C5'	COMMUNICATION LINE DESCRIPTION OBJECT
		X'00C6'	LOGICAL DEVICE DESCRIPTION OBJECT
		X'00C7'	NETWORK INTERFACE DESCRIPTION OBJECT
		X'00C8'	NETWORK INTERFACE NAME
		X'00C9'	NAME
		X'00CA'	JOB NAME
		X'00CB'	LINE NAME
		X'00CC'	MESSAGE FILE NAME
		X'00CD'	MESSAGE QUEUE NAME
		X'00CE'	MESSAGE REFERENCE KEY
		X'00CF'	REMOTE LOCATION NAME
		X'00D0'	FILE NAME
		X'00D1'	LOG RECORD NUMBER
		X'00D2'	CARTRIDGE: A component that holds items to be dispensed
		X'00D3'	AIR FILTER NUMBER
		X'00D4'	TELEPHONE NUMBER
		X'00D5'	CALLING TELEPHONE NUMBER
		X'00D6'	TELEPHONE NUMBER CALLED

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content
		X'00D7' REPORTING TELEPHONE NUMBER: The telephone number of the Alert sender
		X'00D8' TIMER
		X'00D9' LOG RECORD TYPE
		X'00DA' LOG ID
		X'00DB' PUBLICATION NUMBER
		X'00DC' NEW NUMBER
		X'00DD' APPLICATION NAME
		X'00DE' NUMBER
		X'00DF' WORKSTATION ID
		X'00E0' – X'00EF' Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.
		X'00F0' PRODUCT ALERT REFERENCE CODE: A code that identifies an Alert in a user-friendly way. The product Alert reference code is used to index documentation provided by the Alert sending product. This documentation can group the Alerts into natural categories and provide extended explanations or diagnostic information.
		X'00F1' QUEUE
		X'00F2' INTERFACE
		X'00F3' SERVICE ACTION CODE
		X'00F4' PROBLEM MANAGEMENT HARDWARE NUMBER
		X'00F5' PROBLEM DATA
		X'00F6' DISK BLOCK NUMBER
		X'00F7' CONTROLLER BUS ADAPTER
		X'00F8' ENTERPRISE
		X'00F9' CONTACT ID
		X'00FA' SNMP GENERIC-TRAP NUMBER
		X'00FB' SNMP SPECIFIC-TRAP NUMBER
		X'00FC' SNMP MIB VARIABLE NAME
		X'00FD' SNMP MIB VARIABLE VALUE
		X'00FE' INTERNET PROTOCOL ADDRESS
		X'00FF' SERVICE POINT APPLICATION
		X'0100' SYSTEM NAME
		X'0101' CORRELATION ID
		X'0102' TRACE RECORD SEQUENCE NUMBER
		X'0103' SUBSYSTEM ID
		X'0104' PHYSICAL DEVICE
		X'0105' CYLINDER
		X'0106' HEAD
		X'0107' MEDIA MAINTENANCE PROCEDURE
		X'0108' STORAGE CLUSTER
		X'0109' DEVICE PATH
		X'010A' STORAGE PATH
		X'010B' NONVOLATILE STORAGE FUNCTION

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content
		X'010C' DUAL COPY FUNCTION X'010D' CACHE FUNCTION X'010E' CONTROLLER X'010F' RESOURCE MANAGER ID X'0110' PRODUCT ALARM REFERENCE CODE: This codepoint is used in an Alert that was originally an OSI alarm. The Product Alarm Reference Code is used to index documentation provided by the alarm sending product. This documentation can group the alarms into natural categories and provide extended explanation of diagnostic information. X'0111' LUWID: Logical-Unit-of-Work Identifier X'0112' PROBE ID X'0113' PROBE SEVERITY CODE X'E000'–X'EFFF' Reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.
6		Data Encoding: a code point indicating how the accompanying detailed data is encoded, and, thus, how it is to be displayed. Defined code are: X'00' hexadecimal: The data is to be displayed as hexadecimal digits. X'01' binary: The data is the binary representation of an unsigned integer value (8, 16, 24, or 32 bits in length). The decimal equivalent is to be displayed. For example, the value B'11111111' (X'FF') is to be displayed as 255. X'10' hexadecimal code point: The data is a two-byte hexadecimal code point which is used to index a text string stored at the Alert receiver. The text string will be displayed in place of the hexadecimal code point that was transported in the Alert. X'11' Coded Graphic Character Set 00640–00500 plus: The data is to be decoded using Coded Graphic Character Set 00640–00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," plus three additional code points: X'5B' = "$" (dollar sign) X'7B' = "#" (number sign) X'7C' = "@" (at sign) *Note:* Detailed data encoded in this way is limited to codes, numbers, or internationally recognized terms that do not require translation.
7–q		Detailed data, encoded as specified when byte offset 6 = X'00', X'01', or X'11'.
Or		
7–8		Detailed data, encoded as specified when byte offset 6 = X'10' (hexadecimal code point). The hexadecimal code point has a length of two bytes, and provides an index to predefined text that is displayed at the Alert receiver.

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content

An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent. The method chosen must be consistent with the language of the code point text that "imbeds" the X'85' subfield text.

These code points are logically grouped, one group per Data ID code point. Within a group, a code point may have only one text string associated with it. But, the same code point may have different text strings associated with it in other groups. The Data ID determines the "table" that is searched by the Alert receiver in order to find the appropriate text string.

Note: The X'E000' – X'EFFF' range of code points in each table is reserved. Neither IBM products nor non-IBM products may assign a code point in this range.

Specific defined codes and the corresponding displayed text are listed below for each group.

Group: Counter Names (Data ID = X'00C0')

The code point assignments below are valid when used with the Data ID value of X'00C0' (Counter). Defined hexadecimal codes are:

X'0001'	LOSS OF FRAME ALIGNMENT
X'0002'	DETECTED ACCESS TRANSMISSION SYSTEM ERRORS (DTSE) IN
X'0003'	DETECTED ACCESS TRANSMISSION SYSTEM ERRORS (DTSE) OUT
X'0004'	TE-DETECTED CODE VIOLATIONS
X'0005'	NT-DETECTED CODE VIOLATIONS
X'0006'	CRC ERRORS RECEIVED
X'0007'	CRC ERRORS TRANSMITTED
X'0008'	ERRORED SECONDS
X'0009'	SEVERELY ERRORED SECONDS
X'000A'	COLLISION DETECT
X'000B'	SHORT FRAMES RECEIVED
X'000C'	TRANSMIT OVERRUN
X'000D'	RECEIVE UNDERRUN
X'000E'	TA ASYNCHRONOUS ERRORS
X'000F'	TA SYNCHRONOUS ERRORS
X'0010'	ABORTED FRAMES RECEIVED
X'0011'	ABORTED FRAMES TRANSMITTED
X'0012'	MISADDRESSED FRAMES RECEIVED
X'0013'	UNBOUNDED FRAMES RECEIVED
X'0014'	NON-INTEGRAL FRAMES RECEIVED
X'0015'	TOTAL FRAMES TRANSMITTED
X'0016'	TOTAL FRAMES RECEIVED
X'0017'	PDUS RETRANSMITTED
X'0018'	RECEIVE SEQUENCE ERRORS
X'0019'	TOTAL BYTES TRANSMITTED

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content	
		X'001A'	TOTAL BYTES RECEIVED
		X'001B'	TOTAL BYTES RETRANSMITTED
		X'001C'	TOTAL INCOMING CALLS
		X'001D'	TOTAL OUTGOING CALLS
		X'001E'	INCOMING CALLS REJECTED
		X'001F'	OUTGOING CALLS REJECTED
		X'0020'	INFORMATION FRAMES TRANSMITTED
		X'0021'	INFORMATION FRAMES RECEIVED
		X'0022'	PDUS DISCARDED
		X'0023'	TOTAL CONNECTIONS
		X'0024'	UNNUMBERED INFORMATION FRAMES TRANSMITTED
		X'0025'	UNNUMBERED INFORMATION FRAMES RECEIVED
		X'0026'	LAN TYPE 3 FRAMES TRANSMITTED
		X'0027'	LAN TYPE 3 FRAMES RECEIVED
		X'0028'	LAN TYPE 3 FRAMES RETRANSMITTED
		X'0029'	LAN TYPE 2 ACKNOWLEDGMENT TIMER TIMEOUTS
		X'002A'	LOCAL BUSY OCCURRENCES
		X'002B'	TOKEN RING MAC LINE ERRORS
		X'002C'	TOKEN RING MAC BURST ERRORS
		X'002D'	TOKEN RING MAC A/C ERRORS
		X'002E'	TOKEN RING MAC INTERNAL ERRORS
		X'002F'	TOKEN RING MAC LOST FRAME ERRORS
		X'0030'	TOKEN RING MAC RECEIVE CONGESTION ERRORS
		X'0031'	TOKEN RING MAC FRAME-COPIED ERRORS
		X'0032'	TOKEN RING MAC TOKEN ERRORS
		X'0033'	TOKEN RING MAC FREQUENCY ERRORS
		X'0034'	UNRECOGNIZED PDUS
		X'0035'	TEST COMMANDS RECEIVED
		X'0036'	TEST RESPONSES TRANSMITTED
		X'0037'	TIMER (MILLISECONDS)
		X'0038'	FRAME-LOSS SECONDS NEAR-END
		X'0039'	FRAME-LOSS SECONDS FAR-END
		X'003A'	RESTART MESSAGES RECEIVED
		X'003B'	STATUS ENQUIRY MESSAGES RECEIVED
		X'003C'	ERRORED SECONDS NEAR-END
		X'003D'	ERRORED SECONDS FAR-END
		X'003E'	BURSTY ERRORED SECONDS NEAR-END
		X'003F'	BURSTY ERRORED SECONDS FAR-END
		X'0040'	SEVERELY ERRORED SECONDS NEAR-END
		X'0041'	SEVERELY ERRORED SECONDS FAR-END
		X'0042'	SLIP SECONDS NEAR-END
		X'0043'	SLIP SECONDS FAR-END
		X'0044'	BADLY FORMED FRAMES
		X'0045'	TOTAL POLLS TRANSMITTED
		X'0046'	TOTAL POLLS RECEIVED
		X'0047'	TOTAL UNNUMBERED FRAMES TRANSMITTED

Detailed Data Extended (X'85') Network Alert Common Subfield

Byte	Bit	Content
		X'0048' TOTAL UNNUMBERED FRAMES RECEIVED
		X'0049' TOTAL SUPERVISORY FRAMES TRANSMITTED
		X'004A' TOTAL SUPERVISORY FRAMES RECEIVED
		X'004B' CLASS 0 CAUSE CODE
		X'004C' CLASS 1 CAUSE CODE
		X'004D' CLASS 2 CAUSE CODE
		X'004E' CLASS 3 CAUSE CODE
		X'004F' CLASS 4 CAUSE CODE
		X'0050' CLASS 5 CAUSE CODE
		X'0051' CLASS 6 CAUSE CODE
		X'0052' CLASS 7 CAUSE CODE
		X'0053' RAI SECONDS RECEIVED
		X'0054' RAI SECONDS TRANSMITTED
		X'0055' AIS SECONDS RECEIVED

Resolution (X'0002') MS Major Vector

PU → SSCP, CP → CP

This major vector provides unsolicited notification of the resolution of a problem previously reported by an Alert. It contains an identification of the type of problem resolution, and an identification of the actual failing resource.

Resolution (X'0002') MS Major Vector

Byte	Bit	Content
0−1		Length (n+1), in binary, of this MS major vector
2−3		Key: X'0002'
4−n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'−X'7F', and in "Resolution MS Subvectors" on page 9-185 for subvector keys X'80'−X'FE'. The following subvector keys may be used as indicated:

Subvector	Presence in Resolution (X'0002') Major Vector		
	NMVT	CP-MSU	Notes
Date/Time (X'01')	CP	P	Note 1
SNA Address List (X'04')	CP	—	Note 2
Hierarchy/Resource List (X'05')	CP	P	Note 3
Product Set ID (X'10')	P(n)	P(n)	Note 4
Self-Defining Text Msg. (X'31')	O	O	
Relative Time (X'42')	CP	CP	Note 5
Incident Identification (X'4A')	P(n)	P(n)	Note 6
Resolution Data (X'92')	P	P	
Actual Causes (X'93')	P	P	
Actual User Causes (X'94')	CP	CP	Note 7
Actual Install Causes (X'95')	CP	CP	Note 7
Actual Failure Causes (X'96')	CP	CP	Note 7
Actual Cause Undetermined (X'97')	CP	CP	Note 8
Detailed Data (X'98')	O	O	

Key:

—	Not present
P	Present one time
P(n)	Present one or more times
CP	Conditionally present one time (See Notes for conditions)
O	Optionally present one time

Notes:

1. If the PU sending the Resolution major vector has the capability of providing it, it places this subvector in the NMVT. See Note 5. It is always present in a CP-MSU.
2. This subvector is present when it is necessary to identify, with an SNA address, the origin of the resolved problem. If the origin of the resolved problem is the PU sending the Resolution major vector, this subvector is not present.
3. This subvector is present in the NMVT instead of, or in addition to, the SNA Address List (X'04') subvector if the origin (other than the PU sending the Resolution major vector) of the resolved problem cannot be represented in the SNA Address List (X'04') subvector. This subvector is always present in a CP-MSU.
4. An instance of this subvector describing the PU or CP sending the Resolution major vector is always present. A second instance is present if the origin of the resolved problem is a hardware or software product, and is not the PU or CP sending the Resolution major vector. If a second instance is present, it is placed immediately after the first instance of the X'10' subvector.

In a Resolution major vector containing two instances of the Product Set ID subvector, the following terms refer, respectively, to these two instances:

- "Resolution Notification Sender PSID"—identifies the PU or CP sending the resolution notification
- "Indicated Resource PSID"—identifies the resource whose problem has been resolved

In a Resolution major vector with only one instance of the Product Set ID, this instance is referred to both as the Resolution Notification Sender Product Set ID and as the Indicated Resource Product Set ID.

5. If the PU sending the Resolution major vector cannot provide a Date/Time (X'01') subvector, it places this subvector in the NMVT instead.
6. This subvector is present in order to correlate the Resolution major vector (reporting a resolved problem) with the Alert that originally reported the problem. The subvector provides the correlation via a unique token, the incident identifier. Note it is possible that the resolved problem originally appeared to be several problems, and thus caused a group of Alerts to be sent. In this case the incident identifier from each of the Alerts sent may be included in this subvector. If the incident identifiers do not fit within a single X'4A' subvector, then multiple X'4A' subvectors may be present.
7. Any or all of these subvectors may be present in a Resolution major vector, depending on the ability of the sender to relate the actual causes of the resolved problem.
8. This subvector is present in a Resolution major vector if and only if none of the X'94', X'95', and X'96' subvectors is present.

Resolution MS Subvectors

Resolution Data (X'92') Resolution MS Subvector

This subvector transports information related to the resolution of an Alert condition in the form of code points that correspond to strings of text stored at the receiver of this flow.

Resolution Data (X'92') Resolution MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Resolution Data subvector
1		Key: X'92'

Resolution Data (X'92') Resolution MS Subvector

Byte	Bit	Content
2		Resolution type: a hexadecimal value indicating the type of resolution that occurred: X'01' resolved due to service X'02' resolved due to permanent bypass X'03' resolved due to temporary bypass X'04' resolved (component permanently placed offline) X'05' resolved with no action X'06' incident closed, but problem still exists X'07' resolved for unknown reason X'08' resolved into duplicate incident
3–4		Resolution Description Code: A code point that provides an index to predefined text describing the condition that has been resolved. A receiver of this subvector has two options for selecting text to display. It can display the English text documented with each code point, or its national language equivalent; or, for a presentation to an operator of a lower skill level, it can choose the following simpler text (shown all capitalized), or its national language equivalent, based only on the first digit of the code point: X'1xxx' HARDWARE X'2xxx' SOFTWARE X'3xxx' COMMUNICATIONS X'4xxx' PERFORMANCE X'5xxx' CONGESTION X'6xxx' MICROCODE X'7xxx' OPERATOR X'8xxx' SPECIFICATION X'9xxx' INTERVENTION REQUIRED X'Bxxx' NOTIFICATION X'Cxxx' SECURITY X'Fxxx' UNDETERMINED Specific defined codes and the corresponding displayed text (shown all capitalized) are listed under the Alert Description Code field in the "Generic Alert Data (X'92') Alert MS Subvector" on page 9-16. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules).
5–6		Flags:
	0	Initiation indicator: 0 Resolution notification not directly initiated by an operator action. 1 Resolution notification initiated by an operator action.
	1	Held indicator: 0 Resolution notification was sent when the problem was resolved. 1 The problem was resolved earlier, but the resolution notification was not sent at that time because no session was available to send it on.

Resolution Data (X'92') Resolution MS Subvector

Byte	Bit	Content
	2–15	Reserved

Actual Causes (X'93') Resolution MS Subvector

This subvector contains one or more code points denoting the actual causes of the resolved Alert condition. Correction of these causes resulted in the generation of the Resolution major vector. The actual causes are not reported as probabilities, therefore the order of the actual cause code points is not meaningful.

Actual Causes (X'93') Resolution MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Actual Causes subvector
1		Key: (X'93')
2–p		One or more two-byte actual cause code points. Each code point provides an index to predefined text denoting an actual cause which has been corrected. A receiver of this subvector has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point (not indented) above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are the same as those used in the Probable Causes (X'93') Alert MS Subvector, documented on page 9-31. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules).

Actual User Causes (X'94') Resolution MS Subvector

This subvector transports code points for stored text detailing the actual user causes of the resolved Alert condition, and the actual actions taken to correct these causes. It may also transport additional detailed data, to be inserted into the text indexed by the actual user cause and/or actual action code points.

Actual User Causes (X'94') Resolution MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Actual User Causes subvector
1		Key: X'94'
2–p		Two or more subfields containing actual user cause data, as described below for keys X'00'–X'7F' and in "Resolution (X'0002') Common Subfields" on page 9-195 for keys X'80'–X'FE'. X'01' Actual User Causes X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended X'86' Actual Actions *Note:* Subfields X'01' and X'86' are always present and the X'01' always precedes the X'86'. Depending on the code points present in the X'01' and X'86' subfields: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

Actual User Causes (X'01') Actual User Causes Subfield

This subfield contains one or more code points denoting actual user causes of the resolved Alert condition. The actual user causes are not reported as probabilities, therefore the order of the actual user cause code points is not meaningful. An actual user cause is defined to be a condition that originally caused the (now resolved) Alert condition, and which was resolved by an operator without contacting any service organization.

Actual User Causes (X'01') Actual User Causes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Actual User Causes subfield
1		Key: X'01'
2–q		Two-byte actual user cause code points. Each code point provides an index to predefined text, describing the actual user cause, that is displayed at the focal point. A receiver of this subvector has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent.

Actual User Causes (X'01') Actual User Causes Subfield

Byte	Bit	Content

Specific defined codes and the corresponding displayed text (shown all capitalized) are the same as those used in the User Causes (X'94') Alert MS Subvector, documented on page 9-54. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules).

The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the X'94' subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. **Note:** The X'82' and X'85' subfields can *not* be both used in the same subvector.

The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

The third digit of each user cause code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows:

X'xx0x' – X'xx9x':	No detailed data qualifiers.
X'xxAx' – X'xxBx':	One detailed data qualifier.
X'xxCx':	Two detailed data qualifiers.
X'xxDx':	Three detailed data qualifiers.
X'xxEx':	One X'83' subfield.
X'xxFx':	One X'84' subfield.

Actual Install Causes (X'95') Resolution MS Subvector

This subvector transports code points for stored text detailing the actual install causes of the resolved Alert condition, and the actual actions taken to correct these causes. It may also transport additional detailed data, to be inserted into the text indexed by the actual install cause and/or actual action code points.

Actual Install Causes (X'95') Resolution MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Actual Install Causes subvector
1		Key: X'95'
2−p		Two or more subfields containing actual install cause data, as described below for keys X'00'−X'7F' and in "Resolution (X'0002') Common Subfields" on page 9-195 for keys X'80'−X'FE'. X'01' Actual Install Causes X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended X'86' Actual Actions *Note:* Subfields X'01' and X'86' are always present and the X'01' always precedes the X'86'. Depending on the code points present in the X'01' and X'86' subfields: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

Actual Install Causes (X'01') Actual Install Causes Subfield

This subfield contains one or more code points denoting actual install causes of the resolved Alert condition. The actual install causes are not reported as probabilities, therefore the order of the actual install cause code points is not meaningful. An actual install cause is defined to be a condition that originally caused the (now resolved) Alert condition and resulted from the initial installation or set-up of some equipment.

Actual Install Causes (X'01') Actual Install Causes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Actual Install Causes subfield
1		Key: X'01'
2−q		Two-byte actual install cause code points. Each code point provides an index to predefined text, describing the actual install cause, that is displayed at the focal point. A receiver of this subvector has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are the same as those used in the Install Causes (X'95') Alert MS Subvector, documented on page 9-68. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules). The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the X'95' subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. **Note:** The X'82' and X'85' subfields can *not* be both used in the same subvector. The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

Actual Install Causes (X'01') Actual Install Causes Subfield

Byte	Bit	Content
		The third digit of each install cause code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows:
		X'xx0x' – X'xx9x': No detailed data qualifiers.
		X'xxAx' – X'xxBx': One detailed data qualifier.
		X'xxCx': Two detailed data qualifiers.
		X'xxDx': Three detailed data qualifiers.
		X'xxEx': One X'83' subfield.
		X'xxFx': One X'84' subfield.

Actual Failure Causes (X'96') Resolution MS Subvector

This subvector transports code points for stored text detailing the actual failure causes of the resolved Alert condition, and the actual actions taken to correct these causes. It may also transport additional detailed data, to be inserted into the text indexed by the actual failure cause and/or actual action code points.

Actual Failure Causes (X'96') Resolution MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Actual Failure Causes subvector
1		Key: X'96'
2 – p		Two or more subfields containing actual failure cause data, as described below for keys X'00' – X'7F' and in "Resolution (X'0002') Common Subfields" on page 9-195 for keys X'80' – X'FE'. X'01' Actual Failure Causes X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended

Actual Failure Causes (X'96') Resolution MS Subvector

Byte	Bit	Content
		X'86' Actual Actions *Note:* Subfields X'01' and X'86' are always present and the X'01' always precedes the X'86'. Depending on the code points present in the X'01' and X'86' subfields: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

Actual Failure Causes (X'01') Actual Failure Causes Subfield

This subfield contains one or more code points denoting actual failure causes of the resolved Alert condition. The actual failure causes are not reported as probabilities, therefore the order of the actual failure cause code points is not meaningful. An actual failure cause is defined to be a condition that originally caused the (now resolved) Alert condition and resulted from the failure of a resource.

Actual Failure Causes (X'01') Actual Failure Causes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Actual Failure Causes subfield
1		Key: X'01'
2−q		Two-byte actual failure cause code points. Each code point provides an index to predefined text, describing the actual failure cause, that is displayed at the focal point. A receiver of this subvector has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are the same as those used in the Failure Causes (X'96') Alert MS Subvector, documented on page 9-84. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules).

Actual Failure Causes (X'01') Actual Failure Causes Subfield

Byte	Bit	Content

The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the X'96' subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield. **Note:** The X'82' and X'85' subfields can *not* be both used in the same subvector.

The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'01' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'01' subfield.

The third digit of each failure cause code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows:

X'xx0x' – X'xx9x':	No detailed data qualifiers.
X'xxAx' – X'xxBx':	One detailed data qualifier.
X'xxCx':	Two detailed data qualifiers.
X'xxDx':	Three detailed data qualifiers.
X'xxEx':	One X'83' subfield.
X'xxFx':	One X'84' subfield.

Actual Cause Undetermined (X'97') Resolution MS Subvector

This subvector transports code points for stored text detailing the actual actions taken to correct the (now resolved) Alert condition when no actual user, install, or failure causes of the Alert condition could be determined. It may also transport additional detailed data, to be inserted into the text indexed by the actual action code points.

Actual Cause Undetermined (X'97') Resolution MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Actual Cause Undetermined subvector
1		Key: X'97'
2–p		One or more subfields containing actual action data, as described in "Resolution (X'0002') Common Subfields" on page 9-195. X'82' Detailed Data X'83' Product Set ID Index X'84' Resource List Index X'85' Detailed Data Extended X'86' Actual Actions *Note:* Subfield X'86' is always present. Depending on the code points present in the X'86' subfield: • The X'83' and X'84' subfields may be present one or more times. • Either the X'82' or the X'85' subfield may be present one or more times. The X'82' and X'85' subfields are mutually exclusive within a subvector.

Detailed Data (X'98') Resolution MS Subvector

This subvector transports product specific detailed data.

Note: The format of this subvector is identical to that of the Detailed Data (X'98') Alert MS Subvector (documented on page 9-128), except that the Product Set ID Indicator (in each subfield) set to 0 indicates the resolution notification sender PSID (instead of the Alert sender PSID).

Resolution (X'0002') Common Subfields

The following table shows, by key value, the subfields common to subvectors that can occur within the Resolution major vector, and the subvectors in which each can occur.

Key	Subfield	Applicable Resolution Subvectors
X'82'	Detailed Data	Actual User Causes subvector, Actual Install Causes subvector, Actual Failure Causes subvector, Actual Cause Undetermined subvector, Detailed Data subvector
X'83'	Product Set ID Index	Actual User Causes subvector, Actual Install Causes subvector, Actual Failure Causes subvector, Actual Cause Undetermined subvector
X'84'	Resource List Index	Actual User Causes subvector,

	Actual Install Causes subvector, Actual Failure Causes subvector, Actual Cause Undetermined subvector
X'85' Detailed Data Extended	Actual User Causes subvector, Actual Install Causes subvector, Actual Failure Causes subvector, Actual Cause Undetermined subvector, Detailed Data subvector
X'86' Actual Actions	Actual User Causes subvector, Actual Install Causes subvector, Actual Failure Causes subvector, Actual Cause Undetermined subvector

Detailed Data (X'82') Resolution Common Subfield

This subfield contains product specific detailed data to be displayed at a resolution notification receiver. Since it provides additional function, the Detailed Data Extended (X'85') subfield should be used instead of the Detailed Data (X'82') subfield when possible.

Note: The format of this subfield is identical to that of the Detailed Data (X'82') Network Alert Common Subfield (documented on page 9-160), except that the Product Set ID Indicator set to 0 indicates the resolution notification sender PSID (instead of the Alert sender PSID).

Product Set ID Index (X'83') Resolution Common Subfield

This subfield contains a code point and a count that jointly specify a particular Product ID (X'11') subvector within a particular Product Set ID (X'10') subvector in the Resolution major vector.

Note: The format of this subfield is identical to that of the Product Set ID Index (X'83') Network Alert Common Subfield (documented on page 9-170), except that the Product Set ID Indicator set to 0 indicates the resolution notification sender PSID (instead of the Alert sender PSID).

Resource List Index (X'84') Resolution Common Subfield

This subfield contains data identifying a single resource within a resource list subvector in the Resolution major vector.

Note: The format of this subfield is identical to that of the Resource List Index (X'84') Network Alert Common Subfield, documented on page 9-172.

Detailed Data Extended (X'85') Resolution Common Subfield

This subfield contains product specific detailed data to be displayed at an resolution notification receiver. Since it provides additional function (namely additional Data ID encodings), the Detailed Data Extended (X'85') subfield should be used instead of the Detailed Data (X'82') subfield when possible.

Note: The format of this subfield is identical to that of the Detailed Data Extended (X'85') Network Alert Common Subfield (documented on page 9-172), except that the Product Set ID Indicator set to 0 indicates the resolution notification sender PSID (instead of the Alert sender PSID).

Actual Actions (X'86') Resolution Common Subfield

This subfield contains code points for stored text describing actions that were taken to rectify the Alert condition identified in the Resolution major vector.

Actual Actions (X'86') Resolution Common Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Actual Actions subfield
1		Key: X'86'
2−q		2-byte actual action code points. Each code point provides an index to predefined text, describing the action that occurred to resolve the incident, that is displayed at the Alert receiver. An Alert receiver has the option of displaying, for each code point it receives: either the text associated with that code point, or its national language equivalent; or the text associated with the default code point above it, or its national language equivalent. Specific defined codes and the corresponding displayed text (shown all capitalized) are listed below. Note that the codes are grouped by the high-order two hex digits; a low-order 2-digit value of X'00' represents a more general description than a non-X'00'; for this reason, the non-X'00' codes are shown indented, but any of the codes can be sent. The receiver displays the more general text (corresponding to X'**00' code points) if it does not recognize the more specific code point (e.g., because of different release schedules). The expression "(detailed data qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield or a Detailed Data Extended (X'85') subfield. The one or more necessary X'82' or X'85' subfields follow this subfield in the enclosing subvector, in the order in which they are to be associated with the gaps specified in the X'86' subfield. **Note:** The X'82' and X'85' subfields can *not* be both used in the same subvector.

Actual Actions (X'86') Resolution Common Subfield

Byte	Bit	Content

The expression "(sf83 product text)" in the English text similarly indicates a variable-length gap, to be filled in with product identification data from the Product Identifier X'11' subvector indicated by a Product Set ID Index (X'83') subfield. The necessary X'83' subfields follow the X'86' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'86' subfield.

The expression "(sf84 resource text)" in the English text indicates a variable-length gap, to be filled in with resource identification data from a location indicated by a Resource List Index (X'84') subfield. The necessary X'84' subfields follow the X'86' subfield, in the same subvector, in the order in which they are to be associated with the gaps specified in the X'86' subfield.

The third digit of each actual action code point indicates the number of succeeding detailed data qualifiers that are associated with the code point, and whether a X'83' or X'84' subfield is associated with it, as follows:

X'xx0x' – X'xx9x':	No detailed data qualifiers.
X'xxAx' – X'xxBx':	One detailed data qualifier.
X'xxCx':	Two detailed data qualifiers.
X'xxDx':	Three detailed data qualifiers.
X'xxEx':	One X'83' subfield.
X'xxFx':	One X'84' subfield.

Defined codes are:

X'0000' AUTOMATIC HARDWARE RECOVERY SUCCESSFUL

A non-manual recovery procedure corrected the problem that was caused by a failing hardware or microcode component

X'1000' MANUAL HARDWARE RECOVERY SUCCESSFUL

A manual recovery procedure corrected the problem that was caused by a failing hardware component

X'1001' FAILING COMPONENT REPLACED

X'2000' AUTOMATIC CONFIGURATION RECOVERY SUCCESSFUL

A non-manual recovery procedure corrected the problem that was caused by improper configuration of the failing component.

X'3000' MANUAL CONFIGURATION RECOVERY SUCCESSFUL

A manual recovery procedure corrected the problem that was caused by improper configuration of the failing component.

X'4000' AUTOMATIC SOFTWARE RECOVERY SUCCESSFUL

A non-manual recovery procedure corrected the problem that was caused by a failing software component

Actual Actions (X'86') Resolution Common Subfield

Byte	Bit	Content	
		X'5000'	MANUAL SOFTWARE RECOVERY SUCCESSFUL A manual recovery procedure corrected the problem that was caused by a failing software component
		X'6000'	AUTOMATIC ENVIRONMENTAL RECOVERY SUCCESSFUL A non-manual recovery procedure corrected the problem that was caused by a factor external to the failing component that contributed to the failure.
		X'7000'	MANUAL ENVIRONMENTAL RECOVERY SUCCESSFUL A manual recovery procedure corrected the problem that was caused by a factor external to the failing component that contributed to the failure.

Request Change Control (X'8050') MS Major Vector

LU → LU

This major vector is used to request that a change control function be performed.

Request Change Control (X'8050') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'8050'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Request Change Control MS Subvectors" on page 9-201 for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Request Change Control (X'8050') Major Vector	
	CP-MSU	Notes
Execution Window Timing (X'0B')	O	
Install (X'81')	CP	Note 1
Remove (X'83')	CP	Note 2
Accept (X'85')	CP	Note 3
Corequisite Change (X'87')	O(n)	Note 4

Key:

CP	Conditionally present one time (See notes for conditions.)
O	Optionally present one time
O(n)	Optionally present one or more times

Notes:

1. This subvector is used to install changes. If this subvector is present, the X'83' and X'85' subvectors are not present.
2. This subvector is used to remove changes. If this subvector is present, the X'81' and X'85' subvectors are not present.
3. This subvector is used to accept changes. If this subvector is present, the X'81' and X'83' subvectors are not present.
4. This subvector is used to name a corequisite change. It is optionally present one to six times, but if present, the X'81' subvector must be present.

Request Change Control MS Subvectors

Install (X'81') Request Change Control MS Subvector

The Install subvector requests that a change be installed, and carries relevant parameters.

Install (X'81') Request Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Install subvector
1		Key: X'81'
2−p		One or more subfields (listed by Key value below and described in detail following): X'20' Removability X'30' Activation Use X'40' Pre-Test X'50' Automatic Removal X'60' Post-Test X'70' Automatic Acceptance

Removability (X'20') Install Subfield

This subfield indicates the type of removability requested. It is present once. If the Activation Use (X'30') Install subfield specifies Trial (X'10'), then Removability Yes (X'10') must be specified in this subfield.

Removability (X'20') Install Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Removability subfield
1		Key: X'20'
2(=q)		Removability: X'10' Yes X'20' Desired X'30' No

Activation Use (X'30') Install Subfield

This subfield indicates which type of activation will cause components altered by this change to be used. It is present once.

Activation Use (X'30') Install Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Activation Use subfield
1		Key: X'30'
2(=q)		Activation Use: X'10' Trial - the altered components are used during trial activation (only), instead of production versions X'20' Production - the altered components are used during any activation unless superceded by trial versions

Pre-Test (X'40') Install Subfield

This subfield indicates the type of pre-test requested. It is present once.

Pre-Test (X'40') Install Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Pre-Test subfield
1		Key: X'40'
2(=q)		Pre-Test: X'10' Yes X'20' Desired X'30' No

Automatic Removal (X'50') Install Subfield

This subfield indicates the type of automatic removal requested. It is present once, unless removability is prohibited.

Automatic Removal (X'50') Install Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Automatic Removal subfield
1		Key: X'50'
2(=q)		Automatic Removal: X'10' Yes X'20' Desired X'30' No

Post-Test (X'60') Install Subfield

This subfield indicates the type of post-test requested. It is present once.

Post-Test (X'60') Install Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Post-Test subfield
1		Key: X'60'
2(=q)		Post-Test: X'10' Yes X'20' Desired X'30' No

Automatic Acceptance (X'70') Install Subfield

This subfield indicates the type of automatic acceptance requested. It is present once, unless removability is prohibited.

Automatic Acceptance (X'70') Install Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Automatic Acceptance subfield
1		Key: X'70'
2(=q)		Automatic Acceptance X'10' Yes X'20' Desired X'30' No

Remove (X'83') Request Change Control MS Subvector

The Remove subvector requests that a change be removed, and carries relevant parameters.

Remove (X'83') Request Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Remove subvector
1		Key: X'83'
2 − p		One subfield (listed by Key value below and described in detail following): X'60' Post-Test

Post-Test (X'60') Remove Subfield

This subfield indicates the type of post-test requested. It is present once.

Post-Test (X'60') Remove Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Post-Test subfield
1		Key: X'60'
2(= q)		Post-Test: X'10' Yes X'20' Desired X'30' No

Accept (X'85') Request Change Control MS Subvector

The Accept subvector requests that resources necessary to maintain removability of a change be relinquished (immediately).

Accept (X'85') Request Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Accept subvector
1(=p)		Key: X'85'

Corequisite Change (X'87') Request Change Control MS Subvector

This subvector identifies the SNA/File Services file name of a corequisite change (a change that must be handled as part of the same process as that required to handle the change file identified in the server object).

Corequisite Change (X'87') Request Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Corequisite Change subvector
1		Key: X'87'
2−p		An SNA/File Services file name, as defined by SNA/File Services within the registered GDS variable X'1538' (does not include the LLID for Data Object Identifier or the encapsulating LT for Token String, starts with the LT for First Identifier)

Change Control (X'0050') MS Major Vector

LU → LU

This major vector is used to return the results of a change that was requested by a focal point in a Request Change Control (X'8050') major vector, or to report locally requested changes to a focal point in an unsolicited manner.

Change Control (X'0050') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'0050'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Change Control MS Subvectors" on page 9-207 for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Change Control (X'0050') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	P	
Reporting Installation (X'82')	CP	Note 1
Reporting Removal (X'84')	CP	Note 2
Reporting Acceptance (X'86')	CP	Note 3
Reported Change Name (X'88')	CP(n)	Note 4
Reporting Secondary Installation (X'8A')	CP	Note 5
Secondary Installation Change Name (X'8C')	CP(n)	Note 6
Reporting Back-Level (X'8E')	CP	Note 7
Back-Level Change Name (X'90')	CP(n)	Note 8
Reporting Deletion (X'92')	CP	Note 9
Deleted Change Name (X'94')	CP(n)	Note 10
Detailed Data (X'98')	O(n)	

Key:

P	Present one time
CP	Conditionally present one time
CP(n)	Conditionally present more than one time (See notes for conditions.)

O(n) Optionally present more than one time

Notes:

1. This subvector is used to report installation of changes. If this subvector is present, the X'84' and X'86' subvectors are not present.
2. This subvector is used to report removal of changes. If this subvector is present, the X'82' and X'86' subvectors are not present.
3. This subvector is used to report that a change was accepted. If this subvector is present, the X'82' and X'84' subvectors are not present.
4. This subvector is conditionally present one to seven times. At least one is present if one of these subvectors is present: X'82', X'84', or X'86'. An instance of this subvector is present for each change referred to in the request.
5. This subvector is present if a change neither referred to in the request nor one of its corequisites was installed as part of the operation being reported. If it is present, then one or more Secondary Installation Change Name (X'8C') subvectors are present.
6. This subvector is conditionally present one to seven times. An instance of this subvector is present for each change not referred to in the request, but installed as part of the operation being reported.
7. This subvector is present if a change not referred to in the request was put into back-level state as part of the operation being reported. If it is present, then one or more Back-Level Change Name (X'90') subvectors are present.
8. This subvector is conditionally present one to seven times. An instance of this subvector is present for each change not referred to in the request, but put into back-level state as part of the operation being reported.
9. This subvector is present if a change not referred to in the request was deleted as part of the operation being reported. If it is present, then one or more Deleted Change Name (X'94') subvectors are present.
10. This subvector is conditionally present one to seven times. An instance of this subvector is present for each change not referred to in the request, but deleted as part of the operation being reported.

Change Control MS Subvectors

Reporting Installation (X'82') Change Control MS Subvector

The Reporting Installation subvector reports the results of an Install request.

Reporting Installation (X'82') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Installation subvector
1		Key: X'82'

Reporting Installation (X'82') Change Control MS Subvector

Byte	Bit	Content
2−p		One or more subfields (listed by Key value below and described in detail following): X'10' Installation Status X'20' Removability Status X'30' Activation Use Status X'40' Pre-Test Status X'50' Automatic Removal Status X'60' Post-Test Status X'70' Automatic Acceptance Status

Installation Status (X'10') Reporting Installation Subfield

This subfield reports the results of an install. It is always present once.

Installation Status (X'10') Reporting Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Installation Status subfield
1		Key: X'10'
2		Status: X'10' Successful X'20' Attempted, but not successful -- recovery to previous state was effected X'28' Attempted, but not successful -- recovery to previous state was not effected X'30' Not attempted and will not attempt X'40' Will attempt
3(=q)		When effective: X'10' Changed components are now in use X'20' Components are changed, but activation is required X'30' Not applicable (because install not attempted)

Removability Status (X'20') Reporting Installation Subfield

This subfield reports the removability status. It is present once.

Removability Status (X'20') Reporting Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Removability Status subfield
1		Key: X'20'
2(=q)		Status: X'10' Installed removably X'20' Installed, but not removably X'30' Not installed

Activation Use Status (X'30') Reporting Installation Subfield

This subfield indicates which type of activation will cause components altered by this change to be used. It is present once.

Activation Use Status (X'30') Reporting Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Activation Use subfield
1		Key: X'30'
2(=q)		Activation use: X'10' Trial X'20' Production X'30' Installation was unsuccessful

Pre-Test Status (X'40') Reporting Installation Subfield

This subfield reports the results of a pre-test. It is present once if a pre-test was required or desired.

Pre-Test Status (X'40') Reporting Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Pre-Test Status subfield
1		Key: X'40'

Pre-Test Status (X'40') Reporting Installation Subfield

Byte	Bit	Content
2(=q)		Status: X'10' Successful X'20' Not successful X'30' Not attempted

Automatic Removal Status (X'50') Reporting Installation Subfield

This subfield reports the results of an automatic removal. It is present once if automatic removal was required or desired.

Automatic Removal Status (X'50') Reporting Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Automatic Removal Status subfield
1		Key: X'50'
2		Status: X'10' Successful X'20' Attempted, but not successful -- recovery to previous state was effected X'28' Attempted, but not successful -- recovery to previous state was not effected X'30' Not attempted
3(=q)		When effective: X'10' Changed components are now in use X'20' Components are changed, but activation is required X'30' Not applicable (because automatic removal was not attempted)

Post-Test Status (X'60') Reporting Installation Subfield

This subfield reports the results of a post-test. It is present once if a post-test was required or desired.

Post-Test Status (X'60') Reporting Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Post-Test Status subfield
1		Key: X'60'

Post-Test Status (X'60') Reporting Installation Subfield

Byte	Bit	Content
2(=q)		Status: X'10' Successful X'20' Not successful X'30' Not attempted

Automatic Acceptance Status (X'70') Reporting Installation Subfield

This subfield reports the results of an automatic acceptance. It is present once if automatic acceptance was required or desired.

Automatic Acceptance Status (X'70') Reporting Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Automatic Acceptance Status subfield
1		Key: X'70'
2(=q)		Status: X'10' Successful X'20' Not successful X'30' Not attempted and will not attempt X'40' Will attempt at the end of the delay specified

Reporting Removal (X'84') Change Control MS Subvector

The Reporting Removal subvector reports the results of a Remove request.

Reporting Removal (X'84') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Removal subvector
1		Key: X'84'
2−p		One or more subfields (listed by Key value below and described in detail following): X'10' Removal Status X'60' Post-Test Status

Removal Status (X'10') Reporting Removal Subfield

This subfield reports the results of the removal. It is always present once.

Removal Status (X'10') Reporting Removal Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Removal Status subfield
1		Key: X'10'
2		Status: X'10' Successful X'20' Attempted, but not successful -- recovery to previous state was effected X'28' Attempted, but not successful -- recovery to previous state was not effected X'30' Not attempted and will not attempt X'40' Will attempt
3(=q)		When effective: X'10' Changed components are now in use X'20' Components are changed, but activation is required X'30' Not applicable (because remove not attempted)

Post-Test Status (X'60') Reporting Removal Subfield

This subfield reports the results of a post-test. It is present once if a post-test was required or desired.

Post-Test Status (X'60') Reporting Removal Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Post-Test Status subfield
1		Key: X'60'
2(=q)		Status: X'10' Successful X'20' Not successful X'30' Not attempted

Reporting Acceptance (X'86') Change Control MS Subvector

This subvector reports the results of an Accept request.

Reporting Acceptance (X'86') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Acceptance subvector
1		Key: X'86'
2−p		One subfield (listed by Key value below and described in detail following): X'10' Accept Status

Accept Status (X'10') Reporting Acceptance Subfield

This subfield reports the results of an accept. It is always present once.

Accept Status (X'10') Reporting Acceptance Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Accept Status subfield
1		Key: X'10'
2(=q)		Status: X'10' Successful X'20' Attempted, but not successful X'30' Not attempted and will not attempt X'40' Will attempt

Reported Change Name (X'88') Change Control MS Subvector

This subvector identifies the SNA/File Services file name of the change file that is being reported on.

Reported Change Name (X'88') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reported Change Name subvector
1		Key: X'88'
2−p		An SNA/File Services file name, as defined by SNA/File Services within the registered GDS variable X'1538' (does not include the LLID for Data Object Identifier or the encapsulating LT for Token String, starts with the LT for First Identifier)

Reporting Secondary Installation (X'8A') Change Control MS Subvector

The Reporting Secondary Installation subvector reports installation of a change that resulted from a request referring to a different change.

Reporting Secondary Installation (X'8A') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Secondary Installation subvector
1		Key: X'8A'
2–p		One or more subfields (listed by Key value below and described in detail following): X'10' Installation Status X'30' Activation Use Status

Installation Status (X'10') Reporting Secondary Installation Subfield

This subfield reports the results of an install. It is always present once.

Installation Status (X'10') Reporting Secondary Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Installation Status subfield
1		Key: X'10'
2		Status: X'10' Successful
3(=q)		When effective: X'10' Changed components are now in use X'20' Components are changed, but activation is required

Activation Use Status (X'30') Reporting Secondary Installation Subfield

This subfield indicates which type of activation will cause components altered by this change to be used. It is always present once.

Activation Use Status (X'30') Reporting Secondary Installation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Activation Use subfield
1		Key: X'30'
2(=q)		Activation Use: X'10' Trial X'20' Production

Secondary Installation Change Name (X'8C') Change Control MS Subvector

This subvector identifies the SNA/File Services file name of the change file that is being reported on.

Secondary Installation Change Name (X'8C') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Secondary Installation Change Name subvector
1		Key: X'8C'
2−p		An SNA/File Services file name, as defined by SNA/File Services within the Data Object Global Name (X'1538') GDS variable (does not include the LLID for Data Object Identifier or the encapsulating LT for Token String, starts with the LT for First Identifier)

Reporting Back-Level (X'8E') Change Control MS Subvector

The Reporting Back-Level subvector reports that a change was put in back-level state as the result of a request referring to a different change.

Reporting Back-Level (X'8E') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Back-Level subvector
1		Key: X'8E'
2−p		One subfield (listed by Key value below and described in detail following): X'10' Back-Level

Back-Level (X'10') Reporting Back-Level Subfield

This subfield reports the back-level. It is always present once.

Back-Level (X'10') Reporting Back-Level Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Back-Level subfield
1(=q)		Key: X'10'

Back-Level Change Name (X'90') Change Control MS Subvector

This subvector identifies the SNA/File Services file name of the change file that is being reported on.

Back-Level Change Name (X'90') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Back-Level Change Name subvector
1		Key: X'90'
2−p		An SNA/File Services file name, as defined by SNA/File Services within the Data Object Global Name (X'1538') GDS variable (does not include the LLID for Data Object Identifier or the encapsulating LT for Token String, starts with the LT for First Identifier)

Reporting Deletion (X'92') Change Control MS Subvector

The Reporting Deletion subvector reports that a change was deleted as the result of a request referring to a different change.

Reporting Deletion (X'92') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Deletion subvector
1		Key: X'92'

Reporting Deletion (X'92') Change Control MS Subvector

Byte	Bit	Content
2−p		One subfield (listed by Key value below and described in detail following): X'10' Deletion

Deletion (X'10') Reporting Deletion Subfield

This subfield reports the deletion. It is always present once.

Deletion (X'10') Reporting Deletion Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Deletion subfield
1(=q)		Key: X'10'

Deleted Change Name (X'94') Change Control MS Subvector

This subvector identifies the SNA/File Services file name of the change file that is being reported on.

Deleted Change Name (X'94') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Deleted Change Name subvector
1		Key: X'94'
2−p		An SNA/File Services file name, as defined by SNA/File Services within the Data Object Global Name (X'1538') GDS variable (does not include the LLID for Data Object Identifier or the encapsulating LT for Token String, starts with the LT for First Identifier)

Change Object Class (X'96') Change Control MS Subvector

This subvector identifies the SNA/File Services class of the change objects being reported on. It applies to reports on change objects identified in the Reported Change Name (X'88') and any secondary effects (X'8C', X'90', X'94') subvectors.

Change Object Class (X'96') Change Control MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Change Object Class subvector
1		Key: X'96'
2 – 5(= p)		An SNA/File Services data object class, as defined by SNA/File Services within the Data Object Attributes (X'1537') GDS variable (does not include the LLID for Data Object Acceptance or the encapsulating LT for Data Object Class)

Detailed Data (X'98') Change Control MS Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector.

Execute Command (X'8061') MS Major Vector

SSCP → PU, CP → CP

This major vector requests that the message associated with it be interpreted and executed as a command.

Execute Command (X'8061') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'8061'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F'. When the Self Defining Text Message (X'31') subvector is not present in this MS major vector, this MS major vector is followed by one of the following management services parameter major vectors: X'1300' Text Data X'1307' Structured Data X'1309' Transparent Coded Datastream *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Execute Command (X'8061') Major Vector		
	CP-MSU	NMVT	Notes
Name List (X'06')	P	P	
Self-Defining Text Msg (X'31')	CP	CP	

Key:

P Present one time
CP Conditionally present one time (See Note for conditions.)

Note:

1. This subvector is present when the command to be executed is not contained in a parameter major vector. When it is present, no MS parameter major vectors follow the Execute Command (X'8061') MS major vector.

Reply to Execute Command (X'0061') MS Major Vector

PU → SSCP, CP → CP

This major vector transports the reply provided in response to a previous Execute Command. It is followed by a management services parameter major vector except when it returns sense data.

Reply to Execute Command (X'0061') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'0061'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F'.

When the Sense Data (X'7D') subvector is not present, this major vector is followed by one of the following management services parameter major vectors:

X'1300' Text Data
X'1307' Structured Data
X'1309' Transparent Coded Datastream

Note: The following subvector keys may be used as indicated:

Subvector	Presence in Reply to Execute Command (X'0061') Major Vector		
	NMVT	CP-MSU	Notes
Sense Data (X'7D')	CP	CP	Note 1
NMVT Count (X'44')	CP	—	Note 2

Key:

— Not present.
CP Conditionally present one time (See Notes for conditions.)

Notes:

1. This subvector is present only when sense data is returned to the requesting application. When it is present, no other subvectors are present and no MS parameter major vectors follow the Reply to Execute Command (X'0061') MS major vector.
2. This subvector is present only when a single reply to a command is being returned to the requesting application and the reply data spans multiple NMVTs. It appears in the first NMVT, which always contains the Reply to Execute Command (X'0061') MS major vector.

Analyze Status (X'8062') MS Major Vector

SSCP → PU, CP → CP

This major vector requests the gathering of information about one or more listed resources, analysis of that information, and the return of the result in a reply that reports the joint state of all indicated resources.

Analyze Status (X'8062') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'8062'
4–n		MS subvector, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F'. *Note:* The following subvector key is used as indicated:

Subvector	Presence in Analyze Status (X'8062') Major Vector		
	NMVT	CP-MSU	Notes
Name List (X'06')	P	P	

Key:

P Present one time

Reply to Analyze Status (X'0062') MS Major Vector

PU → SSCP, CP → CP

This major vector transports the reply to a previous Analyze Status request. It is followed by management services parameter major vectors except when it is used to return sense data.

Reply to Analyze Status (X'0062') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0062'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F'.

When the Sense Data (X'7D') subvector is not present, this major vector is followed by the following management services parameter major vectors. The Begin Data Parameters (X'130A') is used to begin the set, one Structured Data (X'1307') is present for each resource included in the report, and the set is terminated with the End Parameter Data (X'130B').

X'130A' Begin Data Parameters
X'1307' Structured Data (zero or more)
X'130B' End Parameter Data
Note: The following subvector keys may be used as indicated:

Subvector	Presence in Reply to Analyze Status (X'0062') Major Vector		
	NMVT	CP-MSU	Notes
Sense Data (X'7D')	CP	CP	Note 1
NMVT Count (X'44')	CP	–	Note 2

Key:

– Not present.
CP Conditionally present one time (See Notes for conditions.)

Notes:

1. This subvector is present only when sense data is returned to the requesting application. When it is present, no other subvectors are present and no MS parameter major vectors follow the Reply to Analyze Status (X'0062') MS major vector.
2. This subvector is present only when a single reply to a request is being returned to the requesting application and the reply data spans multiple NMVTs. It

appears in the first NMVT, which always contains the Reply to Analyze Status (X'0062') MS major vector.

Query Resource Data (X'8063') MS Major Vector

SSCP → PU, CP → CP

This major vector requests the gathering of information from one or more resources and reporting of that information in a reply.

Query Resource Data (X'8063') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'8063'
4–n		MS subvector, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F'. *Note:* The following subvector key is used as indicated:

Subvector	Presence in Query Resource Data (X'8063') Major Vector		
	NMVT	CP-MSU	Notes
Name List (X'06')	P	P	

Key:

P Present one time

Reply to Query Resource Data (X'0063') MS Major Vector

PU → SSCP, CP → CP

This major vector transports the reply to a previous Query Resource Data request. It is followed by management services parameter major vectors except when it is used to return sense data.

Reply to Query Resource Data (X'0063') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0063'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F'.

When the Sense Data (X'7D') subvector is not present, this major vector is followed by the following management services parameter major vectors. The Begin Data Parameters (X'130A') is used to begin the set, one Structured Data (X'1307') is present for each resource included in the report, and the set is terminated with the End Parameter Data (X'130B').

X'130A' Begin Data Parameters
X'1307' Structured Data (one or more)
X'130B' End Parameter Data

At least one Structured Data (X'1307') major vector must be present between the X'130A' and X'130B' major vectors.
Note: The following subvector keys may be used as indicated:

Subvector	Presence in Reply to Query Resource Data (X'0063') Major Vector		
	NMVT	CP-MSU	Notes
Sense Data (X'7D')	CP	CP	Note 1
NMVT Count (X'44')	CP	—	Note 2

Key:

— Not present.
CP Conditionally present one time (See Notes for conditions.)

Notes:

1. This subvector is present only when sense data is returned to the requesting application. When it is present, no other subvectors are present and no MS

parameter major vectors follow the Reply to Query Resource Data (X'0063') MS major vector.

2. This subvector is present only when a single reply to a request is being returned to the requesting application and the reply data spans multiple NMVTs. It appears in the first NMVT, which always contains the Reply to Query Resource Data (X'0063') MS major vector.

Test Resource (X'8064') MS Major Vector

SSCP → PU, CP → CP

This major vector requests the testing of one or more resources, the gathering of information from the test and provision of the results as a reply which reports the state of each resource.

Test Resource (X'8064') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'8063'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Test Resource Subvectors" for subvector keys X'80'–X'FF'. *Note:* The following subvector keys are used as indicated:

Subvector	Presence in Test Resource (X'8064') Major Vector		
	NMVT	CP-MSU	Notes
Name List (X'06')	P	P	
Test Setup Data (X'80')	P	P	

Key:

P Present one time

Test Resource Subvectors

Test Setup Data (X'80') Test Resource MS Subvector

This Test Resource subvector transports the details of the requested test to be performed.

Test Setup Data (X'80') Test Resource MS Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the Test Setup Data subvector
1		Key: X'80'

Test Setup Data (X'80') Test Resource MS Subvector

Byte	Bit	Content
2−p		One subfield containing the number of times the test is to be executed *Note:* The following subfield key is used as described in detail following: X'01' Test Request Count

Test Request Count (X'01') Test Setup Data Subfield

This subfield transports the count of iterations requested for the test.

Test Request Count (X'01') Test Setup Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Test Request Count subfield
1		Key: X'01'
2−3(=q)		Test request count: an integer value from 1 to 32727. The receiver is requested to repeat the identified test this many times or until a failure is detected.

Reply to Test Resource (X'0064') MS Major Vector

PU → SSCP, CP → CP

This major vector transports the reply to a previous Test Resource request. It is followed by management services parameter major vectors except when it returns sense data.

Reply to Test Resource (X'0064') MS Major Vector

Byte	Bit	Content
0−1		Length (n + 1), in binary, of this MS major vector
2−3		Key: X'0064'

Reply to Test Resource (X'0064') MS Major Vector

Byte	Bit	Content
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F', and in "Reply to Test Resource Subvectors" on page 9-230 for subvector keys X'80' - X'FF'. When the Sense Data (X'7D') subvector is not present, this major vector is followed by the following management services parameter major vectors. The Begin Data Parameters (X'130A') is used to begin the set, one Structured Data (X'1307') is present for each resource included in the report, and the set is terminated with the End Parameter Data (X'130B'). X'130A' Begin Data Parameters X'1307' Structured Data (0 or more) X'130B' End Parameter Data *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Reply to Test Resource (X'0064') Major Vector		
	NMVT	CP-MSU	Notes
Sense Data (X'7D')	CP	CP	Note 1
Test Result Data (X'81')	CP	CP	Note 2
NMVT Count (X'44')	CP	—	Note 3

Key:

— Not present.
CP Conditionally present one time (See Notes for conditions.)

Notes:

1. This subvector is present only when sense data is returned to the requesting application. When it is present, no other subvectors are present and no MS parameter major vectors follow the Reply to Test Resource (X'0064') MS major vector.
2. This subvector is present only when the receiving application has executed the specified test.
3. This subvector is present only when a single reply to a request is being returned to the requesting application and the reply data spans multiple NMVTs. It appears in the first NMVT, which always contains the Reply to Test Resource (X'0064') MS major vector.

Reply to Test Resource Subvectors

Test Result Data (X'81') Reply to Test Resource MS Subvector

This subvector transports the results of a requested test that was performed.

Test Result Data (X'81') Reply to Test Resource MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Test Result Data subvector
1		Key: X'81'
2−p		Subfields containing the test type and test results *Note:* The following subfield keys are used as described in detail following: X'01' Execution Result X'02' Test Type X'03' Test Request Count X'04' Test Executed Count

Test Execution Result (X'01') Test Result Data Subfield

This subfield transports the result of the requested test.

Test Execution Result (X'01') Test Result Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Test Execution Result subfield
1		Key: X'01'
2(=q)		Test execution result: X'00' no errors detected X'01' errors detected X'02' indeterminate results

Test Type (X'02') Test Result Data Subfield

This subfield transports the type of test requested.

Test Type (X'02') Test Result Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Test Type subfield
1		Key: X'02'
2(=q)		Test type: a code that indicates which type of test is requested. X'00' background self-test (a test of the resource is to be scheduled at the first opportunity that will be nondisruptive to normal operation). X'01' immediate self-test (the resource is to be tested immediately even if such action will be disruptive).

Test Request Count (X'03') Test Result Data Subfield

This subfield transports the count of iterations requested for the test.

Test Request Count (X'03') Test Result Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Test Request Count subfield
1		Key: X'03'
2−3(=q)		Test request count: an integer value from 1 to 32727. This returns the value contained in the matching request.

Test Executed Count (X'04') Test Result Data Subfield

This subfield transports the count of iterations executed for the test.

Test Executed Count (X'04') Test Result Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Test Executed Count subfield
1		Key: X'04'
2−3(=q)		Test executed count: an integer value from 1 to 32727. This returns the number of executions of the test attempted before failure occured. If no failure occured it is the same value as Test Request Count.

Request Activation (X'8066') MS Major Vector

LU → LU, CP → CP

This major vector is used to request that an activation procedure be performed.

Request Activation (X'8066') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'8066'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' – X'7F', and in "Request Activation MS Subvectors" for subvector keys X'80' – X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Request Activation (X'8066') Major Vector	
	CP-MSU	Notes
Configuration Name (X'07')	O	
Reporting Level (X'09')	O	
Execution Window Timing (X'0B')	O	
Activate (X'81')	P	

Key:

P Present one time
O Optionally present one time

Request Activation MS Subvectors

Activate (X'81') Request Activation MS Subvector

The Activate subvector requests MS to cause reactivation of the node in which its LU resides. For example, an "initial microprogram load (IML)" of the node containing the LU may be performed.

Activate (X'81') Request Activation MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Activate subvector
1		Key: X'81'
2 − p		One or more subfields (listed by Key value below and described in detail following): X'10' Force Indication X'20' Change Management Activation Use

Force Indication (X'10') Activate Subfield

This subfield indicates whether to perform the activation based on the quiesced state of the target node. It is always present.

Force Indication (X'10') Activate Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Force Indication subfield
1		Key: X'10'
2		Force indicator: X'10' No (do not force) - reject if not quiesced, otherwise activate X'20' Yes (force) - activate even if not quiesced
3 − 5(=q)		Delay: optional binary number (bytes 3 and 4) and units (byte 5) of minutes, seconds or hours (X'00', X'01' or X'02' respectively) representing the maximum allowed wait time to quiesce (if not already quiesced) before either rejecting or forcing activation. If a zero value is specified or Delay is absent (2=q), immediate action is requested.

Change Management Activation Use (X'20') Activate Subfield

This subfield indicates which installed changes to activate. It is present once, unless the target node is known not to support this subfield in particular, or the change management architecture in general.

Change Management Activation Use (X'20') Activate Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Change Management Activation Use subfield

Change Management Activation Use (X'20') Activate Subfield

Byte	Bit	Content
1		Key: X'20'
2(=q)		Activation use: X'10' Trial and production - use changes that are installed on a trial basis before using changes installed in production X'20' Production only - use changes that are installed in production only X'30' Last used - use whatever was previously set, Trial and production or Production only. If the entry pont does not support change management activation use, then "last used" is effectively treated as "production only" and the command report to the focal point includes sense code X'080C 0015'.

Activation Acceptance (X'0066') MS Major Vector

LU → LU, CP → CP

This major vector is used to reply to Request Activation (X'8066'), to indicate initial acceptance or rejection of the request. After activation is successful, an entry point notifies its focal point using Activation (X'0067').

Activation Acceptance (X'0066') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0066'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Activation Acceptance MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Activation Acceptance (X'0066') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	P	
Activation Acceptance (X'82')	P	
Detailed Data (X'98')	O(n)	

Key:

P Present one time
O(n) Optionally present one or more times

Activation Acceptance MS Subvectors

Activation Acceptance (X'82') Activation Acceptance MS Subvector

The Activation Acceptance subvector reports whether or not an activation will be attempted as requested.

Activation Acceptance (X'82') Activation Acceptance MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Activation Acceptance subvector
1		Key: X'82'
2–p		One or more subfields (listed by Key value below and described in detail following): X'10' Attempt Status

Attempt Status (X'10') Activation Acceptance Subfield

This subfield reports whether or not activation will be attempted as requested. It is always present once.

Attempt Status (X'10') Activation Acceptance Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Attempt Status subfield
1		Key: X'10'
2(=q)		Acceptance: X'10' Will attempt X'20' Will not attempt

Detailed Data (X'98') Activation Acceptance MS Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Activation (X'0067') MS Major Vector

LU → LU, CP → CP

This major vector is used by an entry point to return the results of an activation to a focal point. Activation may have been requested either by the focal point or locally.

Activation (X'0067') MS Major Vector

Byte	Bit	Content
0−1		Length (n + 1), in binary, of this MS major vector
2−3		Key: X'0067'
4−n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'−X'7F', and in "Activation MS Subvectors" for subvector keys X'80'−X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Activation (X'0067') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	P	
Reporting Activation (X'82')	P	
Detailed Data (X'98')	O(n)	

Key:

P	Present one time
O(n)	Optionally present one or more times

Activation MS Subvectors

Reporting Activation (X'82') Activation MS Subvector

The Reporting Activation subvector indicates whether or not activation was successful.

Reporting Activation (X'82') Activation MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Activation subvector

Reporting Activation (X'82') Activation MS Subvector

Byte	Bit	Content
1		Key: X'82'
2–p		One or more subfields (listed by Key value below and described in detail following): X'10' Activation Status X'18' Activation Stage X'20' Change Management Activation Use Status

Activation Status (X'10') Reporting Activation Subfield

This subfield reports whether or not activation was successful. It is always present once.

Activation Status (X'10') Reporting Activation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Activation Status subfield
1		Key: X'10'
2(=q)		Activation status: X'10' Successful X'20' Not successful X'50' Intermediate. The activation is not yet complete, but information concerning the activation process is available and is being reported.

Activation Stage (X'18') Reporting Activation Subfield

This subfield indicates what stage of entry point activation has been achieved. It is present when the activation status is successful, with the following exception for implementations that do not build this subfield: If this subfield is not present, then the target of the activation request is assumed to be successfully activated in compliance with the parameters supplied (or optionally referenced) in the activation request.

Activation Stage (X'18') Reporting Activation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Activation Stage subfield
1		Key: X'18'

Activation Stage (X'18') Reporting Activation Subfield

Byte	Bit	Content
2(=q)		Activation Stage: X'00' Initial connectivity complete X'10' Target hardware active, target software not yet active X'20' Target application function has reached a point where quiesce of that function is required prior to any disruptive command execution X'30' Target has been successfully activated in compliance with the parameters supplied (or optionally referenced) in the activation request

Change Management Activation Use Status (X'20') Reporting Activation Subfield

This subfield reports whether trial components were searched during activation and optionally, whether a special activation-related condition prevails at the reporting node. Alternately, it may indicate that change management or the change management activation use function are not supported at the reporting node. It is present when the activation status is successful.

Change Management Activation Use Status (X'20') Reporting Activation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Change Management Activation Use Status subfield
1		Key: X'20'
2		Activation use: X'10' Trial and production: trial components were searched and used if found X'20' Production only: trial components were not searched X'30' Not supported: change management or the change management activation use function are not supported X'40' The components searched were those searched in the last activation. Both trial and production components were searched. X'50' The components searched were those searched in the last activation. Only production components were searched.
3(=q)		Special condition: optional indicator specifying whether a special activation-related condition prevails at the reporting node. If absent (2=q), no special condition is being reported. X'10' One of the following special conditions exists at the reporting node: • When the CMAU parameter in the request was set to 'trial and production' (or set to 'last', with the previous activation having specified 'trial and production'), only production components were found despite having searched for trial and production components. • The original request execution failed. The CMAU parameter in the original request was set to 'trial and production' (or set to 'last', with the previous activation having specified 'trial and production'). However, a local activation has been performed, using only production components, thus overriding the failed remote trial activation.

Detailed Data (X'98') Activation MS Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Request Initiation (X'8068') MS Major Vector

LU → LU

This major vector is used to request the target node initiate a command procedure (also referred to as a job element). It is followed by a management services parameter major vector.

Request Initiation (X'8068') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'8068'
4–n		MS subvectors, as described (using zero-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' – X'7F'.
		This major vector is followed by the Initiate Agent Request (X'1730') management services parameter major vector.
		Note: The following subvector keys may be used as indicated:

Subvector	Presence in Request Initiation (X'8068') Major Vector	
	CP-MSU	Notes
Execution Window Timing (X'0B')	O	

Key:

O Optionally present one time

Initiation (X'0068') MS Major Vector

LU → LU

This major vector is used by an entry point to return the results of an initiation to a focal point. It is followed by a management services parameter major vector.

Initiation (X'0068') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0068'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F'. This major vector is followed by the Initiate Agent Report (X'1731') management services parameter major vector. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Initiation (X'0068') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	P	
Detailed Data (X'98')	O(n)	

Key:

P	Present one time
O(n)	Optionally present one or more times

Initiation MS Subvectors

Detailed Data (X'98') Initiation MS Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Send Message to Operator (X'006F') MS Major Vector

PU → SSCP, CP → CP

This major vector sends an unsolicited request to the host operator named. It is followed by a management services parameter major vector.

Send Message to Operator (X'006F') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'006F'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F'.
		This major vector is followed by one of the following management services parameter major vectors:
		X'1300' Text Data X'1307' Structured Data X'1309' Transparent Coded Datastream
		Note: The following subvector keys may be used as indicated:

Subvector	Presence in Send Message to Operator (X'006F') Major Vector		
	NMVT	CP-MSU	Notes
Name List (X'06')	P	P	

Key:

P Present one time

Operate (X'8070') MS Major Vector

CP → CP

This major vector is used to indicate that an implementation-specific command, within the same CP-MSU as the Operate major vector, is present. The Operate major vector carries information relative to node configuration, reporting requirements, and execution timing.

Operate (X'8070') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'8070'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Operate MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Operate (X'8070') Major Vector	
	CP-MSU	Notes
Configuration Name (X'07')	O	
Reporting Level (X'09')	O	
Execution Window Timing (X'0B')	O	
Operate Command (X'81')	P	

Key:

P Present one time
O Optionally present one time

Operate MS Subvectors

Operate Command (X'81') Operate MS Subvector

The presence of the Operate Command subvector indicates to the target that implementation-specific commands (not interpreted but transported by MS) are included in GDS variables within the same CP-MSU.

Operate Command (X'81') Operate MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Operate Command subvector
1		Key: X'81'
2 – p		One or more subfields (listed by Key value below and described in detail following): X'10' Force Indication X'20' Change Management Activation Use *Note:* The following subfield keys may be used as indicated:

Subfield	Presence in Operate Command (X'81') Operate MS Subvector	
Force Indication (X'10')	O	
Change Management Activation Use (X'20')	O	

Key:

O Optionally present one time

Force Indication (X'10') Operate Command Subfield

This subfield indicates whether to perform the operate command based on the quiesced state of the target node. If not present, then Force(No) is implied and the command is executed only if the target is quiesced.

Force Indication (X'10') Operate Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Force Indication subfield
1		Key: X'10'
2		Force indicator: X'10' No (do not force) - reject if not quiesced, otherwise perform the operate command X'20' Yes (force) - perform the operate command even if not quiesced
3 – 5(=q)		Delay: optional binary number (bytes 3 and 4) and units (byte 5) of minutes, seconds or hours (X'00', X'01' or X'02' respectively) representing the maximum allowed wait time to quiesce (if not already quiesced) before either rejecting or forcing the operate command. If a zero value is specified or Delay is absent (2=q), immediate action is requested.

Change Management Activation Use (X'20') Operate Command Subfield

This subfield indicates which installed changes to activate. command contains a product-specific activation command. It is present once, unless the target node is known not to support this subfield in particular, or the change management architecture in general, or the command is not a product-specific activation command.

Change Management Activation Use (X'20') Operate Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Change Management Activation Use subfield
1		Key: X'20'
2(=q)		Activation use: X'10' Trial and production - use changes that are installed on a trial basis before using changes installed in production X'20' Production only - use changes that are installed in production only X'30' Last used - use whatever was previously set, Trial and production or Production only. If the entry pont does not support change management activation use, then "last used" is effectively treated as "production only" and the command report to the focal point includes sense code X'080C 0015'.

Operate Report (X'0070') MS Major Vector

CP → CP

This major vector is used to indicate the acceptance, rejection, success, or failure of an Operate command.

Operate Report (X'0070') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0070'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Operate Report MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Operate Report (X'0070') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	O	
Default Character Set ID (X'32')	O	
Reporting Operation (X'82')	P	
Detailed Data (X'98')	O(n)	

Key:

P	Present one time
O	Optionally present one time
O(n)	Optionally present one or more times

Operate Report MS Subvectors

Reporting Operation (X'82') Operate Report MS Subvector

The Reporting Operation subvector indicates the status of an Operate request.

Reporting Operation (X'82') Operate Report MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Operation subvector
1		Key: X'82'
2–p		One or more subfields (listed by Key value below and described in detail following): X'10' Operation Status X'20' Change Management Activation Use Status *Note:* The following subfield keys may be used as indicated:

Subfield	Presence in Reporting Operation (X'82') Operate Report MS Subvector	
Operation Status (X'10')	P	
Change Management Activation Use Status (X'20')	O	

Key:

O Optionally present one time
P Present one time

Operation Status (X'10') Reporting Operation Subfield

This subfield indicates whether or not an operate request was successful or not, accepted or not, or whether the command is proceeding but not yet complete (intermediate). Additionally, an ancillary status may be reported. An ancillary report is sent after a request has finished executing, the execution having caused the sender to enter or exit some state that normally causes the creation of one or more ancillary reports. The contents of the reports are implementation-specific, and are contained in GDS variables within the Operate Report (X'0070') CP-MSU.

Operation Status (X'10') Reporting Operation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Operation Status subfield
1		Key: X'10'
2(=q)		Operation Request Status: X'10' Successful X'20' Unsuccessful

Operation Status (X'10') Reporting Operation Subfield

Byte	Bit	Content
		X'30' Rejected
		X'40' Accepted
		X'50' Intermediate. The execution is not yet complete, but information concerning the operation process is available and is being reported.
		X'60' Ancillary. An application program has information concerning the state it has entered after the operate command has been successfully executed.

Change Management Activation Use Status (X'20') Reporting Operation Subfield

This subfield is defined under the Activation (X'0067') major vector.

Detailed Data (X'98') Reporting Operation Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Request Deactivation (X'8071') MS Major Vector

CP → CP

This major vector is used to request the target to perform a deactivation procedure.

Request Deactivation (X'8071') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'8071'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Request Deactivation MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Request Deactivation (X'8071') Major Vector	
	CP-MSU	Notes
Configuration Name (X'07')	O	
Reporting Level (X'09')	O	
Execute Window Timing (X'0B')	O	
Deactivate Command (X'81')	P	

Key:

P Present one time
O Optionally present one time

Request Deactivation MS Subvectors

Deactivate Command (X'81') Request Deactivation Subvector

The Deactivate Command subvector requests the target to deactivate one more network resources.

Deactivate Command (X'81') Request Deactivation Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Deactivate Command subvector
1		Key: X'81'
2−p		One subfield (optionally present one time and listed by Key value below): X'10' Force Indication

Force Indication (X'10') Deactivate Command Subfield

This subfield indicates whether to perform the deactivation based on the quiesced state of the target node. If not present, Force(NO) is implied and the command is executed only if the target is quiesced.

Force Indication (X'10') Deactivate Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Force Indication subfield
1		Key: X'10'
2		Force indicator: X'10' No (do not force) - reject if not quiesced, otherwise deactivate X'20' Yes (force) - deactivate even if not quiesced
3−5(=q)		Delay: optional binary number (bytes 3 and 4) and units (byte 5) of minutes, seconds or hours (X'00', X'01' or X'02' respectively) representing the maximum allowed wait time to quiesce (if not already quiesced) before either rejecting or forcing deactivation. If a zero value is specified or Delay is absent (2=q), immediate action is requested.

Deactivation Acceptance (X'0071') MS Major Vector

CP → CP

This major vector is used to indicate the acceptance or rejection of a Deactivation command.

Deactivation Acceptance (X'0071') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0071'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Deactivation Acceptance MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Deactivation Acceptance (X'0071') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	P	
Deactivation Acceptance (X'82')	P	
Detailed Data (X'98')	O(n)	

Key:

P	Present one time
O(n)	Optionally present one or more times

Deactivation Acceptance MS Subvectors

Deactivation Acceptance (X'82') Deactivation Acceptance Subvector

The Deactivation Acceptance subvector indicates whether or not a deactivation will be attempted as requested.

Deactivation Acceptance (X'82') Deactivation Acceptance Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the Deactivation Acceptance subvector

Deactivation Acceptance (X'82') Deactivation Acceptance Subvector

Byte	Bit	Content
1		Key: X'82'
2–p		One required subfield (listed by Key value below and described in detail following): X'10' Attempt Status

Attempt Status (X'10') Deactivation Acceptance Subfield

This subfield indicates whether or not a deactivation will be attempted as requested.

Attempt Status (X'10') Deactivation Acceptance Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Attempt Status subfield
1		Key: X'10'
2(=q)		Acceptance: X'10' Will attempt X'20' Will not attempt

Detailed Data (X'98') Deactivation Acceptance Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Deactivation (X'0072') MS Major Vector

CP → CP

This major vector is used to indicate success or failure in the execution of a Deactivation command.

Deactivation (X'0072') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vect
2–3		Key: X'0072'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Deactivation MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Deactivation (X'0072') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	P	
Reporting Deactivation (X'82')	P	
Detailed Data (X'98')	O(n)	

Key:

P Present one time
O(n) Optionally present one or more times

Deactivation MS Subvectors

Reporting Deactivation (X'82') Deactivation Subvector

The Reporting Deactivation subvector indicates whether or not a deactivation was successful. If the deactivation is proceeding (but not yet complete), an "intermediate" indication is given.

Reporting Deactivation (X'82') Deactivation Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Deactivation subvector

Reporting Deactivation (X'82') Deactivation Subvector

Byte	Bit	Content
1		Key: X'82'
2−p		One required subfield (listed by Key value below and described in detail following): X'10' Deactivation Status

Deactivation Status (X'10') Reporting Deactivation Subfield

This subfield indicates whether or not a deactivation was successful. If the deactivation is proceeding (but not yet complete), an "intermediate" indication is given.

Deactivation Status (X'10') Reporting Deactivation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Deactivation Status subfield
1		Key: X'10'
2(=q)		Deactivation Status: X'10' Successful X'20' Unsuccessful X'50' Intermediate. The activation is not yet complete, but information concerning the deactivation process is available and is being reported.

Detailed Data (X'98') Reporting Deactivation Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Set Clock (X'8075') MS Major Vector

CP → CP

This major vector is used to request the target node set a clock or adjust the clock offset from GMT.

Set Clock (X'8075') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'8075'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Set Clock MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Set Clock (X'8075') Major Vector	
	CP-MSU	Notes
Reporting Level (X'09')	O	
Execution Window Timing (X'0B')	O	
Set Clock Command (X'81')	P	

Key:

P	Present one time
O	Optionally present one time

Set Clock MS Subvectors

Set Clock Command (X'81') Set Clock Subvector

The Set Clock Command subvector requests the target to set its internal clock or adjust the GMT offset maintained by the clock.

Set Clock Command (X'81') Set Clock Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the Set Clock Command subvector
1		Key: X'81'

Set Clock Command (X'81') Set Clock Subvector

Byte	Bit	Content
2–p		One or more subfields (listed by Key value below and described in detail following): X'10' Force Indication X'15' Clock Name X'20' Timing Source Name X'30' Date/Time to Set X'40' GMT Offset X'50' Increment/Decrement Interval *Note:* The following subfield keys may be used as indicated:

Subfield	Presence in Set Clock Command (X'81') Set Clock Subvector	
Force Indication (X'10')	O	
Clock Name (X'15')	O	
Timing Source Name (X'20')	CP	Note 1
Date/Time to Set (X'30')	CP	Note 2
GMT Offset (X'40')	CP	Note 2
Increment/Decrement Interval (X'50')	CP	Note 1

Key:

O Optionally present one time
CP Conditionally present one time

Notes:

1 Not present if the Date/Time to Set (X'30') subfield or the GMT Offset (X'40') subfield is present. Otherwise, it is optionally present.
2 Not present if the Timing Source Name (X'20') subfield is present. Otherwise, it is optionally present.

Force Indication (X'10') Set Clock Command Subfield

This subfield indicates whether to perform the set clock command based on the quiesced state of the target node. If not present, Force(NO) is implied and the command is executed only if the target is quiesced.

Force Indication (X'10') Set Clock Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Force Indication subfield

Force Indication (X'10') Set Clock Command Subfield

Byte	Bit	Content
1		Key: X'10'
2		Force indicator: X'10' No (do not force) - reject if not quiesced, otherwise execute the set clock command X'20' Yes (force) - set the clock even if not quiesced
3 – 5(=q)		Delay: optional binary number (bytes 3 and 4) and units (byte 5) of minutes, seconds or hours (X'00', X'01' or X'02' respectively) representing the maximum allowed wait time to quiesce (if not already quiesced) before either rejecting or forcing the set clock command. If a zero value is specified or Delay is absent (2=q), immediate action is requested.

Clock Name (X'15') Set Clock Command Subfield

The Clock Name subfield identifies which clock to set. Timing sources may also be set, in which case Clock Name is the name of the timing source. Coded graphic character set global ID to be used is 01134-00500. No leading or imbedded blanks are allowed. Trailing blanks are allowed.

Clock Name (X'15') Set Clock Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Clock Name subfield
1		Key: X'15'
2 – 17		Clock name. If a timing source time is being modified, the timing source name is specified.

Timing Source Name (X'20') Set Clock Command Subfield

The Timing Source Name subfield identifies which timing source to use to set a clock. The timing source named is used but never set. For example, the timing source value may be incremented or decremented to obtain a new value with which to set the clock named in the Clock Name subfield; however, the actual timing source is left unchanged. Coded graphic character set global ID to be used is 01134-00500. No leading or imbedded blanks are allowed. Trailing blanks are allowed.

Timing Source Name (X'20') Set Clock Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Timing Source Name subfield
1		Key: X'20'
2 – 17(= q)		Timing source name.

Date/Time to Set (X'30') Set Clock Command Subfield

The Date/Time to Set subfield indicates both the date and time the target is to set its clock to.

Date/Time to Set (X'30') Set Clock Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Date/Time to Set subfield
1		Key: X'30'
2 – 3		Year in hexadecimal
4		Month in hexadecimal
5		Day in hexadecimal
6		Hour in hexadecimal
7		Minute in hexadecimal
8		Local time or GMT indicator:
		X'10' Local Time
		X'20' Greenwich Mean Time (GMT)

GMT Offset (X'40') Set Clock Command Subfield

If the receiver's clock is set to local time, the GMT Offset subfield indicates the offset value the receiver is to set in its clock for purposes of calculating Greenwich Mean Time (GMT). If the receiver's clock is set to GMT, the GMT Offset subfield indicates the offset value the receiver is to set in its clock for purposes of calculating the local time.

GMT Offset (X'40') Set Clock Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the GMT Offset subfield
1		Key: X'40'
2–3		GMT Offset
	0	Adjustment indicator
		0 adjustment to be added to the local time to obtain GMT or subtracted from GMT to obtain local time (i.e., all time zones westward, between the Greenwich time zone and the International Date Line)
		1 adjustment to be subtracted from the local time to obtain GMT or added to GMT to obtain local time (i.e., all time zones eastward, between Greenwich time zone and the International Date Line)
	1–3	reserved
	4–7	number of hours of adjustment, in binary.
	8–15	number of minutes of adjustment, in binary.

Increment/Decrement Interval (X'50') Set Clock Command Subfield

The Increment/Decrement Interval subfield indicates how many hours, minutes, and seconds to increment or decrement the current time in the clock named in the Clock Name (X'15') Set Clock Command Subfield. This subfield may also be used to increment or decrement the timing source value named in the Timing Source Name (X'20') Set Clock Command Subfield.

Increment/Decrement Interval (X'50') Set Clock Command Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Increment/Decrement Interval subfield
1		Key: X'50'
2		Adjustment
		X'01' Increment
		X'02' Decrement
3		Hours (in hexadecimal) to adjust (X'00'–X'17')
4		Minutes (in hexadecimal) to adjust (X'00'–X'3B')
5		Seconds (in hexadecimal) to adjust (X'00'–X'3B')
6(=q)		Scope

Increment/Decrement Interval (X'50') Set Clock Command Subfield

Byte	Bit	Content
		X'10' Adjust only the clock
		X'20' Adjust both the clock and the GMT offset (by equal amounts)

Set Clock Report (X'0075') MS Major Vector

CP → CP

This major vector is used to indicate the acceptance, rejection, success, or failure of a Set Clock command.

Set Clock Report (X'0075') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'0075'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Set Clock Report MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Set Clock Report (X'0075') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	O	
Reporting Set Clock (X'82')	P	
Detailed Data (X'98')	O(n)	

Key:

P	Present one time
O	Optionally present one time
O(n)	Optionally present one or more times

Set Clock Report MS Subvectors

Reporting Clock Set (X'82') Set Clock Report Subvector

The Reporting Clock Set subvector indicates the status of a Set Clock request.

Reporting Clock Set (X'82') Set Clock Report Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Clock Set subvector

Reporting Clock Set (X'82') Set Clock Report Subvector

Byte	Bit	Content
1		Key: X'82'
2–p		One required subfield (listed by Key value below and described in detail following): X'10' Set Clock Status

Set Clock Status (X'10') Reporting Clock Set Subfield

This subfield indicates whether or not a set clock request was successful or not, accepted or not, or whether the command is proceeding but not yet complete (intermediate).

Set Clock Status (X'10') Reporting Clock Set Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Set Clock Status subfield
1		Key: X'10'
2(=q)		Set clock request status: X'10' Successful X'20' Unsuccessful X'30' Accepted X'40' Rejected X'50' Intermediate. The execution is not yet complete, but information concerning the set clock process is available and is being reported.

Detailed Data (X'98') Reporting Clock Set Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Request Cancelation (X'8076') MS Major Vector

LU → LU, CP → CP

This major vector is used to request the target node cancel one or more outstanding requests.

Request Cancelation (X'8076') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'8076'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Request Cancelation MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Request Cancelation (X'8076') Major Vector	
	CP-MSU	Notes
Reporting Level (X'09')	O	
Cancel (X'81')	P	

Key:

P Present one time
O Optionally present one time

Request Cancelation MS Subvectors

Cancel (X'81') Request Cancelation MS Subvector

The Cancel subvector requests MS at the target node to stop or prevent the execution of one or more requests previously accepted at the node.

Cancel (X'81') Request Cancelation MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Cancel subvector
1		Key: X'81'

Cancel (X'81') Request Cancelation MS Subvector

Byte	Bit	Content
2−p		One of three mutually exclusive subfields (listed by Key value below and described in detail following): X'10' Cancel Specific Tasks X'20' Cancel All X'30' Cancel All Deferred

Cancel Specific Tasks (X'10') Cancel Subfield

This subfield indicates a specific request to be canceled. It is mutually exclusive with the Cancel All (X'20') and Cancel All Deferred (X'30') subfields, but is always present when either the Cancel All or the Cancel All Deferred subfield is not present. This subfield may appear twice within the Cancel (X'81') subvector in order to specify both the (X'30') and (X'40') code variants.

Cancel Specific Tasks (X'10') Cancel Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Cancel Specific Tasks subfield
1		Key: X'10'
2		One of the following values identifying the request to be canceled: X'10' Cancel the requests identified by the Agent Unit of Work Correlator (X'1549') GDS variables contained in the same CP-MSU as the Request Cancelation MS major vector. Each Agent Unit of Work Correlator within the CP-MSU corresponds to a request that is to be canceled. X'20' Cancel the request currently executing under control of the agent identified in the Routing and Targeting Instructions GDS (registered codepoint X'154D') contained in the same CP-MSU as the Request Cancelation MS major vector. In this case, the Agent Unit of Work Correlator GDS is not present in the CP-MSU.

Cancel Specific Tasks (X'10') Cancel Subfield

Byte	Bit	Content
		X'30' Cancel the currently running request received by a named application program. The name of the application program begins at offset 3 of this subfield. The application program name may be either EP_OPERATIONS_MGMT, EP_CHANGE_MGMT, or some other non-SNA/MS registered application program name. The request against which a cancel is attempted is the request which was received by the named application program and which is still executing under control of the named application program. The named application program may or may not be the same as the application program to which the cancel address is addressed. The named application program must be located at the same entry point as the application program to which the cancel request is addressed. It is not stated whether the application program receiving the cancel request actually performs the cancel action, or otherwise causes the named application program to perform the cancel action.
		X'40' Cancel the currently running request sent by a named application program. The name of the application program begins at offset 3 of this subfield. The application program name may be OPERATIONS_MGMT_NETOP, CHANGE_MGMT_NETOP, or some other non-SNA/MS registered application program name. The request against which a cancel is attempted is the request which was sent by the named application program and which is currently executing under control of the receiver of the cancel request. The application program name is used as a search argument against the requests held by the receiver.
3 – 10(=q)		Conditionally present application program name (may be SNA/MS registered or not). A registered name is recognized when the first character of the name is less than X'40'. This application program name is present if the indicator at byte offset 2 is set to X'30' or X'40'. Otherwise it is not present.

Cancel All (X'20') Cancel Subfield

This subfield is the vehicle to be used for canceling all requests meeting certain criteria. It is mutually exclusive with the Cancel Specific Tasks (X'10') and Cancel All Deferred (X'30') subfields, but is present whenever the Cancel Specific Tasks or the Cancel All Deferred subfield is not present. The Agent Unit of Work Correlator (X'1549') GDS variable is never present in the CP-MSU when this subfield is used. This subfield may appear twice within the Cancel (X'81') subvector in order to specify both the (X'30') and (X'40') code variants.

Cancel All (X'20') Cancel Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Cancel All subfield
1		Key: X'20'
2		Cancel All Indicators: X'10' Cancel all requests received by an application program, namely the one which has received the cancel request. The set of requests against which a cancel is attempted consists of those requests which the receiver of the cancel request had previously received and which are still held by the receiver of the cancel request. X'20' Cancel all requests sent by the sending second level application program of the cancel request, regardless of the node of the sender. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request to be canceled must match the OII of the application program sending the cancel request. If the OII is not present in the Routing and Targeting Instructions (X'154D') GDS variable of the cancel request, then an OII match is not necessary to otherwise have a successful match. • Information from the transport mechanism concerning the origin is not used in this attempt to match. Note that this statement is independent of the nature of the transport mechanism. The set of requests against which a cancel is attempted consists of those requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request. X'21' Cancel all requests sent by the sending second level application program of the cancel request, regardless of the node of the sender. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request, if present, is disregarded. • Information from the transport mechanism concerning the origin is not used in this attempt to match. Note that this statement is independent of the nature of the transport mechanism. The set of requests against which a cancel is attempted consists of those requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request.

Cancel All (X'20') Cancel Subfield

Byte	Bit	Content	
		X'28'	Cancel all requests sent by the sender of the cancel request. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request to be canceled must match the OII of the application program sending the cancel request. If the OII is not present in the Routing and Targeting Instructions (X'154D') GDS variable of the cancel request, then an OII match is not necessary to otherwise have a successful match. • The node identification and other applicable transport mechanism identification data of a candidate request to be canceled must match the same transport mechanism identification data of the sender of the cancel request. When the MDS transport is the transport mechanism, then the net-id, NAU-name and the MS Application name of a candidate request must match the net-id, NAU-name and MS Application name of the sender of the cancel request. The set of requests against which a cancel is attempted consists of those requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request.
		X'29'	Cancel all requests sent by the sender of the cancel request. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request, if present, is disregarded. • The node identification and other applicable transport mechanism identification data of a candidate request to be canceled must match the same transport mechanism identification data of the sender of the cancel request. When the MDS transport is the transport mechanism, then the net-id, NAU-name and the MS Application name of a candidate request must match the net-id, NAU-name and MS Application name of the sender of the cancel request. The set of requests against which a cancel is attempted consists of those requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request.

Cancel All (X'20') Cancel Subfield

Byte	Bit	Content	
		X'30'	Cancel all requests received by a named application program. The name of the application program begins at byte offset 3 of this subfield. The application program name may be either EP_OPERATIONS_MGMT, EP_CHANGE_MGMT, or some other non-SNA/MS registered application program name. The set of requests against which a cancel is attempted consists of those requests which were received by the named application program and which are still held by the named application program. The named application program may or may not be the same application program to which the cancel request is addressed. The named application program must be located at the same entry point as the application program to which the cancel request is addressed. It is not stated whether the application program receiving the cancel request actually performs the cancel action, or otherwise causes the named application program to perform the cancel action.
		X'40'	Cancel all requests sent by the named application program. The name of the application program begins at byte offset 3 of this subfield. The application program name may be OPERATIONS_MGMT_NETOP, CHANGE_MGMT_NETOP, or some other non-SNA/MS registered application program name. The set of requests against which a cancel is attempted consists of those requests which were sent by the named application program and which are still held by the receiver of the cancel request. The application program name is used as a search argument against the requests held by the receiver.
		X'50'	Cancel all requests received by both EP_OPERATIONS_MGMT and EP_CHANGE_MGMT. The set of requests against which a cancel is attempted consists of those requests that were received and are still held by EP_OPERATIONS_MGMT and EP_CHANGE_MGMT. The cancel request may or may not be addressed to either EP_OPERATIONS_MGMT or EP_CHANGE_MGMT. It is not stated whether the application program receiving the cancel request actually performs the cancel action, or otherwise causes EP_OPERATIONS_MGMT or EP_CHANGE_MGMT to perform the cancel action.
		X'60'	Cancel all requests sent by both OPERATIONS_MGMT_NETOP and CHANGE_MGMT_NETOP The set of requests against which the cancel is attempted consists of those requests which were sent by both OPERATIONS_MGMT_NETOP and CHANGE_MGMT_NETOP and which are still help by the receiver of the cancel request.

Cancel All (X'20') Cancel Subfield

Byte	Bit	Content
		X'70' Cancel all requests received by any of the following: • EP_OPERATIONS_MGMT • All second-level application programs served by EP_OPERATIONS_MGMT • EP_CHANGE_MGMT The set of requests against which a cancel is attempted consists of those requests which were received by all of the application programs meeting any of the criteria listed above and which are still held by each application program. The application programs include the application program to which the cancel request is addressed. The application programs must be located at the same entry point as the application program to which the cancel request is addressed. It is not stated whether the application program receiving the cancel request actually performs the cancel action, or otherwise causes each application program to perform the cancel action.
3 – 10(=q)		Conditionally present application program name (may be SNA/MS registered or not). A registered name is recognized when the first character of the name is less than X'40'. This application program name is present if the indicator at byte offset 2 is set to X'30' or X'40'. Otherwise it is not present.

Cancel All Deferred (X'30') Cancel Subfield

This subfield is the vehicle to be used for canceling all deferred requests meeting certain criteria. Deferred requests are defined as those requests which have been accepted for execution but deferred according to the Execution Window Timing (X'0B') subvector parameters in the request. It is mutually exclusive with the Cancel Specific Tasks (X'10') and Cancel All (X'20') subfields, but is present whenever the Cancel Specific Tasks or the Cancel All subfield is not present. The Agent Unit of Work Correlator (X'1549') GDS variable is never present in the CP-MSU when this subfield is used. This subfield may appear twice within the Cancel (X'81') subvector in order to specify both the (X'30') and (X'40') code variants.

Cancel All Deferred (X'30') Cancel Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Cancel All Deferred subfield
1		Key: X'30'
2		Cancel All Deferred Indicators:

Cancel All Deferred (X'30') Cancel Subfield

Byte	Bit	Content	
		X'10'	Cancel all deferred requests received by an application program, namely the one which has received the cancel request. The set of requests against which a cancel is attempted consists of those deferred requests which the receiver of the cancel request had previously received and which are still held by the receiver of the cancel request.
		X'20'	Cancel all deferred requests sent by the sending second level application program of the cancel request, regardless of the node of the sender. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request to be canceled must match the OII of the application program sending the cancel request. If the OII is not present in the Routing and Targeting Instructions (X'154D') GDS variable of the cancel request, then an OII match is not necessary to otherwise have a successful match. • Information from the transport mechanism concerning the origin is not used in this attempt to match. Note that this statement is independent of the nature of the transport mechanism. The set of requests against which a cancel is attempted consists of those deferred requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request.
		X'21'	Cancel all deferred requests sent by the sending second level application program of the cancel request, regardless of the node of the sender. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request, if present, is disregarded. • Information from the transport mechanism concerning the origin is not used in this attempt to match. Note that this statement is independent of the nature of the transport mechanism. The set of requests against which a cancel is attempted consists of those deferred requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request.

Cancel All Deferred (X'30') Cancel Subfield

Byte	Bit	Content	
		X'28'	Cancel all deferred requests sent by the sender of the cancel request. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request to be canceled must match the OII of the application program sending the cancel request. If the OII is not present in the Routing and Targeting Instructions (X'154D') GDS variable of the cancel request, then an OII match is not necessary to otherwise have a successful match. • The node identification and other applicable transport mechanism identification data of a candidate request to be canceled must match the same transport mechanism identification data of the sender of the cancel request. When the MDS transport is the transport mechanism, then the net-id, NAU-name and the MS Application name of a candidate request must match the net-id, NAU-name and MS Application name of the sender of the cancel request. The set of requests against which a cancel is attempted consists of those deferred requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request.
		X'29'	Cancel all deferred requests sent by the sender of the cancel request. The criteria to search the set of requests currently at the receiver are: • The origin application program name (OAN) of a candidate request to be canceled must match the OAN of the application program sending the cancel request. • The origin instance identifier (OII) of a candidate request, if present, is disregarded. • The node identification and other applicable transport mechanism identification data of a candidate request to be canceled must match the same transport mechanism identification data of the sender of the cancel request. When the MDS transport is the transport mechanism, then the net-id, NAU-name and the MS Application name of a candidate request must match the net-id, NAU-name and MS Application name of the sender of the cancel request. The set of requests against which a cancel is attempted consists of those deferred requests which satisfy the stated criteria regarding origin and which are still held by the receiver of the cancel request.

Cancel All Deferred (X'30') Cancel Subfield

Byte	Bit	Content	
		X'30'	Cancel all deferred requests received by a named application program. The name of the application program begins at byte offset 3 of this subfield. The application program name may be either EP_OPERATIONS_MGMT, EP_CHANGE_MGMT, or some other non-SNA/MS registered application program name. The set of requests against which a cancel is attempted consists of those deferred requests which were received by the named application program and which are still held by the named application program. The named application program may or may not be the same application program to which the cancel request is addressed. The named application program must be located at the same entry point as the application program to which the cancel request is addressed. It is not stated whether the application program receiving the cancel request actually performs the cancel action, or otherwise causes the named application program to perform the cancel action.
		X'40'	Cancel all deferred requests sent by the named application program. The name of the application program begins at byte offset 3 of this subfield. The application program name may be OPERATIONS_MGMT_NETOP, CHANGE_MGMT_NETOP, or some other non-SNA/MS registered application program name. The set of requests against which a cancel is attempted consists of those deferred requests which were sent by the named application program and which are still held by the receiver of the cancel request. The application program name is used as a search argument against the requests held by the receiver.
		X'50'	Cancel all deferred requests received by both EP_OPERATIONS_MGMT and EP_CHANGE_MGMT. The set of requests against which a cancel is attempted consists of those deferred requests that were received and are still held by EP_OPERATIONS_MGMT and EP_CHANGE_MGMT. The cancel request may or may not be addressed to either EP_OPERATIONS_MGMT or EP_CHANGE_MGMT. It is not stated whether the application program receiving the cancel request actually performs the cancel action, or otherwise causes EP_OPERATIONS_MGMT or EP_CHANGE_MGMT to perform the cancel action.
		X'60'	Cancel all deferred requests sent by both OPERATIONS_MGMT_NETOP and CHANGE_MGMT_NETOP The set of requests against which the cancel is attempted consists of those deferred requests which were sent by both OPERATIONS_MGMT_NETOP and CHANGE_MGMT_NETOP and which are still help by the receiver of the cancel request.

Cancel All Deferred (X'30') Cancel Subfield

Byte	Bit	Content
		X'70' Cancel all deferred requests received by any of the following: • EP_OPERATIONS_MGMT • All second-level application programs served by EP_OPERATIONS_MGMT • EP_CHANGE_MGMT The set of requests against which a cancel is attempted consists of those deferred requests which were received by all of the application programs meeting any of the criteria listed above and which are still held by each application program. The application programs include the application program to which the cancel request is addressed. The application programs must be located at the same entry point as the application program to which the cancel request is addressed. It is not stated whether the application program receiving the cancel request actually performs the cancel action, or otherwise causes each application program to perform the cancel action.
3 − 10(=q)		Conditionally present application program name (may be SNA/MS registered or not). A registered name is recognized when the first character of the name is less than X'40'. This application program name is present if the indicator at byte offset 2 is set to X'30' or X'40'. Otherwise it is not present.

Cancelation (X'0076') MS Major Vector

CP → CP

This major vector is used by an entry point to return the results of a cancelation to a focal point. Cancelation may have been requested by the focal point or locally.

Cancelation (X'0076') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0076'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Cancelation MS Subvectors" for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Cancelation (X'0076') Major Vector	
	CP-MSU	Notes
Date/Time (X'01')	O	
Reporting Cancelation (X'82')	P	
Detailed Data (X'98')	O(n)	

Key:

P	Present one time
O	Optionally present one time
O(n)	Optionally present one or more times

Cancelation MS Subvectors

Reporting Cancelation (X'82') Cancelation MS Subvector

The Reporting Cancelation subvector indicates whether or not a cancel request has been accepted for execution. It is also used to indicate (if the cancelation is accepted) success or failure of the execution. This subvector may also be used to report intermediate status concerning the cancelation process.

Reporting Cancelation (X'82') Cancelation MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Cancelation subvector
1		Key: X'82'
2–4(=p)		The Cancelation Status (X'10') subfield.

Cancelation Status (X'10') Reporting Cancelation Subfield

This subfield reports whether or not cancelation was successful. It is always present once.

Cancelation Status (X'10') Reporting Cancelation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Cancelation Status subfield
1		Key: X'10'
2(=q)		Cancelation status: X'10' Successful X'20' Attempted, but not successful X'30' Not attempted and will not attempt X'40' Will attempt X'50' Intermediate. The cancelation is not yet complete, but information concerning the cancelation process is available and is being reported.

Detailed Data (X'98') Cancelation MS Subvector

This subvector transports product-specific detailed data.

Note: The format of this subvector is defined under the Alert (X'0000') major vector. However, no Qualified Message Data (X'01') subfields may be present.

Routing/Parsing Report (X'0077') MS Major Vector

CP → CP

This major vector is used to report an error associated with the routing of an MDS-MU to or from an application program served by a focal point or entry point application program in the same node. It may also be used to indicate parsing errors associated with those same MDS-MUs. It is not used for reporting or tranporting MDS error messages. Other formats within the same CP-MSU as the Routing/Parsing Report MS Major Vector contain details of the error.

Routing/Parsing Report (X'0077') MS Major Vector

Byte	Bit	Content
0 – 1		Length (n + 1), in binary, of this MS major vector
2 – 3(=n)		Key: X'0077'

Request Response Time Monitor (X'8080') MS Major Vector

SSCP → PU T2

This major vector enables or disables response time monitoring, transports RTM parameters, and transports a request for RTM data and status from a device.

Request Response Time Monitor (X'8080') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'8080'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Request Response Time Monitor Subvectors" on page 9-279 for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Request RTM (X'8080') Major Vector	
	NMVT	Notes
SNA Address List (X'04')	CP	Note 1
*RTM Request (X'92')	P	
RTM Control (X'94')	CP	Note 2

Key:

*	Command Subvector (for PU parsing)
P	Present one time
CP	Conditionally present one time (See Notes for conditions.)

Notes:

1. This subvector is present in the NMVT containing an X'8080' major vector when the request is for a specific LU (i.e., identified in the X'04' subvector) associated with the PU processing the request. This subvector is not present when the request is to apply to all LUs associated with the PU processing the request.
2. This subvector is present when RTM parameters are being set. If present, it immediately follows the RTM Request (X'92' subvector).

Request Response Time Monitor Subvectors

RTM Request (X'92') Request RTM MS Subvector

This subvector requests RTM data and status or accompanies an RTM control subvector.

RTM Request (X'92') Request RTM MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector
1		Key: X'92'
2		Request indicators (bit is set to 1 to request that the function be performed):
	0	Reset RTM data for the target LU upon reply transmission or immediately if no reply is requested.
	1	Retrieve data and status for all LUs with accumulated RTM data. See Figure 9-1.
	2	Retired
	3	Retrieve data and status for the LU specified in the SNA Address List (X'04') MS common subvector also included in this major vector. See Figure 9-1.
	4	Apply the RTM Control (X'94') MS subvector also included in this major vector to all LUs. *Note:* If this bit is set to 1, the RTM Control (X'94') MS subvector will be present. If this bit is set to 0 and the RTM Control (X'94') subvector is present, the SNA Address List (X'04') MS common subvector will be present.
	5–6	Retired
3(=p)		Reserved

Figure 9-1. Setting of Bits 1 and 3 of Byte 2 of the RTM Request (X'92') Subvector

Request Type	Subvectors present in the Request RTM (X'8080') major vector	Bits	
		B1	B2
Retrieve data for all LUs with accumulated data	92	1	0
Retrieve for specified LU	92, 04	0	1
Set parameters for all LUs	92, 94	0	0
Set parameters for specified LU	92, 94, 04	0	0

RTM Control (X'94') Request RTM MS Subvector

This subvector controls RTM data accumulation.

RTM Control (X'94') Request RTM MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector
1		Key: X'94'
2–3		RTM status and control change mask (bit is set to 1 if the setting specified by the corresponding RTM status and control indicator in bytes 4–5 should be used):
	0–8	Mask bits corresponding respectively to indicator bits 0–8 in bytes 4–5
	9–15	Reserved
4–5		RTM status and control indicators (bit is set to 1 to activate the function or 0 to deactivate it):
	0	RTM measurement active
	1	Return data unsolicited on session deactivation
	2	Return data unsolicited on counter overflow
	3	Retired
	4	Set the RTM measurement definition using byte 8
	5	Set the RTM response time measurement boundaries using bytes 9 and 16–m
	6	Retired
	7	Local display of RTM data
	8	Retired
	9–15	Reserved
6		Reserved
7		Retired
8		RTM measurement definition—defines when the response-time measurement will begin and end for each exchange between session partners: X'01' measured from the Attention or Action key depression to the arrival back at the LU of the first character that can alter the presentation space X'02' measured from the Attention or Action key depression until the LU is ready to accept input from its end user X'03' measured from the Attention or Action key depression to the receipt and processing back at the LU of Change Direction (CD) or End Bracket (EB) or CEB X'04' measured from the Attention or Action key depression to the receipt of the last character of the last message received prior to the next Attention or Action key depression

RTM Control (X'94') Request RTM MS Subvector

Byte	Bit	Content
9		Response-time unit of measure: X'00' 100 milliseconds X'01'–X'7F' retired X'90' retired X'A0' retired X'C0' retired X'D0' retired
10–15		Reserved
16		RTM data collection parameters:
	0–3	Reserved
	4–7	Binary number of 2-byte boundaries in bytes 17-p
17–p		A set of response-time measurement boundaries, specified in binary (as units of response-time units of measure described by byte 9) and increasing in order of magnitude; thus, response-time data is collected for intervals ($0 < r1 \leq b1 \times u$), ($b1 \times u < r2 \leq b2 \times u$), ... up to ($b4 < r5$), where bi is the value of the boundary i, ri is the response-time being measured for interval i, and u is the unit of measure described by byte 9.
17–18		Boundary 1
19–20		Boundary 2
21–p		Additional boundaries as required (the total number is defined by byte 16, bits 4–7), up to a maximum of 4

Response Time Monitor (X'0080') MS Major Vector

PU T2 → SSCP

This major vector transports RTM data. This data includes the collected response time data and current RTM status.

Response Time Monitor (X'0080') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'0080'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Response Time Monitor Subvectors" on page 9-283 for subvector keys X'80'–X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in RTM (X'0080') Major Vector	
	NMVT	Notes
Date/Time (X'01')	CP	Note 1
SNA Address List (X'04')	CP	Note 2
Relative Time (X'42')	CP	Note 3
Data Reset Flag (X'45')	CP	Note 4
Sense Data (X'7D')	CP	Note 5
RTM Status Reply (X'91')	CP	Note 6
RTM Data (X'93')	CP	Note 7

Key:

CP Conditionally present one time (See Notes for conditions.)

Notes:

1. If the PU sending the X'0080' major vector has the capability of providing it, it places this subvector in the NMVT.
2. This subvector is present when positively replying to a request for RTM data and status, or when RTM data and status are sent unsolicited.
3. If the PU sending the X'0080' cannot provide a Date/Time subvector, it places this subvector in the NMVT instead.
4. This subvector is present in an X'0080' major vector when a set of counters has been reset, either as a result of a request or when sent unsolicited.

5. This subvector is present when a Request RTM major vector cannot be processed, or when requested data cannot be gathered and the PU sending this major vector has elected to send sense data in a reply instead of a negative response.
6. This subvector is present when positively replying to a request for RTM data and status, or when RTM data and status are sent unsolicited.
7. This subvector is present when positively replying to a request for RTM data and status if RTM data has been accumulated, or when RTM data and status are sent unsolicited.

Response Time Monitor Subvectors

RTM Status Reply (X'91') RTM MS Subvector

This subvector transports the current status of RTM function for a device.

RTM Status Reply (X'91') RTM MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector
1		Key: X'91'
2		Reply indicators (bit is set to 1 to indicate that the assertion is true):
	0	Reserved
	1	Data not included
	2	An RTM data request has been issued for an LU that has its RTM function disabled
	3	This is the first RTM reply since session activation (used to initiate a recording of the session partner names and the correlation value in bytes 7 – 8 of this subvector while there is reasonable assurance that the session is active); on subsequent replies, the correlation value will be used to associate data from the same LU-LU session
	4	An LU-LU session activation or deactivation has occurred at least once while the included RTM data was being accumulated
	5 – 7	Reserved
3		Reason for unsolicited reply, if any (bit is set to 1 to indicate the appropriate reason):
	0	The session for this resource has ended and is enabled unsolicited-reply-on-session-deactivation
	1	A counter for this LU has overflowed and unsolicited-reply-on-counter-overflow is enabled
	2	Retired
	3 – 5	Reserved
	6	Retired
	7	Reserved

RTM Status Reply (X'91') RTM MS Subvector

Byte	Bit	Content
4		Reason for potential loss of RTM data, if any (bit is set to 1 to indicate the reason):
	0	Reserved
	1	An overflow has occurred on at least one counter and updating for all of this LU's counters has been stopped to retain the relative significance of the data
	2	This LU has been reset since the last reply was sent
	3	A new session was activated before data for the previous session could be transmitted: loss of data for the new session may have occurred; updating for all of this LU's counters has been stopped to retain the relative significance of the data
	4	The RTM definition or response time measurement boundaries have been changed by a request that did not solicit the RTM data and RTM accumulation was active for this LU: any data collected since the last data request has been lost
	5–7	Reserved
5–6		RTM status when this subvector was constructed (a bit set to 1 indicates that the function was active):
	0	RTM measurement active
	1	Data to be sent unsolicited on session deactivation
	2	Data to be sent unsolicited on counter overflow
	3	Retired
	4	RTM definition was set by the control point
	5	RTM boundaries were set by the control point
	6	Retired
	7	Local display of RTM data
	8	Retired
	9–15	Reserved
7–8(=p)		Correlation value: a unique 2-byte value, generated by the PU, that is retained and used in all RTM replies dealing with the same LU-LU session from session activation through the subsequent session deactivation

RTM Data (X'93') RTM MS Subvector

This subvector transports solicited or unsolicited RTM data.

RTM Data (X'93') RTM MS Subvector

Byte	Bit	Content
0		Length (q+9), in binary, of this subvector
1		Key: X'93'

RTM Data (X'93') RTM MS Subvector

Byte	Bit	Content
2		RTM measurement definition in effect: X'01' measured from the Attention or Action key depression to the arrival back at the LU of the first character that can alter the presentation space X'02' measured from the Attention or Action key depression until the LU is ready to accept input from its end user X'03' measured from the Attention or Action key depression to the receipt and processing back at the LU of Change Direction (CD) or End Bracket (EB) X'04' measured from the Attention or Action key depression to the receipt of the last character of the last message received prior to the next Attention or Action key depression.
3		Response time unit of measure: X'00' 100 milliseconds X'01'–X'7F' retired X'90' retired X'A0' retired X'C0' retired X'D0' retired
4–5		Reserved
6–7		Retired
8		RTM data collection parameters:
	0–3	The number, in binary, of response time measurement boundaries returned; all boundaries that were set previously will be returned in this subvector
	4–7	The number, in binary, of boundary sets for which valid data was collected (overflow data—a count of response times exceeding the maximum boundary—is not included in this number but is always present)
9–p		A set of response-time measurement boundaries as previously set at the LU or by the RTM Control (X'94') MS subvector (specified in binary as units of response-time units of measure described by byte 9) and increasing in order of magnitude; thus, response-time data is collected for intervals ($0 < r1 \leq 1 \times u$), ($b1 \times u < r2 \leq b2 \times u$), ... up to ($b4 \times u < r5$), where bi is the value of the boundary i, ri is the response-time being measured for interval i, and u is the unit of measure described by byte 9
9–10		Boundary 1
11–12		Boundary 2
13–p		Additional boundaries as required to equal the number of boundaries set previously and specified by byte 8, bits 0–3
p+1–q		The number of measured exchanges for each response-time interval: the number of exchanges whose duration was within an interval's boundaries is reported in binary separately for each interval
p+1–p+2		Number of exchanges in the (0,b1) range
p+3–p+4		Number of exchanges in the (b1+1,b2) range

RTM Data (X'93') RTM MS Subvector

Byte	Bit	Content
p+5−q		Additional exchange counts to satisfy the number of boundaries defined , up to a maximum of 4
q+1−q+2		Overflow: the number of exchange durations greater than the maximum boundary specified
q+3−q+6		Total of all individual times for all exchanges measured and reported by this record, including overflow, in the measurement units defined by byte 3
q+7−q+8		Last measured exchange duration in the measurement units defined by byte 3

Request Product Set ID (X'8090') MS Major Vector

SSCP → PU

This major vector transports a request for product identification from a network component.

Request Product Set ID (X'8090') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'8090'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and in "Request Product Set ID Subvectors" on page 9-288 for subvector keys X'80'–X'FE'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Request PSID (X'8090') Major Vector	
	NMVT	Notes
*Node Identification (X'81')	CP	Note 1
*Node and Port-Attached Devices Identification (X'83')	CP	Note 2
Unsolicited Criteria (X'85')	CP	Note 3

Key:

*	Command subvector (for PU parsing)
CP	Conditionally present one time. (See notes for conditions.)

Notes:

1. This subvector is present when requesting the PSID for just the node (PU) the major vector is sent to. The PSID is returned in a solicited Reply PSID major vector. Additionally, unsolicited Reply PSID major vectors may be returned (see Note 3).
2. This subvector is present when requesting the PSID for the node (PU) the major vector is sent to *and* its port-attached devices. The PSIDs are returned in solicited Reply PSID major vectors. Additionally, unsolicited Reply PSID major vectors may be returned (see Note 3).
3. This subvector is present when requesting the target to send unsolicited Reply PSID major vectors when specific criteria are met. This subvector specifies those criteria. This subvector may be present with either the X'81' or X'83' command subvector. If this X'85' subvector is absent in the Request major vector, unsolicited Reply PSID major vectors shall not be sent by the Request target to the requesting CP.

Independent of the presence of the X'85' subvector, solicited PSID is returned since either the X'81' or X'83' subvector shall be included in the Request PSID major vector.

Request Product Set ID Subvectors

Node Identification (X'81') Request PSID MS Subvector

This subvector requests product identification from the node receiving the request. The presence of the X'85' subvector may cause additional Reply PSID major vectors to be sent after the node is reported on with a single Reply PSID major vector.

Node Identification (X'81') Request PSID MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector
1(=p)		Key: X'81'

Node and Port-Attached Devices Identification (X'83') Request PSID MS Subvector

This subvector requests product identification from the node the major vector is sent to. The node must build one Reply Product Set ID (X'0090') major vector for itself and one for each port-attached device for which the node has product identification. The presence of the X'85' subvector may cause additional Reply PSID major vectors to be sent after the node and all port-attached devices are reported on.

Node and Port-Attached Devices Identification (X'83') Request PSID MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector
1(=p)		Key: X'83'

Unsolicited Criteria (X'85') Request PSID MS Subvector

This subvector specifies the criteria which the Request PSID target uses to determine if a Reply PSID major vector is to be sent when a product's identification changes. If this subvector is present with the X'81' command subvector, unsolicited Reply PSID major vectors are sent when the node (implementing the PU) changes and the criteria specified in X'85' are satisfied. If this subvector is present with the X'83' command subvector, unsolicited Reply PSID major vectors are sent when the node (implementing the PU) changes or when port-attached products change and criteria specified in X'85' are satisfied.

Unsolicited Criteria (X'85') Request PSID MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector
1		Key: X'85'
2 (=p)		Criteria for sending unsolicited PSID: X'01' send Reply PSID major vector each time a hardware product powers on or off.

Reply Product Set ID (X'0090') MS Major Vector

PU → SSCP

This major vector transports product identification information that was requested by the Request Product Set ID (X'8090') major vector. The Reply PSID major vector can flow unsolicited when:

- The criteria specified in the Unsolicited Criteria (X'85') subvector (of the Request PSID major vector) are met or
- Unsolicited Reply PSID major vector support is indicated in the PU Capabilities (X'80') control vector of the ACTPU RU.

Reply Product Set ID (X'0090') MS Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'0090'
4–n		MS subvectors, as described (using zero-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' – X'7F', and in "Reply PSID MS Subvectors" on page 9-291 for subvector keys X'80' – X'FE'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Reply PSID (X'0090') Major Vector	
	NMVT	Notes
Date/Time (X'01')	CP	Note 1
SNA Address List (X'04')	CP	Note 2
Product Set ID (X'10')	P	
Relative Time (X'42')	CP	Note 3
Port-Attached Device Configuration Description (X'82')	CP	Note 4
Product Set Attributes (X'84')	O	
Additional Product Set Attributes (X'86')	O(n)	

Key:

P	Present one time
CP	Conditionally present one time (See Notes for conditions.)
O	Optionally present one time
O(n)	Optionally present one or more times

Notes:

1. If the PU sending the X'0090' major vector has the capability of providing it, this subvector is placed in the NMVT.

2. This subvector is present when the major vector is reporting on a port-attached device. The address present in this subvector identifies the LU most closely associated with the device.
3. If the PU sending the X'0090' cannot provide a Date/Time subvector, it places this subvector in the NMVT instead.
4. This subvector is present when the major vector is reporting on a port-attached device.

Reply PSID MS Subvectors

Port-Attached Device Configuration Description (X'82') Reply PSID MS Subvector

This MS subvector describes the configuration of a device port-attached to the node to which the Request PSID major vector was sent.

Port-Attached Device Configuration Description (X'82') Reply PSID MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector.
1		Key: X'82'
2–p		The following subfields are required: X'10' Port number X'20' Power-on status X'30' Power-on since last solicitation

Port Number (X'10') Port-Attached Device Config. Des. Subfield

This subfield contains the port number the device is attached to. The port number is associated with the "parent" node (the node to which the Request Product Set ID major vector was sent). Port is a generic term. It may be a physical port which is the attached device's interface to the product containing the PU sending the Reply PSID major vector. Alternatively, bytes 2-q of this subfield may carry a logical interface number. For example, when multiple physical devices share a single physical port, bytes 2-q may contain a logical address (e.g., physical port 2/station 5).

Port Number (X'10') Port-Attached Device Config. Des. Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of this subfield
1		Key: X'10'

Port Number (X'10') Port-Attached Device Config. Des. Subfield

Byte	Bit	Content
2−q		Number of the port the device is attached to: numeric characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."

Power-on Status (X'20') Port-Attached Device Config. Des. Subfield

This subfield transports the power-on status of the attached device.

Power-on Status (X'20') Port-Attached Device Config. Des. Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of this subfield
1		Key: X'20'
2(=q)		Power-on Status of the device: X'01' device is currently powered on X'02' device is currently powered off

Power-on Since Last Solicitation (X'30') Port-Attached Device Config. Des. Subfield

This subfield states whether the device was powered on since the last solicition (i.e., since the last time a Reply PSID major vector was sent by this node).

Power-on Since Last Solicitation (X'30') Port-Attached Device Config. Des. Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of this subfield
1		Key: X'30'
2(=q)		Power-on since last solicitation X'01' device was powered on (from a power-off position) at least once X'02' device was not powered on (although it may have been powered on before the previous solicitation and remained powered on) since the last solicitation

Product Set Attributes (X'84') Reply PSID MS Subvector

This MS subvector transports attributes describing the product set.

Product Set Attributes (X'84') Reply PSID MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of this subvector.
1		Key: X'84'
2−p		One or both of the the following subfields: X'00' Physical location X'10' LAN Universally-Administered Address

Physical Location (X'00') Product Set Attributes Subfield

This subfield contains the physical location of the product set (e.g, city/room/building).

Physical location (X'00') Product Set Attributes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of this subfield
1		Key: X'00'
2−q		Physical location of the product set: up to 50 characters from Coded Graphic Character Set 00640 − 00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."

LAN Universally-Administered Address (X'10') Product Set Attributes Subfield

This subfield contains the LAN universally-administered address.

LAN Universally-Administered Address (X'10') Product Set Attributes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of this subfield
1		Key: X'10'

LAN Universally-Administered Address (X'10') Product Set Attributes Subfield

Byte	Bit	Content
2–q		6 bytes (12 hexadecimal digits) identifying the LAN universally-administered MAC Address. No blanks are allowed. The 6 bytes are unique across all LAN adapters whose addresses are controlled by the IEEE.

Additional Product Set Attributes (X'86') Reply PSID MS Subvector

This MS subvector transports additional data describing the product set. The data is labeled and is user-defined. This subvector is to be used to pass product set data not defined by the architecture (i.e., not in the X'84' subvector). Sample values of X'00' and X'10' subfields are "LESSOR=" and "XYZ, INC.," respectively.

Additional Product Set Attributes (X'86') Reply PSID MS Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of this subvector
1		Key: X'86'
2–p		One Label (X'00') subfield followed by one Data (X'10') subfield

Label (X'00') Additional Product Set Attributes Subfield

This subfield contains the label which describes the data in the Data (X'10') subfield.

Label (X'00') Additional Product Set Attributes Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of this subfield
1		Key: X'00'
2–q		Label describing the data in the X'10' subfield: Up to 25 characters from Coded Graphic Character Set 00640–00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."

Data (X'10') Additional Product Set Attributes Subfield

This subfield contains the data which is described by the Label (X'00' subfield) immediately preceding this subfield.

Data (X'10') Additional Product Set Attributes Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of this subfield
1		Key: X'10'
2−q		Up to 224 characters from Coded Graphic Character Set 00640-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."

MS Capabilities (X'80F0') MS Major Vector

CP → CP

This major vector is used to establish focal-point/entry-point relationships, as well as to convey current focal-point information.

MS Capabilities (X'80F0') MS Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'80F0'
4–n		MS subvectors, as described (using zero-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00'–X'7F', and immediately below for subvector keys X'80'–X'FE'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in MS Capabilities (X'80F0') Major Vector	
	CP-MSU	Notes
Focal Point Identification (X'21')	P	Note 1
Focal Point Authorization Request (X'61')	CP	Note 2
Focal Point Authorization Reply (X'62')	CP	Note 3
Entry Point Authorization Request (X'63')	CP	Note 4
Entry Point Authorization Reply (X'64')	CP	Note 5
Focal Point Notification (X'E1')	CP	Note 6

Key:

P Present one time (See Notes for conditions.)
CP Conditionally present one time (See Notes for conditions.)

Notes:

1. This subvector is always present and is always last for senders at the current level of SNA. Previous implementations, which are supported for migration, do not include this subvector. For those implementations, the identity of the focal point is understood to be the session partner that sends an MS Capabilities request. The only request supported by back-level focal-point implementations is the Focal Point Authorization Request (X'61') subvector. The only reply supported by back-level entry-point implementations is the Focal Point Authorization Reply (X'62') subvector.

2. This subvector is present when the request is sent from a focal point to an entry point. It is sent by a focal point to request that an entry point be in its sphere of control.
3. This subvector is present in the response to an MS Capabilities major vector containing the Focal Point Authorization Request (X'61') subvector. It is also present in an unsolicited MS Capabilities major vector sent to revoke a focal point.
4. This subvector is present when the request is sent from an entry point to a focal point. It is sent by an entry point to request services from a focal point.
5. This subvector is present in response to receipt of an MS Capabilities major vector containing an Entry Point Authorization Request (X'63') subvector.
6. This subvector is present when the request is sent from a network node server client end node.

Focal Point Notification (X'E1') MS Capabilities Subvector

The Focal Point Notification subvector flows from a network node server to a client end node. It, together with the Focal Point Identification (X'21') MS common subvector, indicates the name of the current focal point for the specified MS Category.

Focal Point Notification (X'E1') MS Capabilities Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Focal Point Notification subvector
1(=p)		Key: X'E1'

MS Parameter Major Vector Formats

The MS Parameter Major Vectors are defined as follows:

Text Data (X'1300') MS Parameter Major Vector

SSCP → PU, PU → SSCP, CP → CP

This MS parameter major vector accompanies one of three MS major vectors: Execute Command (X'8061'), Reply to Execute Command (X'0061') and Send Message to Operator (X'006F'). It transports one or more messages.

Text Data (X'1300') MS Parameter Major Vector

Byte	Bit	Content
0–1		Length (n + 1), in binary, of this MS major vector
2–3		Key: X'1300'
4–n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Message Data Parameters (X'1300') Major Vector		
	NMVT	CP-MSU	Notes
Qualified Message (X'0A')	CP(n)	CP(n)	Note 1
Self-Defining Text Message (X'31')	CP(n)	CP(n)	Note 2

Key:

CP(n) Conditionally present one or more times (See Notes for conditions.)

Notes:

1. This subvector is present one or more times when the Self-Defining Text Message subvector is not present. One of the two is required.
2. This subvector is present one or more times when the Qualified Message subvector is not present. One of the two is required.

Structured Data (X'1307') MS Parameter Major Vector

SSCP → PU, PU → SSCP, CP → CP

This MS parameter major vector accompanies one of six MS major vectors: Execute Command (X'8061'), Reply to Execute Command (X'0061'), Reply to Analyze Status (X'0062'), Reply to Query Resource Data (X'0063'), Reply to Test Resource (X'0064'), and Send Message to Operator (X'006F'). It transports one or more resource data items.

Structured Data (X'1307') MS Parameter Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'1307'
4–n		MS subvectors, as described (using 0-origin indexing) below for subvector keys X'80'–X'FE', and in "MS Common Subvectors" on page 9-325 for subvector keys X'00' - X'7F'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Structured Data (X'1307') Major Vector		
	NMVT	CP-MSU	Notes
Hierarchy/Resource List (X'05')	CP	CP	Note 1
Resource Data (X'80')	CP(n)	CP(n)	Note 2

Key:

CP Conditionally present one time. (See Notes for conditions.)
CP(n) Conditionally present one or more times. (See Notes for conditions.)

Notes:

1. This subvector is present when the Structured Data MS parameter major vector follows the Reply to Analyze Status (X'0062') MS major vector in order to identify the resource or set of resources associated with the subvectors in the Begin Data Parameters (X'130A') MS parameter major vector. Otherwise it may be optionally included to identify the content of the resource data transported in its associated Resource Data (X'80') subvector.
2. This subvector is present one or more times when the Structured Data MS parameter major vector follows the Execute Command (X'8061') MS major vector, Reply To Execute Command (X'0061') MS major vector, Reply To Query Resource Data (X'0063') MS major vector, Reply To Test Resource (X'0064') MS major vector or Send Message to Operator (X'006F') MS major vector.

Structured Data Subvectors

Resource Data (X'80') Structured Data MS Subvector

This Structured Data subvector transports data about a single resource.

Resource Data (X'80') Structured Data MS Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Resource Data subvector
1		Key: X'80'
2–p		Subfield containing either a resource name or a resource address and a subfield containing data related to the resource. Either the X'01' subfield or the X'10' subfield is always present. One of the remaining four subfields is also present. *Note:* The following subfield keys are used as indicated: (X'01') Resource Item Name (X'02') Resource Item Hex Value (X'03') Resource Item Character Value (X'04') Resource Item Integer Value (X'05') Resource Item Bit String Value (X'10') Resource Item Address

Resource Item Name (X'01') Resource Data Subfield

This subfield transports the name of the resource data item, i.e. a label.

Resource Item Name (X'01') Resource Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Item Name subfield
1		Key: X'01'
2−q		Resource Item Name - a string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types"

Resource Item Hex Value (X'02') Resource Data Subfield

This subfield transports hexadecimal data to be displayed as hex digits.

Resource Item Hex Value (X'02') Resource Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Item Hex Value subfield
1		Key: X'02'
2−q		Resource Item Hex Value - a string of hexadecimal bytes.

Resource Item Character Value (X'03') Resource Data Subfield

This subfield transports character data.

Resource Item Character Value (X'03') Resource Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Item Character Value subfield
1		Key: X'03'
2−q		Resource Item Character Value - a string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types"

Resource Item Integer Value (X'04') Resource Data Subfield

This subfield transports integer data.

Resource Item Integer Value (X'04') Resource Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Item Integer Value subfield
1		Key: X'04'
2−q		Resource Item Integer Value - a one to four byte integer value.

Resource Item Bit String Value (X'05') Resource Data Subfield

This subfield transports hexadecimal data to be displayed as a string of 1's and 0's.

Resource Item Bit String Value (X'05') Resource Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Item Bit String Value subfield
1		Key: X'05'
2−q		Resource Item Bit String Value - a string of hexadecimal bytes.

Resource Item Address (X'10') Resource Data Subfield

This subfield transports the hexadecimal name of the resource data item, i.e. an address.

Resource Item Address (X'10') Resource Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Item Address subfield
1		Key: X'10'
2−q		Resource Item Address - a string of hexadecimal bytes.

Transparent Coded Datastream (X'1309') MS Parameter Major Vector

SSCP → PU, PU → SSCP, CP → CP

This MS parameter major vector accompanies one of three MS major vectors: Execute Command (X'8061'), Reply to Execute Command (X'0061') and Send Message to Operator (X'006F'). It contains data whose structure is not defined by SNA Management Services, but which is known by the sending and receiving applications.

Transparent Coded Datastream(X'1309') MS Parameter Major Vector

Byte	Bit	Content
0−1		Length (n+1), in binary, of this MS major vector
2−3		Key: X'1309'
4−n		Data of an architecturally undefined structure.

Begin Data Parameters (X'130A') MS Parameter Major Vector

PU → SSCP, CP → CP

This MS parameter major vector accompanies one of three MS major vectors: Reply to Analyze Status (X'0062'), Reply to Query Resource Data (X'0063'), and and Reply to Test Resource (X'0064'). It serves as a starting delimiter for a sequence of other MS parameter major vectors, as well as transporting failure data itself.

Begin Data Parameters (X'130A') MS Parameter Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS major vector
2–3		Key: X'130A'
4–n		MS subvectors, as described below. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Begin Data Parameters (X'130A') Major Vector		
	NMVT	CP-MSU	Notes
Resource State (X'82')	CP	CP	Note 1
Probable Causes (X'93')	CP	CP	Note 1

Key:

CP Conditionally present one time. (See Note for conditions.)

Note:

1. This subvector is present one time whenever the Begin Data Parameters MS parameter major vector follows the Reply To Analyze Status (X'0062') MS major vector.

Begin Data Parameters Subvectors

Resource State (X'82') Begin Data Parameters MS Subvector

This Begin Data Parameters subvector transports the state of an analyzed resource or set of resources.

Resource State (X'82') Begin Data Parameters MS Subvector

Byte	Bit	Content
0		Length (x'03') in binary, of the Resource State subvector
1		Key: X'82'
2		A value indicating the resource state. X'00' no failure detected X'01' detected failure with failing resource isolated X'02' detected failure with location not isolated X'03' detected failure upstream from the managed resource set X'04' detected failure within the managed resource set X'05' detected failure downstream from the managed resource set

Probable Causes (X'93') Begin Data Parameters MS Subvector

This subvector contains one or more code points denoting probable causes of a failure. The probable causes appear in order of decreasing probability.

Note: The format of this subvector is defined under the Alert (X'0000') major vector.

End Parameter Data (X'130B') MS Parameter Major Vector

PU → SSCP, CP → CP

This MS parameter major vector accompanies one of three MS major vectors: Reply to Analyze Status (X'0062'), Reply to Query Resource Data (X'0063'), and and Reply to Test Resource (X'0064'). It serves as an ending delimiter for a sequence of other MS parameter major vectors.

End Parameter Data (X'130B') MS Parameter Major Vector

Byte	Bit	Content
0–1		Length (4), in binary, of this MS major vector
2–3		Key: X'130B'

Initiate Agent Request (X'1730') MS Parameter Major Vector

LU → LU

This MS parameter major vector accompanies the Request Initiation (X'8068') MS major vector. It contains an Initiate command and associated parameters.

Initiate Agent Request (X'1730') MS Parameter Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS parameter major vector
2–3		Key: X'1730'
4–n		MS GDS variables, as described below. *Note:* The following GDS variables may be used as indicated:

GDS Variable	Presence in Initiate Agent Request (X'1730') Parameter Major Vector	
	CP-MSU	Notes
Command (X'1732')	P	
Job Element Spec (X'1735')	P	

Key:

P Present one time

Initiate Agent Request GDS Variables

Command (X'1732') Initiate Agent Request GDS Variable

This GDS variable contains the command. It is always the first GDS variable in the parameter major vector.

Command (X'1732') Initiate Agent Request GDS Variable

Byte	Bit	Content
0−1		Length (p + 1), in binary, of the Command GDS variable
2−3		Key: X'1732'
4(=p)		A value that identifies the command: X'01' Initiate job element

Job Element Spec (X'1735') Initiate Agent Request GDS Variable

This GDS variable contains parameters for the job element to be initiated. It always follows the Command (X'1732') GDS variable in the Initiate Agent Request (X'1730') parameter major vector.

Job Element Spec (X'1735') Initiate Agent Request GDS Variable

Byte	Bit	Content
0–1		Length (p + 1), in binary, of the Job Element Spec GDS variable
2–3		Key: X'1735'
4–p		GDS variables containing the job element specifications. *Note:* The following GDS variables may be used as indicated and in the order shown: X'1736' Submission Type X'173A' Object Number X'1747' Object Disposition X'1749' Command Procedure Parameters

Submission Type (X'1736') Job Element Spec GDS variable

This GDS variable specifies the type of job element initiation the target agent is to perform. It is present one time.

Submission Type (X'1736') Job Element Spec GDS variable

Byte	Bit	Content
0–1		Length (q + 1), of the Submission Type GDS variable
2–3		Key: X'1736'
4(=q)		A value that identifies the type of job element submission: X'01' Run_CL (run command language)

Object Number (X'173A') Job Element Spec GDS variable

This GDS variable specifies the relative sequence of this job element in a series of job elements to be initiated. It is present one time.

Object Number (X'173A') Job Element Spec GDS variable

Byte	Bit	Content
0–1		Length (q + 1), of the Object Number GDS variable
2–3		Key: X'173A'
4–5(=q)		Two byte binary sequence number. Its value is always X'0001'.

Object Disposition (X'1747') Job Element Spec GDS variable

This GDS variable specifies whether the target agent should keep or delete the job element object after having initiated it. It is optionally present. When absent, the target agent determines the disposition of the object.

Object Disposition (X'1747') Job Element Spec GDS variable

Byte	Bit	Content
0–1		Length (q + 1) of the Object Disposition GDS variable
2–3		Key: X'1747'
4(=q)		A value that specifies the disposition of the object. X'01' Keep X'03' Delete

Command Procedure Parameters (X'1749') Job Element Spec GDS variable

This GDS variable carries one or more system specific parameters, and the codepage and character set from which their encoded, to be passed to the command procedure when initiated. It is optionally present one time.

Command Procedure Parameters (X'1749') Job Element Spec GDS variable

Byte	Bit	Content
0–1		Length (q + 1) of the Command Procedure Parameters GDS variable
2–3		Key: X'1749'
4–n		System Specific Parameters. *Note:* The following subfields are used in the order shown: X'10' CCSID X'20' Parameter Data

CCSID (X'10') Command Procedure Parameters Subfield

This subfield specifies a Coded_Character_Set_ID identifying the codepage and character set in which the Parameter Data (X'20') subfield(s) are encoded. It is present one time.

CCSID (X'10') Command Procedure Parameters Subfield

Byte	Bit	Content
0		Length (r+1) of the CCSID subfield
1		Key: X'10'
2−3(=r)		A two byte CCSID as documented in *Code Page Architecture - Stage 1, CPA-S1-AD-8909-00.*

Parameter Data (X'20') Command Procedure Parameters Subfield

This subfield specifies parameter data to be passed to the command procedure when initiated. It is present one or more times.

Parameter Data (X'20') Command Procedure Parameters Subfield

Byte	Bit	Content
0		Length (s+1) of the Parameter Data subfield
1		Key: X'20'
2−s		Up to 253 bytes of system specific parameter data encoded according to the codepage and character set identified in the CCSID (X'10') subfield.

Initiate Agent Report (X'1731') MS Parameter Major Vector

LU → LU

This MS parameter major vector accompanies the Initiation (X'0068') MS major vector. It contains the requested Initiate command and its associated report data.

Initiate Agent Report (X'1731') MS Parameter Major Vector

Byte	Bit	Content
0–1		Length (n+1), in binary, of this MS parameter major vector
2–3		Key: X'1731'
4–n		MS GDS variables, as described below. *Note:* The following GDS variables may be used as indicated:

GDS Variable	Presence in Initiate Agent Request (X'1731') Parameter Major Vector	
	CP-MSU	Notes
Command (X'1732')	P	
Report Data Prefix (X'1741')	P	
Job Element State (X'1739')	P	
Completion Report (X'1748')	CP	Note 1
Report Data Suffix (X'1746')	P	

Key:

P	Present one time
CP	Conditionally present one time. (See Notes for conditions.)

Notes:

1. This subvector is present one time whenever the agent does not encounter any SNA-registered exception conditions in processing the request MU (i.e. the job element was initiated or is scheduled for initiation). When present, the SNA Condition Report (X'1532') GDS variable is not present in the agent object.

Initiate Agent Report GDS Variables

Command (X'1732') Initiate Agent Report GDS Variable

This GDS variable contains the command. It is always the first GDS variable in the parameter major vector.

Command (X'1732') Initiate Agent Report GDS Variable

Byte	Bit	Content
0 – 1		Length (p + 1), in binary, of the Command GDS variable
2 – 3		Key: X'1732'
4(= p)		A value that identifies the command: X'01' Initiate job element

Report Data Prefix (X'1741') Initiate Agent Report GDS Variable

This GDS variable serves as a prefix for the report data. It immediately follows the Command (X'1732') GDS variable in the Initiate Agent Report (X'1731') parameter major vector.

Report Data Prefix (X'1741') Initiate Agent Report GDS Variable

Byte	Bit	Content
0–1		Length (p + 1), in binary, of the Report Data Prefix GDS variable
2–3(=p)		Key: X'1741'

Job Element State (X'1739') Initiate Agent Report GDS Variable

This GDS variable contains the state of the job element. It immediately follows the Report Data Prefix (X'1741') GDS variable.

Job Element State (X'1739') Initiate Agent Report GDS Variable

Byte	Bit	Content
0–1		Length (p + 1), in binary, of the Job Element State GDS variable.
2–3		Key: X'1739'
4(=p)		A value that identifies job element state. X'03' Accepted for initiation X'07' Ended (either sucessfully or not)

Completion Report (X'1748') Initiate Agent Report GDS Variable

This GDS variable contains the completion code of the initiated job element. When present, it immediately follows the Job Element State (X'1739') GDS variable.

Completion Report (X'1748') Initiate Agent Report GDS Variable

Byte	Bit	Content
0–1		Length (p + 1), in binary, of the Completion Report GDS variable
2–3		Key: X'1748'

Completion Report (X'1748') Initiate Agent Report GDS Variable

Byte	Bit	Content
4–5(=p)		Two byte binary completion code from the initiated job element, or zeros in the case when the job element's state is accepted for initiation.

Report Data Suffix (X'1746') Initiate Agent Report GDS Variable

This GDS variable serves as a suffix for the report data. It immediately follows either the Job Element State (X'1739') GDS variable or, when present, the Completion Report (X'1748') GDS variable.

Report Data Suffix (X'1746') Initiate Agent Report GDS Variable

Byte	Bit	Content
0–1		Length (p+1), in binary, of the Report Data Suffix GDS variable
2–3(=p)		Key: X'1746'

MS GDS Variable Formats

The MS GDS Variables are defined as follows:

Text Command (X'1320') GDS Variable

Text Command (X'1320') GDS Variable

Description:	The Text Command GDS variable contains one or more operator commands in text form destined for processing at an MS entry point. Neither the syntax nor the semantic of the text command is defined by MS.
Format:	Implementation-defined

Byte	Content
0-3	LLID Header
4-n	One or more Self-Defining Text Message (X'31') MS Common Subvectors

Partial Format Processing Method (X'1321') GDS Variable

Partial Format Processing Method (X'1321') GDS Variable

Description:	The Partial Format Processing Method GDS variable specifies the format type (for transparent coded datastream) or the display datastream type (for a display datastream) of the data enveloped by the sender in the MS partial format.
Condition of Presence:	Present only if either the Transparent Coded Datastream (X'1309') or Display Datastream (X'1322') GDS variables are present in the CP-MSU.

Figure 9-2. The Partial Format Processing Method GDS Variable

Structure Name	Struct Ref Page	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Subtab
Partial Format Processing Method	9-316	ID	1321	≥7	—	Y	N	1	—
Coded Datastream Proc. Method	9-316	T	10	3-18	0-1	—	—	0	—
Display Datastream Proc. Method	9-317	T	20	3-18	0-1	—	—	0	—

Coded Datastream Processing Method (X'10') Field

Description:	The Coded Datastream Processing Method identifies the format type of the binary data contained in an MS partial format.
Condition of Presence:	Present if the Transparent Coded Datastream (X'1309') GDS variable is present in the CP-MSU.
Format:	Character String, 01134-00500

Display Datastream Processing Method (X'20') Field

Description:	The Display Datastream Processing Method subvector identifies the format type of the display datastream contained in an MS partial format.
Condition of Presence:	Present if the Display Datastream (X'1322') GDS variable is present in the CP-MSU.
Format:	Character string, 01134-00500

Display Datastream (X'1322') GDS Variable

Display Datastream (X'1322') GDS Variable

Description:	The Display Datastream GDS variable contains a display datastream not defined by MS (but qualified by the Partial Format Processing Method (X'1321') GDS variable).
Format:	Character String. If a coded character set identification is required, it will be specified using the Default Character Set ID (X'32') MS common subvector.

Byte	Content
0-3	LLID Header
4-n	Display Datastream (not defined by MS)

Context-Identified Values (X'1323') GDS Variable

Context-Identified Values (X'1323') GDS Variable

Description: The Context-Identified Values GDS variable contains values identified by another special value that specifies the context in which the values should be interpreted. The values are not defined by MS.

Figure 9-3. The Context-Identified Values GDS Variable

Structure Name	Struct Ref Page	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Subtab
Context-Identified Values	9-318	ID	1323	≥13	—	N	Y	≥1	—
Value Processing Method	9-320	ID	1328	7-18	0-1	N	N	1-2	—
Value Type	9-320	T	01	3	1	—	—	0	—
Coded Character Set ID	9-321	T	02	6	0-1	—	—	0	—
National Language ID	9-321	T	12	5	0-1	—	—	0	—
Context Identifier Group	9-319	ID	1324	9-221	1	N	N	1-2	—
Value Processing Method	9-320	ID	1328	7-18	0-1	N	N	1-2	—
Value Type	9-320	T	01	3	1	—	—	0	—
Coded Character Set ID	9-321	T	02	6	0-1	—	—	0	—
National Language ID	9-321	T	12	5	0-1	—	—	0	—
Context Identifier	9-319	ID	1325	5-204	1	—	—	0	—
Value Group	9-319	ID	1326	≤31722	≥0	N	Y	≥1	—
Character Value Descriptor	9-319	ID	1327	5-20	0-1	—	—	0	—
Hexadecimal Value Descriptor	9-322	ID	132A	5-20	0-1	—	—	0	—
Value Instance Identifier	9-323	ID	1331	5-20	0-1	—	—	0	—
Nested Value Processing Method	9-322	ID	132B	7-24	0-1	N	N	1-2	—
Nested Value Type	9-322	T	01	3	1	—	—	0	—
Coded Character Set ID	9-321	T	02	6	0-1	—	—	0	—
Array Entry Length	9-323	T	03	3-6	0-1	—	—	0	—
National Language ID	9-321	T	12	5	0-1	—	—	0	—
Value	9-321	ID	1329	≤31713	0-1	—	—	≥0	—

Context Identifier Group (X'1324') GDS Variable

Context Identifier Group (X'1324') GDS Variable

Description:	The Context Identifier Group GDS variable contains a context identifier and, optionally, information describing how to process it.

Context Identifier (X'1325') GDS Variable

Context Identifier (X'1325') GDS Variable

Description:	The Context Identifier GDS variable contains a value that sets the context for other values in the Context-Identified Values (X'1323') GDS variable.
Condition of Presence:	The Context Identifier GDS variable is required, and must precede any Value Group (X'1326') GDS variables.

Value Group (X'1326') GDS Variable

Value Group (X'1326') GDS Variable

Description:	The Value Group GDS variable contains a value descriptor and a value, and optionally, processing information about the value.

Character Value Descriptor (X'1327') GDS Variable

Character Value Descriptor (X'1327') GDS Variable

Description:	The Character Value Descriptor GDS variable contains a short character text field describing the value.
Condition of Presence:	Required if Hexadecimal Value Descriptor (X'132A') GDS variable is not present. Not allowed if Hexadecimal Value Descriptor (X'132A') GDS variable is present.
Format:	Character String

Value Processing Method (X'1328') GDS Variable

Value Processing Method (X'1328') GDS Variable

Description:	The Value Processing Method GDS variable contains instructions as to how to process a value or context identifier.
Condition of Presence:	Optional. If *specified before* the Context Identifier Group (X'1324') GDS variable, it establishes a default processing method for the Context Identifier (X'1325') GDS variable and any related Value (X'1329') GDS variables in the Context-Identified Value (X'1323') GDS variable. If *not specified before* the Context Identifier Group (X'1324') GDS variable, a default processing method of type hexadecimal (X'01') is implied for the Context Identifier (X'1325') GDS variable and any related Value (X'1329') GDS variables in the Context-Identified Value (X'1323') GDS variable. If specified in a Context Identifier Group (X'1324') GDS variable, the default processing method, described above, is overridden for the associated Context Identifier (X'1325') GDS variable.

Value Type (X'01') Field

Description:	The Value Type field indicates what type of value is to be processed. For character values, the Coded Character Set ID (X'02') field is also present.

Byte	Content	
0-1	LT Header	
2	Value type. Defined codes are:	
	X'01'	Hexadecimal. Hexadecimal data is packaged as a string of 8-bit bytes.
	X'02'	Character. Character data is packaged as a string of characters. A single-byte character set (SBCS) has each character occupying a single byte. A double-byte character set (DBCS) has each character occupying two bytes where the most significant bit of a character code point is the leftmost bit position of a two-byte field, and the least significant bit of a character code point is the rightmost bit position of a two-byte field. A DBCS character set definition may permit transitions between DBCS and SBCS which results in a transition between the two sets of rules.
	X'03'	Unsigned decimal integer. An unsigned decimal integer is packaged as its hexadecimal equivalent.

X'04'	Bit string. A string of bits is encoded as if it were a binary number.
X'05'	Processing method implied by context identifier. A context-specific encoding scheme is defined by the particular context whenever such encoding applies.

Coded Character Set ID (X'02') Field

Description:	The Coded Character Set ID field identifies the character set and code page of the value.
Condition of Presence:	Present if the Value Type (X'01') subfield is "character" *and* one of the following is true: • The Default Character Set ID (X'32') MS common subvector *is not* present in the CP-MSU. • The Default Character Set ID (X'32') MS common subvector *is* present in the CP-MSU but does not apply to this value.
Format:	01134-00500

Byte	Content
0-1	LT Header
2-5	Coded character set ID: two 4-digit hexadecimal numbers that specify uniquely the coded character set in which the accompanying user text message is encoded. Receivers are responsible for documenting the coded character set IDs, as well as the coded character sets themselves, that they support in this subvector.
2-3	A 4-digit hexadecimal number identifying a character set
4-5	A 4-digit hexadecimal number identifying a code page

National Language ID (X'12') subfield

Description:	This subfield is defined in the description of the Self-Defining Text Message (X'31') MS Common Subvector.

Value (X'1329') GDS Variable

Value (X'1329') GDS Variable

Description:	The Value GDS variable contains a value not defined by MS.
Condition of Presence:	If there is only one Value Group (X'1326') GDS variable present, only one Value (X'1329') GDS may be present. If there is more than one Value Group (X'1326') GDS variable present (i.e., "nested" X'1326' GDS variables), there can be one X'1329' GDS variable per nested X'1326' GDS variable.

Hexadecimal Value Descriptor (X'132A') GDS Variable

Hexadecimal Value Descriptor (X'132A') GDS Variable

Description:	The Hexadecimal Value Descriptor GDS variable contains a short hexadecimal field describing the value.
Condition of Presence:	Required if Character Value Descriptor (X'1327') GDS variable is not present. Not allowed if Character Value Descriptor (X'1327') GDS variable is present.

Format: Hexadecimal string.

Nested Value Processing Method (X'132B') GDS Variable

Nested Value Processing Method (X'132B') GDS Variable

Description:	The Nested Value Processing Method GDS Variable contains instructions as to how to process the Value (X'1329') which follows it. The Value may, but not necessarily, contain nested Value Groups (X'1326').
Condition of Presence:	Optional. If present, it overrides the default that was set earlier by any previous occurance of the Value Processing Method (X'1328') GDS Variable in the Context-Identified Values (X'1323') GDS Variable.

Nested Value Type (X'01') Field

Description:	The Nested Value Type Field is similar to the Value type field. The difference occurs in the addition of another type which allows definition of the structure for the associated Value Variable (X'1329').

Byte	Content
0-1	LT Header
2	Nested Value Type. Defined Values are:
	X'01' Hexadecimal. Hexadecimal data is packaged as a string of 8-bit bytes.
	X'02' Character. Character data is packaged as a string of characters. A single-byte character set (SBCS) has each character occupying a single byte. A double-byte character set (DBCS) has each character occupying two bytes where the most significant bit of a character code point is the leftmost bit position of a two-byte field, and the least significant bit of a character code point is the rightmost bit position of a two-byte field. A DBCS character set definition may permit transitions between DBCS and SBCS which results in a transition between the two sets of rules.

X'03'	Unsigned decimal integer. An unsigned decimal integer is packaged as its hexadecimal equivalent.
X'04'	Bit string. A string of bits is encoded as if it were a binary number.
X'05'	Processing method implied by context identifier. A context-specific encoding scheme is defined by the particular context whenever such encoding applies.
X'06'	Value is composed of one or more Nested Value Group (X'1326') GDS Variables

Array Entry Length (X'03') Field

Description: The Array Entry Length Field, if present, indicates that the content of the associated Value (X'1329') GDS variable is a one-dimensional array of elements. Each element is equal in length to all other elements. In addition, the value of this field specifies the length (number of bytes) of each element in the array.

Byte	Content
0-1	LT Header
2-5	Unsigned integer value (hex) specifying length of array element.

National Language ID (X'12') Field

Description: This subfield is defined in the description of the Self-Defining Text Message (X'31') MS Common Subvector.

Value Instance Identifier (X'1331') GDS Variable

Value Instance Identifier (X'1331') GDS Variable

Description: The Value Instance Identifier GDS variable, when present, may be used to uniquely identify a Value Group (X'1326') GDS variable. The format of the content is implementation-specific and therefore not defined by SNA/MS.

Condition of Presence: Optional. If present, there is only one Value Instance Identifier for a particular Value Group.

Routing and Targeting Instructions (X'154D') GDS Variable

Routing and Targeting Instructions (X'154D') GDS Variable

Description: The Routing and Targeting Instructions GDS variable carries the Name List (X'06') MS Common Subvector which contains the names of one or more second-level resources. The named resources may constitute both the route an MDS-MU is to take to get to a second-level target (after reaching the MS entry point or focal point), as well as the target itself. Alternatively, the second-level target alone may be named. Additionally, the Correlator Control (X'0D') MS Common Subvector may be included to specify how the Agent Unit of Work Correlator (X'1549') GDS Variable in the MDS header (or an implementation-specific correlator carried in the Correlator Control subvector) is to be interpreted by the second-level application programs involved.

Byte	Content
0-3	LLID Header
4-n	MS subvectors, as described below.

Note: The following subvector keys may be used as indicated:

Subvector	Presence in Routing and Targeting Instructions (X'154D') GDS Variable	
	CP-MSU	Notes
Name List (X'06')	P	
Correlator Control (X'0D')	O	

Key:

P Present one time
O Optionally present one time

MS Common Subvectors

The common MS subvectors are defined as follows (using 0-origin indexing):

Text Message (X'00') MS Common Subvector

This MS common subvector transports EBCDIC data.

Text Message (X'00') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Text Message subvector
1		Key: X'00'
2−p		Text message in EBCDIC *Note:* The coded character set that may be transported in this field is dependent on the implementation that provided the text or allowed an operator to input the text, as well as the output device used by the Alert processor. The installation management ensures the compatibility of these products.

Date/Time (X'01') MS Common Subvector

This MS common subvector is used by PUMS or CPMS for time-stamping the NMVT or CP-MSU in which it is carried.

Date/Time (X'01') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Date/Time subvector
1		Key: X'01'
2–p		One or more of the following subfields: X'10' Local Date/Time (required subfield) X'20' Greenwich Mean Time Offset

Local Date/Time (X'10') Date/Time Subfield

This subfield transports the local date and time of the creation of the major vector.

Local Date/Time (X'10') Date/Time Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Local Date/Time subfield
1		Key: X'10'
2–4		Local date
2		Year, in binary, consisting of the last two digits of the year
3		Month, in binary (X'01'–X'0C')
4		Day, in binary (X'01'–X'1F')
5–q		Local time
5		Hours, in binary (X'00'–X'17')
6		Minutes, in binary (X'00'–X'3B')
7		Seconds, in binary (X'00'–X'3B')
8–q		Optional extension of time: a binary value to provide finer granularity than seconds

Greenwich Mean Time Offset (X'20') Date/Time Subfield

This subfield transports the Greenwich Mean Time (GMT) offset of the node that originated the management services RU (i.e., the origin node). It is optionally included in a major vector by the origin node or by the control point in whose domain the origin node resides.

Greenwich Mean Time Offset (X'20') Date/Time Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Greenwich Mean Time Offset subfield
1		Key: X'20'
2 – 3(=q)		Time zone adjustment to Greenwich Mean Time: an interval of time to be added to, or subtracted from, the local time given in the Local Date/Time (X'10') subfield to adjust that time to Greenwich Mean Time
	0	Positive or negative adjustment indicator: 0 adjustment to be added to the local time (i.e., all time zones westward, between the Greenwich time zone and the International Date Line) 1 adjustment to be subtracted from the local time (i.e., all time zones eastward, between the Greenwich time zone and the International Date Line)
	1 – 3	Reserved
	4 – 7	Number of hours of adjustment, in binary (X'0' – X'C')
	8 – 15	Number of minutes of adjustment, in binary (X'00' – X'3B')

Hierarchy Name List (X'03') MS Common Subvector

This MS common subvector identifies target resources, other than the reporting PU, that are within the same domain as the origin PU, but cannot be represented in the SNA Address List subvector.

Hierarchy Name List (X'03') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Hierarchy Name List subvector
1		Key: X'03'
2		Reserved
3		Number, in binary, of name entries in the hierarchy name list.
4−p		Hierarchy Name List Entries (1 to 5 entries may be present)
Note:		Each entry contains a Name field and a Resource Type field, and has the following form (shown 0-origin):
0		Length (q + 1), in binary, of the following name plus this Length field
1−q		Name of resource in upper-case alphanumeric EBCDIC characters *Note:* Resource name never exceeds eight characters.
q+1−q+4		Resource type identifier: category in which the resource (named in bytes 1−q) belongs: X'C1C4C1D7' adapter X'C2D9C4C7' LAN bridge X'C3C2E740' computerized branch exchange X'C3C2E4E2' carrier-sense multiple-access with collision detection (CSMA/CD) bus X'C3C8C1D5' channel X'D3C9D5C5' communication link X'C3E3D9D3' controller X'C4C9E2D2' disk X'C4E2D2E3' diskette X'D2E8C2C4' keyboard X'D3C1D540' local-area network (LAN) X'D3C3D6D5' link connection *Note:* This resource type is used for logical link connections not known to SNA, such as a LAN manager's connection with a management server. X'D3D6D6D7' loop X'D7C2D440' personal banking machine X'D7D6E240' point-of-sale unit X'D7C2E740' private branch exchange X'D7D3E3D9' plotter X'D7D9E3D9' printer

Hierarchy Name List (X'03') MS Common Subvector

Byte	Bit	Content	
		X'D9C9D5C7'	token-ring
		X'E2D74040'	service point
		X'E3F1D9D4'	T1 resource manager
		X'E3C1D7C5'	tape
		X'E3C1E440'	teller assist unit
		X'E3C2E4E2'	token bus
		X'E3C5D9D4'	terminal
		X'C4C5E540'	unspecified device

SNA Address List (X'04') MS Common Subvector

This MS common subvector is used in both request and data NMVTs. In a request NMVT, it identifies one or more destinations of the MS request when the destination is not the PU addressed in the transmission header (TH). In a data NMVT, it identifies the resource associated with the data when the resource is not the PU addressed in the TH.

If present, this subvector is the first subvector within the MS major vector.

SNA Address List (X'04') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the SNA Address List subvector
1		Key: X'04'
2		Address Count: For address entity format types X'00', X'40', X'80', and X'C0', a binary number indicating the number of individual addresses present in the X'04' subvector. This field is set to X'00' for all other address entity format types. *Note:* This field provides a count of individual addresses; thus, for format X'40', each pair of addresses counts as two.
3		Address entity format type: X'00' address format is one or more single local addresses X'40' address format is one or more pairs of session-partner local addresses, each pair identifying a session X'80' address format is one or more single network addresses X'A0' address format is one or more network-qualified address pairs, each pair identifying a session X'C0' address format is one or more pairs of session partner network addresses, each pair identifying a session
4−p		Address entities: one or more address entities, each having one of the formats defined below (0-origin):
For a single local address (byte 3 = X'00'):		
0−4		Reserved
5		Local address
For a pair of session-partner local addresses (byte 3 = X'40'):		
0−4		Reserved
5		Local address of SLU
6		Retired
7−11		Reserved
12		Session index (local address of PLU)

SNA Address List (X'04') MS Common Subvector

Byte	Bit	Content
For a single network address (byte 3 = X'80'):		
0–5		Network address
For a network-qualified address pair (byte 3 = X'A0'):		
0–5		Network address of NAU1
6–11		Network address of NAU2
12–19		Network ID of the subnetwork in which the above addresses are valid
For a pair of session-partner network addresses (byte 3 = X'C0'):		
0–5		Network address 1
6		X'80'
7–12		Network address 2

Hierarchy/Resource List (X'05') MS Common Subvector

This MS common subvector flows in management services units (MSU's) to communicate resource names between MS components in nodes. When flowing in an MSU to a focal point, the HRL includes the names of the resources of the domain hierarchy for the affected resource. The HRL containing the complete domain hierarchy is built from configuration knowledge in the control point and the TH. The Hierarchy/Resource List subvector also carries the hierarchy of control points that received, processed, and forwarded a CP-MSU, and, in some cases, the names of one or more resources. e.g., session partners, logically associated with the reported hierarchy. If the sender is a PU, then this subvector identifies resources, hierarchically below the sending PU, that cannot be represented in the SNA Address List subvector.

Hierarchy/Resource List (X'05') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Hierarchy/Resource List subvector
1		Key: X'05'
2−p		One or more subfields containing a hierarchical list of resources, or a list of resources of the same type (listed by key value below and described in detail following): X'01' Resource Identifier X'10' Hierarchy Name List X'11' Associated Resources X'20' Processing Node List

Resource Identifier (X'01') Hierarchy/Resource List Subfield

This subfield uniquely identifies a resource for which infomation is being reported.

Resource Identifier (X'01') Hierarchy/Resource List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Identifier subfield
1		Key: X'01'
2		Length of resource type field which follows (acceptable values range from 1 to 8)
3−m		Resource type - a string of 1 to 8 characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types"

Resource Identifier (X'01') Hierarchy/Resource List Subfield

Byte	Bit	Content
m+1		Length of resource name field which follows (acceptable values range from 1 to 8)
m+2−n(=q)		Name of resource - a string of 1 to 8 characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types"

Hierarchy Name List (X'10') Hierarchy/Resource List Subfield

This subfield contains a list specifying the names of resources in a hierarchy. The last name in the list specifies the resource to which the data present in the major vector pertains.

Hierarchy Name List (X'10') Hierarchy/Resource List Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Hierarchy Name List subfield
1		Key: X'10'
2		
	0	Hierarchy complete indicator. The hierarchy sent from a control point should always be complete, the hierarchy sent from a type 2 or type 4 node should never be complete. 0 The receiver must not modify the Hierarchy Name List (X'10') subfield 1 The receiver must modify the Hierarchy Name List (X'10') subfield by appending its name, the name of the link from this receiver to the sending PU, and the name of the sending PU; above the names already contained in the Hierarchy Name List subfield.
	1−7	Reserved, must be 0.
3−q		Hierarchy Name List Entries (entries left to right indicate resources down the hierarchy)
Note:		Each entry contains a Length field, a Name field, a Flags byte and a Resource Type field, and has the following form (shown 0-origin):
0		Length (r+1), in binary, of the following name plus this length field
1−r		Name of resource in upper-case alphanumeric EBCDIC characters *Note:* Resource name never exceeds eight characters.
r+1		Flags
	0	Reserved
	1	Display resource name indicator: 0 This name should be displayed if the receiver elects to display a single resource name and type as part of its presentation of the MSU containing this subvector.

Hierarchy Name List (X'10') Hierarchy/Resource List Subfield

Byte	Bit	Content
		1 This name should not be displayed if the receiver elects to display a single resource name and type as part of its presentation of the MSU containing this subvector.
	2	Resource list indexing flag: 0 Do not count this entry for resource list indexing, as defined in the Resource List Index (X'84') Network Alert common subfield 1 Do count this entry for resource list indexing
	3–7	Reserved
r+2		Resource type identifier: category to which the resource (named in bytes 1–r) belongs: X'00' unspecified device X'11' disk X'13' printer X'16' tape X'17' terminal X'18' transaction program X'19' program product X'1A' modem X'1B' operating system X'20' storage device X'21' adapter X'25' diskette X'27' loop X'29' keyboard X'2A' self-service terminal X'2B' plotter X'2C' transmission group X'2D' line group X'2E' token-ring X'2F' computerized branch exchange X'30' T1 resource manager X'31' private branch exchange X'32' carrier-sense multiple-access with collision detection (CSMA/CD) bus X'33' token bus X'34' printer server X'35' personal banking machine X'36' teller assist unit X'37' point-of-sale unit X'38' local controller X'39' local area network (LAN) X'3A' LAN bridge

Hierarchy Name List (X'10') Hierarchy/Resource List Subfield

Byte	Bit	Content
		X'3B' logical link connection *Note:* See also Resource Type Identifier X'F9' (link). Identifier X'3B' is used for logical link connections not known to SNA, such as a LAN manager's connection with a management server. Identifier X'F9' is used for logical link connections that are known to SNA.
		X'3C' management server
		X'3D' line
		X'3E' domain
		X'3F' port
		X'40' application
		X'41' relational data base
		X'42' requester
		X'43' server
		X'55' interconnect controller
		X'80' controller
		X'81' service point
		X'82' communication controller
		X'83' central processing unit
		X'84' programmable work station (PWS)
		X'85' reserved
		X'86' reserved
		X'E0'–X'EF' reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.
		X'F0' boundary function physical unit
		X'F1' physical unit
		X'F2' OSI management server
		X'F3' logical unit
		X'F4' control point
		X'F5' network ID
		X'F7' link station
		X'F8' SNA channel
		X'F9' link
		X'FA' D-channel
		X'FB' B-channel
		X'FC' SNMP device

Associated Resources (X'11') Hierarchy/Resource List Subfield

This subfield contains one or more entries identifying resources associated with the reported condition that may not be related hierarchically to the sender of the major vector. The entries are used by the receiver to correlate this major vector with other data associated with the resources that the entries identify. The resources identified in this subfield are not hierarchically related to each other.

Associated Resources (X'11') Hierarchy/Resource List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Associated Resources subfield
1		Key: X'11'
2		Reserved
3−q		Associated Resource Entries
Note:		Each entry contains a Length field, a Name field, a Flags byte, and a Resource Type field, and has the following form (shown 0-origin):
0		Length (r + 1), in binary, of the following name plus this length field
1−r		Name of resource in upper-case alphanumeric EBCDIC characters *Note:* Resource name never exceeds 56 characters.
r+1		Flags
	0−1	Reserved
	2	Resource list indexing flag: 0 Do not count this entry for resource list indexing, as defined in the Resource List Index (X'84') Network Alert common subfield 1 Do count this entry for resource list indexing
	3−7	Reserved
r+2		Resource type identifier: category to which the resource (named in bytes 1−r) belongs: X'00' unspecified device X'11' disk X'13' printer X'16' tape X'17' terminal X'18' transaction program X'19' program product X'1A' modem X'1B' operating system X'20' storage device X'21' adapter X'25' diskette X'27' loop

Associated Resources (X'11') Hierarchy/Resource List Subfield

Byte	Bit	Content
		X'29' keyboard
		X'2A' self-service terminal
		X'2B' plotter
		X'2C' transmission group
		X'2D' line group
		X'2E' token-ring
		X'2F' computerized branch exchange
		X'30' T1 resource manager
		X'31' private branch exchange
		X'32' carrier-sense multiple-access with collision detection (CSMA/CD) bus
		X'33' token bus
		X'34' printer server
		X'35' personal banking machine
		X'36' teller assist unit
		X'37' point-of-sale unit
		X'38' local controller
		X'39' local area network (LAN)
		X'3A' LAN bridge
		X'3B' logical link connection *Note:* See also Resource Type Identifier X'F9' (link). Identifier X'3B' is used for logical link connections not known to SNA, such as a LAN manager's connection with a management server. Identifier X'F9' is used for logical link connections that are known to SNA.
		X'3C' management server
		X'3D' line
		X'3E' domain
		X'3F' port
		X'40' application
		X'41' relational data base
		X'42' requester
		X'43' server
		X'55' interconnect controller
		X'80' controller
		X'81' service point
		X'82' communication controller
		X'83' central processing unit
		X'84' programmable work station (PWS)
		X'85' reserved
		X'86' reserved
		X'E0'–X'EF' reserved *Note:* This range of code points is reserved for use by non-IBM products and customer written applications. No IBM product will send a code point from within this range.
		X'F0' boundary function physical unit
		X'F1' physical unit
		X'F2' OSI management server

Associated Resources (X'11') Hierarchy/Resource List Subfield

Byte	Bit	Content
		X'F3' logical unit X'F4' control point X'F5' network ID X'F7' link station X'F8' SNA channel X'F9' link X'FA' D-channel X'FB' B-channel X'FC' SNMP device

Processing Node List (X'20') Hierarchy/Resource List Subfield

This subfield contains a list specifying the names of nodes that have received, processed, and forwarded an MSU. When required to add its name to the list, a node places its name at the front of the list.

Processing Node List (X'20') Hierarchy/Resource List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Processing Node List subfield
1		Key: X'20'
2		Reserved
3−q		Processing Node List Entries
Note:		Each entry contains a Length field, a Name field, a reserved byte and a Resource Type field, and has the following form (shown 0-origin):
0		Length (r + 1), in binary, of the following name plus this length field
1−r		Name of resource in upper-case alphanumeric EBCDIC characters *Note:* Resource name never exceeds eight characters.
r+1		Reserved
r+2		Resource type identifier: category to which the resource (named in bytes 1−r) belongs: X'EF' control point serving as a focal point

Name List (X'06') MS Common Subvector

This MS common subvector is used in requests, to identify one or more resources to which the request pertains. It may also contain the name of a network management application or network operator to which the receiver is to route the request.

Name List (X'06') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Name List subvector
1		Key: X'06'
2–p		One or more subfields containing a hierarchy and/or a list of peer resources (listed by key value below and described in detail following): X'01' Associated Resource Name List X'10' Cascaded Resource Name List X'30' Destination Name List X'50' Destination Application Name X'60' Origin Application Name X'70' Destination Instance Identifier X'80' Origin Instance Identifier

Associated Resource Name List (X'01') Name List Subfield

This subfield contains a list specifying the names of associated resources. The relationship among the resources is not defined.

Associated Resource Name List (X'01') Name List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Associated Resource Name List subfield
1		Key: X'01'
2–q		Associated Resource Name List Entries
Note:		Each entry contains Length and Name fields and has the following form (shown 0-origin):
0		Length (r + 1), in binary, of the following name field plus this length field

Associated Resource Name List (X'01') Name List Subfield

Byte	Bit	Content
1−r		Name of resource - a string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" *Note:* Resource name never exceeds eight characters. Trailing blanks are allowed; leading and embedded blanks are not.

Cascaded Resource Name List (X'10') Name List Subfield

This subfield contains a list specifying the names of cascaded resources. The last name in the list specifies the resource immediately above (hierarchically) the resources specified in the Destination Name List subfield present in the X'06' subvector.

Cascaded Resource Name List (X'10') Name List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Cascaded Resource Name List subfield
1		Key: X'10'
2		Reserved
3−q		Cascaded Resource Name List Entries
Note:		Each entry contains Length, Flag, and Name fields and has the following form (shown 0-origin):
0		Length (r + 1), in binary, of the following flag and name fields plus this length field
1		Flags:
	0	Resource indicator, always set to 0 in a request major vector. When this subfield is present in a data (reply) major vector, the bit is set as follows: 0 indicator off 1 indicator on *Note:* Sense Data (X'7D') subvector specifies the meaning of the resource indicator bit
	1	Reserved
	2	SNA resource indicator 0 this resource is not an SNA resource for which name-to-address translation is required 1 this resource is an SNA resource for which name-to-address translation is required
	3−7	Reserved

Cascaded Resource Name List (X'10') Name List Subfield

Byte	Bit	Content
2−r		Name of resource in upper-case alphanumeric EBCDIC characters *Note:* Resource name never exceeds eight characters. Trailing blanks are allowed; leading and embedded blanks are not.

Destination Name List (X'30') Name List Subfield

This subfield contains a list specifying the names of peer resources. The request present in the major vector pertains to the resources named in this list.

Destination Name List (X'30') Name List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Destination Name List subfield
1		Key: X'30'
2−q		Destination Name List Entries
Note:		Each entry contains Length, Flag, and Name fields and has the following form (shown 0-origin):
0		Length (r + 1), in binary, of the following flag and name fields plus this length field
1		Flags:
	0	Resource indicator, always set to 0 in a request major vector. When this subfield is present in a data (reply) major vector, the bit is set as follows: 0 indicator off 1 indicator on *Note:* Sense Data (X'7D') subvector specifies the meaning of the resource indicator bit
	1	Session partner indicator 0 this resource is not a session partner with the next resource in this list 1 this resource is a session partner with the next resource in this list
	2	SNA resource indicator 0 this resource is not an SNA resource for which name-to-address translation is required 1 this resource is an SNA resource for which name-to-address translation is required
	3−7	Reserved
2−r		Name of resource in upper-case alphanumeric EBCDIC characters *Note:* Resource name never exceeds eight characters. Trailing blanks are allowed; leading and embedded blanks are not.

Destination Application Name (X'50') Name List Subfield

This subfield identifies either a network management application or a network/system operator at the destination to which the request is to be routed.

Destination Application Name (X'50') Name List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Destination Application Name subfield
1		Key: X'50'
2−q		Name of destination application (or network/system operator) - a string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" *Note:* Application name never exceeds eight characters. Trailing blanks are allowed; leading and embedded blanks are not.

Origin Application Name (X'60') Name List Subfield

This subfield identifies the application program that is the originator of the request or report in which this subfield is contained.

Origin Application Name (X'60') Name List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Origin Application Name subfield
1		Key: X'60'
2−q		Name of origin application program - a string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" *Note:* Application program name never exceeds eight characters. Trailing blanks are allowed; leading and embedded blanks are not.

Destination Instance Identifier (X'70') Name List Subfield

This subfield identifies an instance of an application program named in the Destination Application Name (X'50') subfield.

Destination Instance Identifier (X'70') Name List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Destination Instance Identifer subfield
1		Key: X'70'
2−q		Name of the destination application program instance - a string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" *Note:* Application program instance name never exceeds eight characters. Trailing blanks are allowed; leading and embedded blanks are not.

Origin Instance Identifier (X'80') Name List Subfield

This subfield identifies an instance of an application program named in the Origin Application Name (X'60') subfield.

Origin Instance Identifier (X'80') Name List Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Origin Instance Identifer subfield
1		Key: X'80'
2−q		Name of the origin application program instance - a string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" *Note:* Application program instance name never exceeds eight characters. Trailing blanks are allowed; leading and embedded blanks are not.

Configuration Name (X'07') MS Common Subvector

This MS common subvector identifies a set of resources at the receiving node to which the command applies. For example the configuration name could be one of the identification tokens in the SNA/File Services global name of a data object containing resource identification information and instructions to perform the command. If absent, applicability to the entire receiving node is implied.

Configuration Name (X'07') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Configuration Name subvector
1		Key: X'07'
2−p		Name of the configuration - a string of characters (with no leading, imbedded or trailing blanks) from Coded Graphic Character Set 01134 − 00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types." *Note:* The name of the configuration never exceeds sixteen characters.

Reporting Level (X'09') MS Common Subvector

This MS common subvector is used to specify the type of reporting to be provided by the target node in reply to the command. If absent, architecturally defined defaults are implied.

Reporting Level (X'09') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Reporting Level subvector
1		Key: X'09'
2−p		One or more instances of the Reporting Requirement (X'10') subfield. Each instance identifies the specification for a particular type of report. The defaults are implied for report types not specified in an instance of the subfield.
Note:		If the target node is incapable of recognizing this subvector, the target node determines the type of reporting to be provided in reply to the command, whether or not this subvector is present. Reporting Level defaults do not apply in this case.

Reporting Requirement (X'10') Reporting Level Subfield

This subfield identifies the specification for a particular type of report.

Reporting Requirement (X'10') Reporting Level Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Reporting Requirement subfield
1		Key: X'10'
2		Report Classification: the type of report on which the specification is being given - it is one of the following values: X'10' Rejection X'20' Acceptance X'30' Success X'40' Failure X'50' Intermediate X'60' Ancillary
3(=q)		Report Specification: the specification for the identified type of report - it is one of the following values: X'10' Required (default for rejection, success and failure reports) X'20' Allowed (default for acceptance reports) X'30' Not Allowed (default for intermediate and ancillary reports)

Qualified Message (X'0A') MS Common Subvector

This MS common subvector contains a formatted identifier for a message stored at the receiver. It also contains a number of replacement text strings to be inserted into the message. The particular message being indexed determines how many text strings are included.

Qualified Message (X'0A') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Qualified Message subvector
1		Key: X'0A'
2–p		Subfields containing a formatted message identifier, and possibly one or more strings of text to be inserted into the message indexed by the identifier. *Note:* The following subfield keys are used as indicated:

Subfield	Presence in Qualified Message (X'0A') Common Subvector	
Message ID (X'01')	P	
Replacement Text (X'02')	CP(n)	Note 1

Key:

P Present one time
CP(n) Conditionally present one or more times (See notes for conditions.)

Notes:

1. The number of instances of this subfield present in the X'0A' subvector is determined by the number of strings of text required for insertion into the message indexed by the X'01' subfield.

Message ID (X'01') Qualified Message Subfield

This subfield contains a formatted identifier that indexes a message stored at the receiver. The exact format of the identifier is at the discretion of the sending and receiving applications.

Message ID (X'01') Qualified Message Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Message ID subfield
1		Key: X'01'

Message ID (X'01') Qualified Message Subfield

Byte	Bit	Content
2−q		Message ID: A string of characters from Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," identifying a text message stored at the receiver. The format of the message ID is left up to the discretion of the sender and the receiver.

Replacement Text (X'02') Qualified Message Subfield

This subfield transports replacement text, to be substituted by the receiver into the message indexed by the Message ID (X'01') subfield.

Replacement Text (X'02') Qualified Message Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Text Message subfield
1		Key: X'02'
2−q		Replacement text: A string of characters from Coded Graphic Character Set 00640−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," to be substituted into the message indexed by the Message ID (X'01') subfield

Execution Window Timing (X'0B') MS Common Subvector

This MS common subvector is used to identify the earliest or latest time at which the target node may begin execution the command. If both times are specified, an explicit time window is established. If either the earliest time, or latest time, is specified, an implicit time window is established. If this subvector is absent, immediate command execution without repetition is requested.

Execution Window Timing (X'0B') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Execution Window Timing subvector
1		Key: X'0B'
2−p		An execute-not-before time subfield and/or execute-not-after time subfield and possibly a time interval subfield on which to base repeated executions. *Note:* The following subfield keys are used as indicated:

Subfield	Presence in Execution Window Timing (X'0B') Common Subvector	
Execute Not Before Time (X'10')	CP	Note 1
Execute Not After Time (X'20')	CP	Note 2
Time Interval (X'30')	O	
Execution Count (X'40')	CP	Note 3

Key:

O Optionally present one time
CP Conditionally present one time (See notes for conditions.)

Notes:

1. This subfield is present if the Execute Not After Time (X'20') subfield is not present. Otherwise, it is optional.
2. This subfield is present if the Execute Not Before Time (X'10') subfield is not present. Otherwise, it is optional.
3. This subfield is optionally present if the Time Interval (X'30') subfield is present. Otherwise, it is not present.

Execute Not Before Time (X'10') Execution Window Timing Subfield

This subfield identifies the earliest date and time that the command may be executed. It may not specify a time later than the the Execute Not After Time (X'20') subfield, if present.

Execute Not Before Time (X'10') Execution Window Timing Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Execute Not Before Time subfield
1		Key: X'10'
2–3		Year in binary
4		Month in binary
5		Day in binary
6		Hour in binary
7		Minute in binary
8(=q)		GMT or local time indicator - it is one of the following values: X'10' Local Time X'20' Greenwich Mean Time (GMT)

Execute Not After Time (X'20') Execution Window Timing Subfield

This subfield identifies the latest date and time that the command may be executed. It may not specify a time earlier than the the Execute Not Before Time (X'10') subfield, if present.

Execute Not After Time (X'20') Execution Window Timing Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Execute Not After Time subfield
1		Key: X'20'
2–3		Year in binary
4		Month in binary
5		Day in binary
6		Hour in binary
7		Minute in binary

Execute Not After Time (X'20') Execution Window Timing Subfield

Byte	Bit	Content
8(=q)		GMT or local time indicator - it is one of the following values: X'10' Local Time X'20' Greenwich Mean Time (GMT)

Time Interval (X'30') Execution Window Timing Subfield

This subfield identifies the base time and a time interval on which repetitive executions of the command are to be established. Repetitive executions may be established for explicit or implicit execution windows. In either case, after the command is executed, the time interval is added to the base, thus establishing a new base for the next execution. If repetitive execution windows are specified, the ending time of a window may not occur after the beginning time of the next window.

Time Interval (X'30') Execution Window Timing Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Time Interval subfield
1		Key: X'30'
2		Binary number of days in the interval
3		Binary number of hours in the interval
4(=q)		Binary number of minutes in the interval

Execution Count (X'40') Execution Window Timing Subfield

This subfield specifies the number of times the command is to be executed. It is used only in conjunction with the Time Interval (X'30') subfield. When not present, the number of execution iterations according to a specified time interval is considered to be indefinite until the request execution is cancelled by a command, or as the result of a pre-set timer value having been reached, or by some other means.

Execution Count (X'40') Execution Window Timing Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Execution Count subfield

Execution Count (X'40') Execution Window Timing Subfield

Byte	Bit	Content
1		Key: X'40'
2−3(=q)		Binary execution count

Correlator Control (X'0D') MS Common Subvector

The Correlator Control MS Common Subvector indicates how an Agent Unit of Work Correlator (AUOWC, X'1549') GDS variable, or an implementation-defined correlator, is to be interpreted/used by the second-level application programs named in the Routing and Targeting Instructions (X'154D') GDS Variable. Indications are given as to whether, for a particular request (or reply), it is the first use of the correlator by the application programs, or not the first. Similarly, an indication is given as to whether it is the last (or not last) use of the correlator by these application programs. In this way, requests and their associated replies (both immediate and delayed) can be tracked as second-level application program units of work without affecting MDS Transport units of work. This subvector also indicates where the correlator is located.

Correlator Control (X'0D') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Correlator Control subvector
1		Key: X'0D'
2–p		One or more subfields (listed by Key value below and described in detail following): X'01' Application Correlator Control X'02' Application Correlator *Note:* The following subfield keys may be used as indicated:

Subfield	Presence in Correlator Control (X'0D') MS Common Subvector	
Application Correlator Control (X'01')	P	
Application Correlator (X'02')	CP	Note 1

Key:

P Present one time
CP Conditionally present one time

Notes:

1 Present if the Application Correlator Control subfield indicates that the AUOWC is not in the MDS header.

Application Correlator Control (X'01') Correlator Control Subfield

This subfield contains indicators specifying the usage of the correlator by a second-level application program.

Application Correlator Control (X'01') Correlator Control Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Application Correlator Control subfield
1		Key: X'01'
2(=q)		Correlation Indicators:
	0	Correlation First Message Indicator 0 Not the first use of the correlator 1 First use of the correlator
	1	Correlation Last Message Indicator 0 Not the last use of the correlator 1 Last use of the correlator
	2	Correlation Iteration Indicator 0 Not the last use of the correlator for the current execution iteration (report); or, not part of any iteration (reply) 1 Last use of the correlator for the current execution iteration (report) **Note:** Bit offset 2 is significant only when bit offset 1 = 0. When bit offset 1 = 1, the correlator is not used again, so iteration distinction has no meaning.
	3	Correlator Location Indicator 0 Correlator is an AUOWC and is found in the MDS header of the MDS-MU containing this subvector 1 Correlator (AUOWC or implementation-defined) is found in the Application Correlator (X'02') subfield of this subvector.
	4–7	reserved

Application Correlator (X'02') Correlator Control Subfield

The Application Correlator subfield contains a second-level application correlator, which may be an AUOWC. If it is not an AUOWC, then the format of this correlator is defined and agreed to by the communicating products.

Application Correlator (X'02') Correlator Control Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Application Correlator subfield
1		Key: X'02'
2–q		Implementation-defined second-level application correlator, or an AUOWC.

Product Set ID (X'10') MS Common Subvector

This MS common subvector identifies one or more products that implement a network component.

Product Set ID (X'10') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Product Set ID subvector
1		Key: X'10'
2		Retired
3 − p		Network product identifier consisting of one or more Product ID (X'11') MS common subvectors, as described below (using 0-origin indexing). Each Product ID (X'11') MS Common Subvector uniquely identifies a product. Products fall into two categories: hardware (with or without microcode) and software.

Product Identifier (X'11') MS Common Subvector

This MS common subvector uniquely identifies a single product. A product may consist of electronic circuitry (hardware), executable instructions (software), or both (in the case of hardware containing microcode).

Product Identifier (X'11') MS Common Subvector

Byte	Bit	Content
0		Length (q + 1), in binary, of the Product Identifier subvector
1		Key: X'11'
2	0–3	Reserved
	4–7	Product classification: X'1' IBM hardware X'3' IBM or non-IBM hardware (not distinguished) X'4' IBM software X'9' non-IBM hardware X'C' non-IBM software X'E' IBM or non-IBM software (not distinguished)
3–q		One or more subfields containing product- and installation-specific information on hardware, microcode, and software. *Note:* The subfields may be used as indicated in the table on the following page.

Conditions of Subfield Presence in Product Identifier Subvector							
Subfield	X'11' Type (Note 1)	X'0000' Alert (Note 2)		X'0090' QPI	XID3 (Note 3)	FMH7 for LU 6.2	Notes
		Sender	Resource				
X'00'	HW	P	P	P	P	P	
X'01'	HW	CP	CP	CP	CP	CP	Note 4
X'02'	SW	CP	CP	CP	CP	CP	Note 5
X'04'	SW	CP	CP	CP	CP	CP	Note 6
X'06'	SW	P	P	P	O	P	
X'07'	SW	–	CP	CP	O	CP	Note 7
X'08'	SW	CP	CP	CP	CP	CP	Note 6
X'09'	SW	–	CP	CP	O	CP	Note 7
X'0B'	HW	O	O	O	O	O	
X'0E'	HW	O	O	O	O	O	
X'0F'	SW or HW	–	O	O	O	O	Note 8

Key:

–	Not present
P	Present one time
CP	Conditionally present one time
O	Optionally present one time

Subfield Names:

X'00' – Hardware Product Identifier
X'01' – Emulated Product Identifier
X'02' – Software Product Serviceable Component Identifier
X'04' – Software Product Common Level
X'06' – Software Product Common Name
X'07' – Software Product Customization Identifier
X'08' – Software Product Program Number
X'09' – Software Product Customization Date and Time
X'0B' – Microcode EC Level
X'0E' – Hardware Product Common Name
X'0F' – Vendor Identification

Notes:

1. The hardware (HW) X'11' Product Identifier subvector is present when the Product Classification field (byte 2, bits 4–7) is X'1', X'3', or X'9'. The software (SW) X'11' Product Identifier subvector is present when this field is X'4', X'C', or X'E'.

2. If a PU is sending an Alert for itself, a single Product Set ID (X'10') subvector is present. This is the "Indicated Resource" for purposes of reading this matrix. If the PU is reporting on an Alert for an attached device, two X'10' subvectors are present, in the following order:

   a. "Alert Sender"—identifies the PU sending the Alert

b. "Indicated Resource"—identifies the resource that the Alert is reporting upon

3. In XID3, the Hardware and Software X'11' subvectors are carried in the X'10' control vector rather than the X'10' MS Common subvector.
4. This subfield is present in the hardware X'11' when a product is emulating another hardware product.
5. This subfield is present in the software X'11' for IBM products assigned a component ID by the IBM National Service Division. For products not assigned a component ID, the X'04' and X'08' subfields are present. See Note 6.
6. The X'04' and X'08' subfields are present in the software X'11' if the X'02' subfield is not present. They are optional when the X'02' is present. See Note 5. If, however, the software identified is a customer-written application, only the X'08' subfield is present.
7. At least one of the X'07' and X'09' subfields is required in the software X'11' for software products modified by the customer.
8. The X'0F' subfield is present for non-IBM products. Therefore, the X'0F' subfield must be present if bits 4−7 of byte 2 of the Product Identifier (X'11') MS Common Subvector are X'9' or X'C'.

Note: Unless otherwise indicated, characters in these subfields are to be decoded using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."

Hardware Product Identifier (X'00') Product ID Subfield

This subfield uniquely identifies an instance of a hardware product.

Hardware Product Identifier (X'00') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Hardware Product Identifier subfield
1		Key: X'00'
2		Format type: X'10' product instance is identified by a serial number (i.e., plant of manufacture and sequence number) unique by machine type X'11' product instance is identified by a serial number (i.e., plant of manufacture and sequence number) unique by machine type and model number X'12' product instance is identified by a serial number (i.e., plant of manufacture and sequence number) unique by machine type (as in format X'10' above). This format provides the model number not to uniquely identify a product instance but, for the purpose of additional information only. X'13' retired

Hardware Product Identifier (X'00') Product ID Subfield

Byte	Bit	Content
		X'20' product instance is identified by a repair ID number (i.e., plant of manufacture and sequence number) unique by machine type X'21' product instance is identified by a repair ID number (i.e., plant of manufacture and sequence number) unique by machine type and model number X'22' product instance is identified by a repair ID number (i.e., plant of manufacture and sequence number) unique by machine type (as in format X'10' above). This format provides the model number not to uniquely identify a product instance but for the purpose of additional information only. X'40' retired X'41' retired
3–r		Product identification *Note:* The originator of a message unit (e.g., NMVT, XID), reporting for another product that does not supply information required for the Hardware Product Identifier subfield, inserts binary 0's into the appropriate fields (except for the Machine Type field where EBCDIC 0's [X'F0'] are inserted) of the Product Identification field to indicate that no identification information is available.
Format X'10'		
3–6		Machine type: four numeric characters
7–8		Plant of manufacture: two characters
9–15(=r)		Sequence number: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left
Format X'11'		
3–6		Machine type: four numeric characters
7–9		Machine model number: three characters
10–11		Plant of manufacture: two characters
12–18(=r)		Sequence number: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left
Format X'12'		
3–6		Machine type: four numeric characters
7–9		Machine model number: three characters
10–11		Plant of manufacture: two characters
12–18(=r)		Sequence number: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left
Format X'20'		
3–6		Machine type: four numeric characters
7–8		Plant of manufacture: two characters

Hardware Product Identifier (X'00') Product ID Subfield

Byte	Bit	Content
9–15(=r)		Sequence number: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left
Format X'21'		
3–6		Machine type: four numeric characters
7–9		Machine model number: three characters
10–11		Plant of manufacture: two characters
12–18(=r)		Sequence number: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left
Format X'22'		
3–6		Machine type: four numeric characters
7–9		Machine model number: three characters
10–11		Plant of manufacture: two characters
12–18(=r)		Sequence number: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left

Emulated Product Identifier (X'01') Product ID Subfield

This subfield identifies the hardware of the product being emulated in sufficient detail to allow problem determination.

Emulated Product Identifier (X'01') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Emulated Product Identifier subfield
1		Key: X'01'
2–5		Machine type of product being emulated: four numeric characters
6–8(=r)		Model number of product being emulated: three characters

Software Product Serviceable Component Identifier (X'02') Product ID Subfield

This subfield transports the serviceable component identifier and release level as assigned by service personnel.

Software Product Serviceable Component Identifier (X'02') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Software Product Serviceable Component Identifier subfield
1		Key: X'02'
2–10		Serviceable component identifier: nine characters
11–13(=r)		Serviceable component release level: three numeric characters

Software Product Common Level (X'04') Product ID Subfield

This subfield transports the common version, release, and modification level numbers as given in the product announcement documentation.

Software Product Common Level (X'04') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Software Product Common Level subfield
1		Key: X'04'
2–3		Common version identifier: numeric characters, right-justified with X'F0' fill on left
4–5		Common release identifier: numeric characters, right-justified with X'F0' fill on left
6–7(=r)		Common modification identifier: numeric characters, right-justified with X'F0' fill on left

Software Product Common Name (X'06') Product ID Subfield

This subfield transports the software common name as given in the product announcement documentation.

Software Product Common Name (X'06') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Software Product Common Name subfield
1		Key: X'06'

Software Product Common Name (X'06') Product ID Subfield

Byte	Bit	Content
2−r		Up to thirty characters identifying the software product common name. The name is to be decoded using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," plus three additional code points: X'48' = "." (period); X'60' = "-" (minus sign); X'61' = "/" (slash).

Software Product Customization Identifier (X'07') Product ID Subfield

This subfield identifies a set of executable instructions, customized to the user's environment.

Software Product Customization Identifier (X'07') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Software Product Customization Identifier subfield
1		Key: X'07'
2−r		Customization identifier: up to eight characters

Software Product Program Number (X'08') Product ID Subfield

This subfield transports either the program product number as assigned by distribution personnel, or a substitute value supplied by a user-written software program.

Software Product Program Number (X'08') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Software Product Program Number subfield
1		Key: X'08'

Software Product Program Number (X'08') Product ID Subfield

Byte	Bit	Content
2−8(=r)		Program product number: seven characters *Note for Basic Alert (X'91' subvector) senders only*: A user-written application program does not send a program product number in this field. Instead it sends one of 16 substitute values comprised of seven characters from Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," having the following form: characters 1−4 are the letters USER; character 5 is one of the characters 0−9, or A-F; characters 6−7 are space (X'40') characters. Installation managers have the sole responsibility for managing the usage of these substitute values within their networks.

Software Product Customization Date and Time (X'09') Product ID Subfield

This subfield identifies the date and time that a set of executable instructions was customized to the user's environment.

Software Product Customization Date and Time (X'09') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Software Product Customization Date and Time subfield.
1		Key: X'09'
2		Year in unsigned packed decimal (i.e., one hex digit for each decimal digit)
3−4		Julian day in unsigned packed decimal, right-justified with 0's as fill
5		Hour in unsigned packed decimal (24-hour clock)
6(=r)		Minute in unsigned packed decimal

Microcode EC Level (X'0B') Product ID Subfield

This subfield identifies the engineering change (EC) level of the failing microcode component (e.g., microcode feature EC level or microcode subsystem level such as channel, power, or storage)

Microcode EC Level (X'0B') Product ID Subfield

Byte	Bit	Content
0		Length (r+1), in binary, of the Microcode EC Level subfield.

Microcode EC Level (X'0B') Product ID Subfield

Byte	Bit	Content
1		Key: X'0B'
2–r		Microcode EC Level: up to eight characters

Hardware Product Common Name (X'0E') Product ID Subfield

This subfield provides the hardware common name as given in the product announcement documentation.

Hardware Product Common Name (X'0E') Product ID Subfield

Byte	Bit	Content
0		Length (r + 1), in binary, of the Hardware Product Common Name subfield
1		Key: X'0E'
2–r		Up to fifteen characters identifying the hardware product common name. The name is to be decoded using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," plus three additional code points: X'48' = "." (period); X'60' = "-" (minus sign); X'61' = "/" (slash).

Vendor Identification (X'0F') Product ID Subfield

This subfield provides the vendor name, e.g., name of corporation.

Vendor Identification (X'0F') Product ID Subfield

Byte	Bit	Content
0		Length (r + 1), in binary, of this subfield.
1		Key: X'0F'
2–r		Up to 16 characters identifying the vendor.

Focal Point Identification (X'21') MS Common Subvector

The Focal Point Identification subvector is included in all MS Capabilities (X'80F0') major vector requests and replies to indicate the current focal point (NETID, NAU name, and application program name) and to optionally identify the backup focal point (NETID, NAU name, and application program name). This subvector is always the last subvector in the major vector.

Focal Point Identification (X'21') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Focal Point Identification subvector
1		Key: X'21'
2–p		The following subfields as indicated:

Subfield	Presence in Focal Point Identification (X'21') MS Common Subvector	
MS Category (X'01')	P	
Focal Point Identification Flags (X'02')	P	
Focal Point NETID (X'10')	P	Note 1
Focal Point NAU Name (X'11')	P	Note 1
Focal Point Application Name (X'12')	P	Note 1
Backup Focal Point NETID (X'20')	O	Note 2
Backup Focal Point NAU Name (X'21')	O	Note 2
Backup Focal Point Application Name (X'22')	O	Note 2

Key:

P	Present one time
CP	Conditionally present one time (See Notes for conditions.)
O	Optionally present one time (See Notes for conditions.)

Notes:

1. These subfields together represent the identification of the focal point. They are present in the following order in relation to each other: X'10' is always first, X'11' is always second, and X'12' is always third. These subfields are present in every MS Capabilities major vector. These subfields may be empty to indicate there is no current focal point, but they may not be omitted. An empty subfield is either one with a length of 2 (consisting of just the length and key

fields), or one that contains a name with no significant characters (all space characters).

2. These subfields together represent the identification of the backup focal point. They are present in the following order in relation to each other when they are present: X'20' is always first, X'21' is always second, and X'22' is always third. They are optionally present in MS Capabilities exchanges containing the X'61' subvector to convey backup focal point information to an entry point. They are not present with the X'62', X'63', X'64', or X'E1' subvectors.

MS Category (X'01') Focal Point Identification Subfield

This subfield specifies the MS Category for which authorization is being requested.

MS Category (X'01') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the MS Category subfield
1		Key: X'01'
2−q		MS Category code: indicates the category of management services that applies to the focal point in this subvector. Either a 1-to-8 character installation-defined application name, defined using Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," or one of the 4-byte architecturally-defined values for management services application programs that use MS Capabilities exchanges, listed in *SNA/Management Services Reference*. Trailing space (X'40') characters may be present, but are insignificant. Leading or embedded space characters may not be present.

Focal Point Identification Flags (X'02') Focal Point Identification Subfield

This subfield contains flags related to focal point identification.

Focal Point Identification Flags (X'02') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Focal Point Identification Flags subfield
1		Key: X'02'

Focal Point Identification Flags (X'02') Focal Point Identification Subfield

Byte	Bit	Content
2−3(=q)		Flags
	0	Backup focal point **Note:** This flag is defined only when the Focal Point Identification (X'21') subvector is present in an MS Capabilities major vector containing the Focal Point Authorization Request (X'61') subvector. Otherwise, this bit is reserved. 0 normal processing: backup focal point identified by the X'20', X'21', and X'22' subfields is accepted. If backup focal-point subfields are not present, current backup focal-point name is deleted. 1 keep current backup focal point, if any. This flag is used if no X'20', X'21', and X'22' subfields are present, but the sender does not want the receiver to delete backup focal-point entry, if there are any.
	1−15	Reserved

Focal Point NETID (X'10') Focal Point Identification Subfield

This subfield contains the NETID portion of the network-qualified name of the focal point.

Focal Point NETID (X'10') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Focal Point NETID subfield
1		Key: X'10'
2−q		A 1-to-8 character NETID (encoded using Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types"). Trailing space characters may be present, but are insignificant. Leading or embedded space characters may not be present. This subfield may be empty to indicate there is no current focal point. An empty subfield is either one with a length of 2 (consisting of just the Length and Key fields), or one that contains a NAU Name field with no significant characters (all space characters).

Focal Point NAU Name (X'11') Focal Point Identification Subfield

This subfield contains the unqualified portion of the NAU name of the focal point. It is a CP or LU name.

Focal Point NAU Name (X'11') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Focal Point NAU Name subfield
1		Key: X'11'
2−q		A 1-to-8 character NAU name (encoded using Coded Graphic Character Set 01134−00500 documented in Appendix A, "SNA Character Sets and Symbol-String Types"). Trailing space characters may be present, but are insignificant. Leading or embedded space characters may not be present. This subfield may be empty to indicate there is no current focal point. An empty subfield is either one with a length of 2 (consisting of just the Length and Key fields), or one that contains a NAU Name field with no significant characters (all space characters).

Focal Point Application ID (X'12') Focal Point Identification Subfield

This subfield specifies the application program name of the focal point.

Focal Point Application ID (X'12') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Focal Point Application ID subfield
1		Key: X'12'
2−q		Either a 1-to-8 character installation-defined application program name, defined using Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," or one of the 4-byte architecturally-defined values for management services application programs, listed in *SNA/Management Services Reference.* Trailing space characters may be present, but are insignificant. Leading or embedded space characters may not be present. This subfield may be empty to indicate there is no current focal point. An empty subfield is either one with a length of 2 (consisting of just the Length and Key fields), or one that contains a NAU Name field with no significant characters (all space characters).

Backup Focal Point NETID (X'20') Focal Point Identification Subfield

This subfield contains the NETID portion of the network-qualified name of the backup focal point.

Backup Focal Point NETID (X'20') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Backup Focal Point NETID subfield
1		Key: X'20'
2–q		A 1-to-8 character NETID (encoded using Coded Graphic Character Set 01134–00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types"). Trailing space characters may be present, but are insignificant. Leading or embedded space characters may not be present. This subfield may be empty to indicate there is no backup focal point. An empty subfield is either one with a length of 2 (consisting of just the Length and Key fields), or one that contains a NAU Name field with no significant characters (all space characters).

Backup Focal Point NAU Name (X'21') Focal Point Identification Subfield

This subfield contains the unqualified name of the backup focal point.

Backup Focal Point NAU Name (X'21') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Backup Focal Point NAU Name subfield
1		Key: X'21'
2–q		A 1-to-8 character NAU name (encoded using Coded Graphic Character Set 01134–00500 documented in Appendix A, "SNA Character Sets and Symbol-String Types"). Trailing space characters may be present, but are insignificant. Leading or embedded space characters may not be present. This subfield may be empty to indicate there is no backup focal point. An empty subfield is either one with a length of 2 (consisting of just the Length and Key fields), or one that contains a NAU name field with no significant characters (all space characters).

Backup Focal Point Application ID (X'22') Focal Point Identification Subfield

This subfield specifies the application program name of the backup focal point.

Backup Focal Point Application ID (X'22') Focal Point Identification Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Backup Focal Point Application ID subfield
1		Key: X'21'
2−q		Either a 1-to-8 character installation-defined application program name, defined using Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," or one of the 4-byte architecturally-defined values for management services application programs, listed in *SNA/Management Services Reference*. Trailing space characters may be present, but are insignificant. Leading or embedded space characters may not be present. This subfield may be empty to indicate there is no backup focal point. An empty subfield is either one with a length of 2 (consisting of just the Length and Key fields), or one that contains a NAU Name field with no significant characters (all space characters).

Self-Defining Text Message (X'31') MS Common Subvector

This MS common subvector transports a text message, additional data identifying the nature of the message sender, the language of the message, and how the message is encoded.

Self-Defining Text Message (X'31') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Self-Defining Text Message subvector
1		Key: X'31'
2–p		Subfields containing a text message, as well as additional information characterizing the message. *Note:* The following subfield keys are used as indicated:

Subfield	Presence in Self-Defining Text Message (X'31') Common Subvector	
Coded Character Set ID (X'02')	P	
National Language ID (X'12')	CP	Note 1
Sender ID (X'21')	CP	Note 1
Text Message (X'30')	P	

Key:

P Present one time
CP Conditionally present one time (See Notes for conditions.)

Notes:

1. This subfield is present in an Alert.

Coded Character Set ID (X'02') Self-Defining Text Message Subfield

This subfield identifies the coded character set in which the text message is encoded.

Coded Character Set ID (X'02') Self-Defining Text Message Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Coded Character Set ID subfield
1		Key: X'02'

Coded Character Set ID (X'02') Self-Defining Text Message Subfield

Byte	Bit	Content
2−5(=q)		Coded character set ID: two 4-digit hexadecimal numbers that specify uniquely the coded character set in which the accompanying user text message is encoded. Bytes 2−3 contain a 4-digit hexadecimal number identifying a character set, while bytes 4−5 contain a 4-digit hexadecimal number identifying a code page. Receivers are responsible for documenting the coded character set IDs, as well as the coded character sets themselves, that they support in this subvector.

National Language ID (X'12') Self-Defining Text Message Subfield

This subfield identifies the coded national language in which the text message is written.

National Language ID (X'12') Self-Defining Text Message Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the National Language ID subfield
1		Key: X'12'
2−4(=q)		National Language Code: a code point indicating the national language in which the text message is written. A national language is identified by three upper-case alpha EBCDIC characters from Coded Graphic Character Set 01134−00500. The three character IDs are defined in Volume 2 of the *National Language Information and Design Guide*, SE09−8002. For example, American English would be identified in this field as X'C5D5E4', which is decoded as ENU. Other examples are: DEU for German, FRC for Canadian French and ENG for UK English. Receivers are responsible for documenting the national language IDs that they support in this subvector.

Sender ID (X'21') Self-Defining Text Message Subfield

This subfield identifies, in generic terms, the nature of the entity that sent the text message. This information will be displayed by the receiver of the message.

Sender ID (X'21') Self-Defining Text Message Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Sender ID subfield
1		Key: X'21'

Sender ID (X'21') Self-Defining Text Message Subfield

Byte	Bit	Content
2(=q)		Sender ID code: a code point characterizing the sender of the text message. Defined codes are: X'01' terminal user: A person who, when entering the message, is solely a consumer of system resources, i.e., plays no role in providing them X'02' operator: A person who, when entering the message, is in some way involved in providing or managing system resources X'11' application program: A program written for or by an end user that applies to the end user's work *Note:* This program may be implemented in either software or microcode. X'12' control program: A program that controls other system resources. *Note:* This program may be implemented in either software or microcode

Text Message (X'30') Self-Defining Text Message Subfield

This subfield transports a text message.

Text Message (X'30') Self-Defining Text Message Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Text Message subfield
1		Key: X'30'
2−q		Text message

Default Character Set ID (X'32') MS Common Subvector

The Default Character Set ID subvector establishes a character set and code page, and optionally a national language ID, that is to be used to interpret character fields throughout an MS structure if that structure has none of this information specified for it otherwise.

Default Character Set ID (X'32') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Default Character Set ID subvector
1		Key: X'32'
2–p		One or more subfields (listed by Key value below and described in detail following): X'02' Coded Character Set ID X'12' National Language ID *Note:* The following subfield keys may be used as indicated:

Subfield	Presence in Default Character Set ID (X'32') MS Common Subvector	
Coded Character Set ID (X'02')	P	
National Language ID (X'12')	O	

Key:

P Present one time
O Optionally present one time

Coded Character Set ID (X'02') Default Character Set ID Subfield

This subfield is defined in the description of the Self-Defining Text Message (X'31') MS Common Subvector.

National Language ID (X'12') Default Character Set ID Subfield

This subfield is defined in the description of the Self-Defining Text Message (X'31') MS Common Subvector.

Relative Time (X'42') MS Common Subvector

This MS common subvector indicates when a record was created relative to other records created by the originating component.

Relative Time (X'42') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Relative Time subvector
1		Key: X'42'
2		Time units: X'00' tenths of a second X'01'–X'7F' a number that, when divided into the timer data (in bytes 3–6), converts the value to seconds X'90' microseconds X'A0' milliseconds X'C0' minutes (not used in Alerts) X'D0' hours (not used in Alerts) X'EF' indicates time value is purely a sequence indicator showing relative order only
3–6(=p)		Time, in binary, in the units defined by byte 2

NMVT Count (X'44') MS Common Subvector

The NMVT Count subvector is an MS common subvector that is sent in the first NMVT of a single reply to an NMVT request, when the reply data spans multiple NMVTs. The subfields in this subvector are used to determine the size of the entire reply.

NMVT Count (X'44') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the NMVT Count subvector
1		Key: X'44'
2 – p		The following subfields: X'01' Number of Replies X'10' Number of NMVTs X'11' Maximum Size of NMVTs

Number of Replies (X'01') NMVT Count Subfield

This subfield specifies the number of replies that pertain to the soliciting request. This subfield is always present.

Number of Replies (X'01') NMVT Count Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Number of Replies subfield
1		Key: X'01'
2 – 3(=q)		Reply Count: the number of replies, in binary, that pertain to the soliciting request. This field always has a value of X'0001'.

Number of NMVTs (X'10') NMVT Count Subfield

This subfield specifies the number of NMVTs that comprise the reply. This subfield is always present.

Number of NMVTs (X'10') NMVT Count Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Number of NMVTs subfield
1		Key: X'10'
2 – 5(= q)		NMVT count: the number, in binary, of NMVTs that comprise the reply

Maximum Size of NMVTs (X'11') NMVT Count Subfield

This subfield specifies the maximum NMVT size of each of the NMVTs in the reply. This subfield is always present.

Maximum Size of NMVTs (X'11') NMVT Count Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Maximum Size of NMVTs subfield
1		Key: X'11'
2 – 5(= q)		Maximum NMVT size: the maximum size, in binary, of each NMVT in the reply.

Data Reset Flag (X'45') MS Common Subvector

This MS common subvector acknowledges that the reset function has been performed.

Data Reset Flag (X'45') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Data Reset Flag subvector
1(=p)		Key: X'45'

MSU Correlation (X'47') MS Common Subvector

This MS common subvector transports one or more tokens to be used by a receiver for correlating Management Services Units from different senders pertaining to the same condition. Each correlation token is an *entire* subfield, including the length and key.

MSU Correlation (X'47') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the MSU Correlation subvector
1		Key: X'47'
2–p		Subfields containing one or more correlation tokens *Note:* The following subfield keys are used as indicated:

Subfield	Presence in MSU Correlation (X'47') Common Subvector			
	Session Errors	Link Errors	Shared Hardware Resource Errors	Notes
Link Instance Identifier (X'01')	—	CP	—	Note 1
Resource Instance Identifier (X'10')	—	—	CP	Note 2
Fully Qualified PCID (X'60')	P	—	—	Note 3

Key:

—	Not present
P	Present one time
CP	Conditionally present one time

Notes:

1. This subfield is present in an MSU reporting a link or link station error if XID 3 exchange on the link has been completed.
2. This subfield is present when the MSU reports a condition requiring correlation of data sent by multiple senders concerning a hardware resource they share.
3. This subfield is present when the MSU reports an error related to a session.

Link Instance Identifier (X'01') MSU Correlation Subfield

This subfield is present in an MSU reporting a problem for a link or link station after an XID format 3 exchange for that link has taken place. Of the names of the control points at the two ends of the link, it contains the one that occurs earlier in the alphabet. It also contains a unique 4-byte value assigned to the link instance by that control point. With this data, MSUs sent from both sides of a link can be correlated, because they have identical Link Instance Identifier subfields.

Link Instance Identifier (X'01') MSU Correlation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Link Instance Identifier subfield
1		Key: X'01'
2–5		Link instance index: A 4-byte value assigned by the control point named in bytes 6–q. This value is unique, through time, across all of the link instances between this control point and any other node
6–q		Network-qualified control point name *Note:* The format of this name is identical to that in the Network Name (X'0E') control vector, with the restriction that the NETID must be present.

Resource Instance Identifier (X'10') MSU Correlation Subfield

This subfield is present in an MSU reporting a condition present in shared hardware resources. It contains the failing resource name and number, which, together with a sequence number, uniquely identifies a single instance of failure that is reported by multiple senders.

Resource Instance Identifier (X'10') MSU Correlation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Resource Instance Identifier subfield
1		Key: X'10'
2–33		Resource name: The name of the resource experiencing the failure. The name is to be decoded using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol- String Types".
34–37		Resource number: A number that uniquely identifies a single piece of hardware contained in the resource named in byte offset 2-33.

Resource Instance Identifier (X'10') MSU Correlation Subfield

Byte	Bit	Content
38−q		Sequence number: A number that is incremented each time a new reportable condition occurs in the resource identified in byte offset 34-37. Each resource has a sequence number associated with it. This sequence number may or may not have the same value as other resources' sequence numbers at any given time.

Fully-qualified Session PCID (X'60') MSU Correlation Subfield

This subfield specifies the fully-qualified procedure correlation identifier used to uniquely identify a session.

Fully-qualified Session PCID (X'60') MSU Correlation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Fully-qualified Session PCID subfield
1		Key: X'60'
2−9		PCID
10		Length, in binary, of network-qualified CP name (values 3 to 17 are valid)
11−q		Network-qualified CP name *Note:* The format of this name is identical to that in the Network Name (X'0E') control vector, with the restriction that the NETID must be present.

Supporting Data Correlation (X'48') MS Common Subvector

This MS common subvector transports one or more tokens to be used by a receiver for retrieval of additional data related to the event reported by the Management Services Unit containing this subvector.

Supporting Data Correlation (X'48') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Supporting Data Correlation subvector
1		Key: X'48'
2–p		One or more subfields, each containing one correlation token *Note:* The following subfield keys are used as indicated:

Subfield	Presence in Supporting Data Correlator (X'48') Common Subvector	
Fully-Qualified Session PCID (X'60')	O	Note 1
Detailed Data (X'82')	O(n)	Notes 2, 3
Detailed Data Extended (X'85')	O(n)	Notes 2, 3

Key:

O Optionally present one time
O(n) Optionally present one or more times

Notes:

1. This subfield is present to indicate that the Alert sender has stored supporting data that can be accessed by use of the fully-qualified PCID present in the subfield. The techniques needed to access or retrieve supporting data by means of the identifier contained in this subfield are not defined by the architecture.
2. This subfield is present to indicate that the Alert sender has stored supporting data that can be accessed by use of the file or record identifier present in the subfield. The techniques needed to access or retrieve supporting data by means of the identifier contained in this subfield are not defined by the architecture.
3. The X'82' and X'85' subfields can *not* be both used in the same subvector.

Fully-Qualified Session PCID (X'60') Supporting Data Correlation Subfield

This subfield specifies the fully-qualified procedure correlation identifier used to uniquely identify a session.

Fully-qualified Session PCID (X'60') Supporting Data Correlation Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Fully-Qualified Session PCID subfield
1		Key: X'60'
2–9		PCID
10		Length, in binary, of the network-qualified CP name (values 3 to 17 are valid)
11–q		Network-qualified CP name: a 3-to-17 byte name consisting of a mandatory 1-to-8 byte type 1134 symbol-string network identifier (NETID) concatenated with a period to a 1-to-8 byte type 1134 symbol-string name

Detailed Data (X'82') Supporting Data Correlation Subfield

This subfield identifies either a file containing supporting data, or one or more records within such a file. In both cases the identifications are meaningful to the Alert sender. The techniques needed to access or retrieve supporting data by means of the identifier contained in this subfield are not defined by the architecture.

Note: The format of this subfield is defined under the Alert (X'0000') major vector, in the section entitled "Network Alert (X'0000') Common Subfields" on page 9-131.

Detailed Data Extended (X'85') Supporting Data Correlation Subfield

This subfield identifies either a file containing supporting data, or one or more records within such a file. In both cases the identifications are meaningful to the Alert sender. The techniques needed to access or retrieve supporting data by means of the identifier contained in this subfield are not defined by the architecture.

Note: The format of this subfield is defined under the Alert (X'0000') major vector, in the section entitled "Network Alert (X'0000') Common Subfields" on page 9-131.

Incident Identification (X'4A') MS Common Subvector

This MS subvector transports one or more tokens used to identify the incidents (Alert conditions) to which the containing major vector is related. Multiple major vectors containing the same incident identifier token can be considered to deal with the same incident.

Incident Identification (X'4A') Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Incident Identification subvector
1		Key: X'4A'
2−p		Subfields containing identification tokens The following subfield keys are used as indicated:

Subfield	Presence in the Incident Identification (X'4A') Subvector		
	Alert	Resolution	Notes
Incident Identifier (X'01')	P	P(n)	Note 1

Key:

P	Present one time
P(n)	Present one or more times

Notes:

1. This subfield is present once when the X'4A' subvector is in an Alert major vector to uniquely identify the Alert condition (incident) being reported.

   This subfield is present one or more times when the X'4A' subvector is in a Resolution major vector to identify the incidents (Alert conditions) which were corrected. All of the incidents were corrected by fixing the one problem the Resolution major vector is reporting as resolved. Multiple incidents are corrected by resolving a single problem when the problem originally appeared to be several problems, or the same problem occurred at different points in time, thus causing multiple Alerts to be sent. If multiple subfields are present, all other information contained within the Resolution major vector can be assumed to apply to all identified incidents.

Incident Identifier (X'01') Incident Identification Subfield

This subfield is present in an MSU that reports information concerning an Alert condition or incident. All MSUs related to the same incident are to contain identical incident identifier subfields. A receiver can make no assumptions as to the structure of information contained within this subfield.

Incident Identifier (X'01') Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Incident Identifier subfield
1		Key: X'01'
2		Encoding Scheme: A single-byte value denoting the encoding format of the remainder of this subfield.
Encoding Scheme = X'01'		
		Note: Character strings in this encoding scheme are decoded using Coded Graphic Character Set 01134-00500 (documented in Appendix A, "SNA Character Sets and Symbol-String Types"), and padded with trailing blanks (X'40'). Leading and embedded blanks are not allowed.
3 – 10		Network ID: Character string containing the ID of the network in which the application generating the identifier is located.
11 – 18		Network Addressable Unit: Character string identifying the network addressable unit on which the application generating the identifier is located.
19 – 26		Application Name: Character string containing the name of the application generating the identifier.
27 – q		Uniqueness Token: variable-length (1 to 16 bytes, in binary) uniqueness key that is guaranteed to be unique by the application identified in bytes 3 through 26. GMT time may be used in this field, as may a sequence number that is remembered across IPLs of the machine that generates the identifier.
Encoding Scheme = X'02' (to be used only by IBM products)		
		Note: Characters in this encoding scheme are decoded using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."
3 – 6		Machine type of the failing resource: four numeric characters
7 – 9		Model number of the failing resource: three characters
10 – 11		Plant of manufacture of the failing resource: two characters
12 – 18		sequence number of the failing resource: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left
19 – q		Uniqueness Token: variable-length (1 to 16 bytes, in binary) uniqueness key that is guaranteed to be unique by the application that is reporting the incident on behalf of the failing resource. GMT time may be used in this field, as may a sequence number that is remembered across IPLs of the machine that generates the identifier.
Encoding Scheme = X'03' (to be used only by IBM products)		
		Note: Characters in this encoding scheme are decoded using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."
3 – 6		Machine type of the failing resource: four numeric characters

Incident Identifier (X'01') Subfield

Byte	Bit	Content
7–8		Plant of manufacture of the failing resource: two characters
9–15		sequence number of the failing resource: seven characters, right-justified, with EBCDIC 0's (X'F0') fill on the left
16–q		Uniqueness Token: variable-length (1 to 16 bytes, in binary) uniqueness key that is guaranteed to be unique by the application that is reporting the incident on behalf of the failing resource. GMT time may be used in this field, as may a sequence number that is remembered across IPLs of the machine that generates the identifier.
Encoding Scheme = X'04'		
		Note: Characters in this encoding scheme are decoded using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types."
3–8		Machine type of the failing resource: six characters
9–11		Model number of the failing resource: three characters
12–14		Manufacturer: three characters identifying the manufacturer of the failing resource
15–16		Plant of manufacture of the failing resource: two characters
17–28		sequence number of the failing resource: twelve characters
29–q		Uniqueness Token: variable-length (1 to 16 bytes, in binary) uniqueness key that is guaranteed to be unique by the application that is reporting the incident on behalf of the failing resource. GMT time may be used in this field, as may a sequence number that is remembered across IPLs of the machine that generates the identifier.
Encoding Scheme = X'05'		
		Note: Character strings in this encoding scheme are decoded using Coded Graphic Character Set 01134-00500 (documented in Appendix A, "SNA Character Sets and Symbol-String Types"), and padded with trailing blanks (X'40'). Leading and embedded blanks are not allowed.
3–10		Network ID: Character string containing the ID of the network in which the application generating the identifier is located.
11–18		Network Addressable Unit: Character string identifying the network addressable unit on which the application generating the identifier is located.
19–26		Application Name: Character string containing the name of the application generating the identifier.
27–36		Uniqueness Token: fixed-length (10 bytes, in binary) uniqueness key that is guaranteed to be unique by the application identified in bytes 3 through 26. It is necessary that this key be unique even across IPLs of the machine or application that generates the identifier.

LAN Link Connection Subsystem Data (X'51') MS Common Subvector

This MS common subvector transports data on the elements of the LAN link connection.

LAN Link Connection Subsystem Data (X'51') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the LAN Link Connection Subsystem Data subvector
1		Key: X'51'
2–p		One or more subfields containing data specific to the link connection elements (listed by Key value below and described in detail following): X'02' Ring or Bus Identifier X'03' Local Individual MAC Address X'04' Remote Individual MAC Address X'05' LAN Routing Information X'06' Ring Fault Domain Description X'07' Beaconing Data X'08' Single MAC Address X'09' Fault Domain Error Weight Pair X'0A' Bridge Identifier X'23' Local Individual MAC Name X'24' Remote Individual MAC Name X'26' Fault Domain Names X'28' Single MAC Name

Ring or Bus Identifier (X'02') LLC Subsystem Data Subfield

This subfield transports the ring number (for a token-ring LAN) or the bus number (for a CSMA or token-bus LAN).

Ring or Bus Identifier (X'02') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the ring or bus identifier subfield
1		Key: X'02'
2–3(=q)		Ring or bus number, in hexadecimal

Local Individual MAC Address (X'03') LLC Subsystem Data Subfield

This subfield transports the address of the MAC within the node sending the MS major vector.

Local Individual MAC Address (X'03') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the local individual MAC address subfield
1		Key: X'03'
2–7(=q)		Local individual MAC address, in hexadecimal

Remote Individual MAC Address (X'04') LLC Subsystem Data Subfield

This subfield transports the address of the MAC, part of the link connection, within the adjacent node.

Remote Individual MAC Address (X'04') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the remote individual MAC address subfield
1		Key: X'04'
2–7(=q)		Remote individual MAC address, in hexadecimal

LAN Routing Information (X'05') LLC Subsystem Data Subfield

This subfield transports the routing information used by a link.

LAN Routing Information (X'05') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the LAN routing information subfield
1		Key: X'05'
2–q		Routing information, not to exceed 18 bytes, in hexadecimal. For details, see the Routing Information field in *IBM Token-Ring Network Architecture Reference*, SC30-3374.

Fault Domain Description (X'06') LLC Subsystem Data Subfield

This subfield identifies a pair of LAN token-ring stations as a fault domain, i.e., the upstream and the downstream LAN token-ring stations and the cable between them.

Fault Domain Description (X'06') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Ring Fault Domain Description subfield
1		Key: X'06'
2–7		Individual MAC address of downstream station, in hexadecimal
8–13(=q)		Individual MAC address of upstream station, in hexadecimal

Beaconing Data (X'07') LLC Subsystem Data Subfield

This subfield specifies the type of beacon detected by the LAN adapter.

Beaconing Data (X'07') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Beaconing Data subfield
1		Key: X'07'
2(=q)		Beaconing type: X'01' type 1, recovery mode set X'02' type 2, signal loss X'03' type 3, streaming signal

Single MAC Address (X'08') LLC Subsystem Data Subfield

This subfield transports the address of the MAC element associated with the failure.

Single MAC Address (X'08') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Single MAC Address subfield
1		Key: X'08'
2 – 7(= q)		Single MAC address, in hexadecimal

Fault Domain Error Weight Pair (X'09') LLC Subsystem Data Subfield

This subfield indicates the severity of the problems reported by two MAC elements (LAN stations) belonging to a fault domain.

Fault Domain Error Weight Pair (X'09') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Fault Domain Error Weight Pair subfield
1		Key: X'09'
2 – 3		Severity weight, in binary, for the downstream MAC element (LAN station) problems
4 – 5(= q)		Severity weight, in binary, for the upstream MAC element (LAN station) problems

Bridge Identifier (X'0A') LLC Subsystem Data Subfield

This subfield transports the bridge identifier of a LAN bridge.

Bridge Identifier (X'0A') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Bridge Identifier subfield
1		Key: X'0A'

Bridge Identifier (X'0A') LLC Subsystem Data Subfield

Byte	Bit	Content
2–5		Bridge identifier, composed of three hexadecimal parts: a ring or bus number, followed by a bridge number, followed by another ring or bus number. The ring or bus with the lower number is always identified first. The bridge identifier occupies less than 4 bytes, the amount less depending on the partitioning of the LAN routing information field. The bridge identifier is left-justified, with the remaining portion of the subfield being 0's. **Note:** The partitioning of this field into its three parts is not specified, but is necessarily unique within a LAN.

Local Individual MAC Name (X'23') LLC Subsystem Data Subfield

This subfield transports the name of the MAC element within the sending node.

Local Individual MAC Name (X'23') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Local Individual MAC Name subfield
1		Key: X'23'
2–q		Local individual MAC name: is defined using Coded Graphic Character Set 01134-00500 documented in Appendix A, "SNA Character Sets and Symbol-String Types" and four additional code points: X'5B' = $ (dollar sign) X'6C' = % (percent sign) X'7B' = # (pound or number sign) X'7C' = @ (at sign)

Remote Individual MAC Name (X'24') LLC Subsystem Data Subfield

This subfield transports the name of the MAC element, part of the link connection, within the adjacent node.

Remote Individual MAC Name (X'24') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Remote Individual MAC Name subfield
1		Key: X'24'

Remote Individual MAC Name (X'24') LLC Subsystem Data Subfield

Byte	Bit	Content
2−q		Remote individual MAC name: is defined using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" and four additional code points: X'5B' = $ (dollar sign) X'6C' = % (percent sign) X'7B' = # (pound or number sign) X'7C' = @ (at sign)

Fault Domain Names (X'26') LLC Subsystem Data Subfield

This subfield transports the names of the upstream and the downstream LAN ring stations belonging to a fault domain.

Fault Domain Names (X'26') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Ring Fault Domain Names subfield
1		Key: X'26'
2−q		Pair of Entries **Note:** Each entry contains a Length field and a Name field; the first entry is for the downstream MAC element, and the second entry is for the upstream MAC element. Each entry has the following form (shown 0-origin).
0		Length (r + 1), in binary, of the following name plus this length field
1−r		Individual MAC name: is defined using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" and four additional code points: X'5B' = $ (dollar sign) X'6C' = % (percent sign) X'7B' = # (pound or number sign) X'7C' = @ (at sign)

Single MAC Name (X'28') LLC Subsystem Data Subfield

This subfield transports the name of the MAC related to the failure.

Single MAC Name (X'28') LLC Subsystem Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Single MAC Name subfield
1		Key: X'28'
2−q		Single MAC name: is defined using Coded Graphic Character Set 01134-00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types" and four additional code points: X'5B' = $ (dollar sign) X'6C' = % (percent sign) X'7B' = # (pound or number sign) X'7C' = @ (at sign)

Link Connection Subsystem Configuration Data (X'52') MS Common Subvector

This MS common subvector transports data for link connections.

Link Connection Subsystem Configuration Data (X'52') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the LCS Configuration Data subvector
1		Key: X'52'
2–p		One or more subfields containing LCS configuration data (listed by key value below and described in detail following): X'01' Port Address X'02' Remote Device Address X'04' Local Device Address X'06' LCS Link Station Attributes X'07' LCS Link Attributes X'08' LPDA Fault LSL Descriptor X'09' Remote Telephone Number X'0A' Local Telephone Number X'0B' Adapter Number X'0C' Channel Number

Port Address (X'01') Link Connection Subsystem Config. Data Subfield

This subfield transports the port address of the link connection.

Port Address (X'01') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Port Address subfield
1		Key: X'01'
2–3(=q)		Port address, in hexadecimal

Remote Device Address (X'02') Link Connection Subsystem Config. Data Subfield

This subfield transports the DLC address of the remote link station.

Remote Device Address (X'02') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Remote Device Address subfield
1		Key: X'02'
2(= q)		Remote link station DLC address, in hexadecimal; e.g., for a LAN, the destination link service access point (DSAP) address

Local Device Address (X'04') Link Connection Subsystem Config. Data Subfield

This subfield transports the address of the local link station.

Local Device Address (X'04') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Local Device Address subfield
1		Key: X'04'
2 − 3(= q)		Local link station DLC address, in hexadecimal; e.g., for a LAN, the source service access point (SSAP) address or for ISDN, the data link control identifier (DLCI) for LAPE and the service access point identifier (SAPI) and terminal equipment identifier (TEI) for LAPD

LCS Link Station Attributes (X'06') Link Connection Subsystem Config. Data Subfield

This subfield identifies link station attributes.

LCS Link Station Attributes (X'06') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the LCS Link Station Attributes subfield
1		Key: X'06'
2		Link station role: X'01' primary X'02' secondary X'03' negotiable

LCS Link Station Attributes (X'06') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
3(=q)		Node type for the remote link station: X'01' type 1 X'02' type 2.0 X'03' type 4 X'04' type 2.1 X'80' non-SNA, e.g., used for BSC links

LCS Link Attributes (X'07') Link Connection Subsystem Config. Data Subfield

This subfield transports LCS link connection attributes.

LCS Link Attributes (X'07') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the LCS Link Attributes subfield
1		Key: X'07'
2		Link connection type used: X'01' nonswitched X'02' switched X'03' packet switched
3		Half- or full-duplex: X'01' half-duplex X'02' full-duplex
4		DLC protocol type: X'01' SDLC X'02' BSC X'03' start-stop X'04' LAPB X'05' LAPD X'06' LAPE X'07' LAN LLC
5(=q)		Point-to-point or multipoint: X'01' point-to-point X'02' multipoint

LPDA Fault LSL Descriptor Subfield (X'08') Link Connect. Subsys. Config. Data Subfield

This subfield transports the link segment identifier, also referred to as level, of the multi-segment LPDA link where the failure occurred.

LPDA Fault LSL Descriptor Subfield (X'08') Link Connect. Subsys. Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the LPDA Fault LSL Descriptor subfield
1		Key: X'08'
2		LPDA fault link segment level (LSL) descriptor value, in binary

Remote Telephone Number (X'09') Link Connection Subsystem Config. Data Subfield

This subfield transports the telephone number of the remote node.

Remote Telephone Number (X'09') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Remote Telephone Number subfield
1		Key: X'09'
2		Telephone number encoding used: X'00' IA5-EBCDIC (the data is to be decoded using Coded Graphic Character Set 001169 – 00500) X'01' Binary Coded Decimal (i.e. an unsigned number with one hex digit for each decimal digit padded with one X'F' after the last digit in cases of odd number of digits)
3 – q		Remote telephone number

Local Telephone Number (X'0A') Link Connection Subsystem Config. Data Subfield

This subfield transports the telephone number of the local node.

Local Telephone Number (X'0A') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Local Telephone Number subfield

Local Telephone Number (X'0A') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
1		Key: X'0A'
2		Telephone number encoding used: X'00' IA5-EBCDIC (the data is to be decoded using Coded Graphic Character Set 001169−00500) X'01' Binary Coded Decimal (i.e. an unsigned number with one hex digit for each decimal digit padded with one X'F' after the last digit in cases of odd number of digits)
3−q		Local telephone number

Adapter Number (X'0B') Link Connection Subsystem Config. Data Subfield

This subfield transports the adapter number of the adapter used for the link connection.

Adapter Number (X'0B') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Adapter Number subfield
1		Key: X'0B'
2−3(=q)		Adapter number, in hexadecimal

Channel Number (X'0C') Link Connection Subsystem Config. Data Subfield

This subfield transports the channel number of the B channel used by the link connection.

Channel Number (X'0C') Link Connection Subsystem Config. Data Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Channel Number subfield
1		Key: X'0C'
2−3(=q)		Channel number, in hexadecimal

Focal Point Authorization Request (X'61') MS Common Subvector

The Focal Point Authorization Request subvector carries a request from a focal point to an entry point in its sphere of control, to receive unsolicited management service units (MSUs) for the specified MS category, from that entry point and its domain.

Focal Point Authorization Request (X'61') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Focal Point Authorization Request subvector
1		Key: X'61'
2 – p		One of the following subfields: X'10' Primary Authorization Request X'20' Secondary Authorization Request

Primary Authorization Request (X'10') Focal Point Authorization Request Subfield

This subfield identifies this request as being from a primary focal point. It specifies the value X'0000' or X'FFFF' depending on the value in the MS Category (X'01') subfield of the Focal Point Identification (X'21') subvector.

Primary Authorization Request (X'10') Focal Point Authorization Request Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Primary Authorization Request subfield
1		Key: X'10'
2 – 3		A value, for migration, loosely coupled with the MS category code in the (X'01') subfield of the Focal Point Identification (X'21') subvector, as defined below: X'0000' if MS category = X'23F0F3F1' X'FFFF' if MS category ≠ X'23F0F3F1'
4(= q)	0	MS Capabilities recovery responsibility: set by the focal point to indicate which party (focal point or entry point) is responsible for recovery of the focal-point/entry-point relationship after a failure (such as loss of connectivity). 0 Focal point will perform recovery by resending MS Capabilities with the X'61' subvector after a failure.

Primary Authorization Request (X'10') Focal Point Authorization Request Subfield

Byte	Bit	Content
	1	The explicit primary focal point name, found in the Focal Point Identification (X'21') MS common subvector that accompanies this subvector in the MS Capabilities (X'80F0') MS major vector, will be saved by the entry point as its new implicit primary focal point. After a failure, the entry point will perform recovery by sending MS Capabilities with the X'63' subvector, thus establishing an implicit primary focal point relationship with the focal point that was previously its explicit primary. Refer to *SNA Management Services Reference* for specific information about focal point recovery.
	1–7	Reserved.

Default Authorization Request (X'20') Focal Point Authorization Request Subfield

This subfield identifies this request as being from a default focal point. It specifies the value X'0000' or X'FFFF', depending on the value in the MS category (X'01') subfield of the Focal Point Identification (X'21') subvector.

Default Authorization Request (X'20') Focal Point Authorization Request Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Default Authorization Request subfield
1		Key: X'20'
2–3		A value, for migration, loosely coupled with the MS category code in the (X'01') subfield of the Focal Point Identification (X'21') subvector, as defined below: X'0000' if MS category = X'23F0F3F1' X'FFFF' if MS category ≠ X'23F0F3F1'
4(=q)		Reserved

Focal Point Authorization Reply (X'62') MS Common Subvector

The Focal Point Authorization Reply subvector flows from an entry point to its focal point to indicate the status of the focal-point/entry-point relationship for the specified MS Category.

Focal Point Authorization Reply (X'62') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Focal Point Authorization Reply subvector
1		Key: X'62'
2 – p		The following subfields as indicated: X'10' Authorization Request Accepted X'20' Authorization Request Rejected X'30' Authorization Revoked X'40' Current Focal Point NAU Name X'41' Current Focal Point Application Name

Authorization Request Accepted (X'10') Focal Point Authorization Reply Subfield

This subfield indicates that the authorization request has been accepted. It specifies the value X'0000' or X'FFFF' depending on the value in the MS Category (X'01') subfield of the Focal Point Identification (X'21') subvector.

Authorization Request Accepted (X'10') Focal Point Authorization Reply Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Authorization Request Accepted subfield
1		Key: X'10'
2 – 3		A value, for migration, loosely coupled with the MS category code in the (X'01') subfield of the Focal Point Identification (X'21') subvector, as defined below: X'0000' if MS category = X'23F0F3F1' X'FFFF' if MS category ≠ X'23F0F3F1'
4(= q)		Reserved

Authorization Request Rejected (X'20') Focal Point Authorization Data Subfield

This subfield indicates that the request for authorization has been rejected. It specifies the value X'0000' or X'FFFF' depending on the value in the MS Category (X'01') subfield of the Focal Point Identification (X'21') subvector and indicates the reason for the rejection of the request.

Authorization Request Rejected (X'20') Focal Point Authorization Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Authorization Request Rejected subfield
1		Key: X'20'
2–3		A value, for migration, loosely coupled with the MS category code in the (X'01') subfield of the Focal Point Identification (X'21') subvector, as defined below: X'0000' if MS category = X'23F0F3F1' X'FFFF' if MS category ≠ X'23F0F3F1'
4		Reserved
5(= q)		Authorization status code: indicates the reason authorization for the specified category was not granted. One of the following code points is present: X'01' Authorization rejected: function not supported X'02' Authorization rejected: the request was from a focal point of lower rank and priority than the focal point this node currently has. When this code point is used, the Current Focal Point NAU Name (X'40') subfield and Current Focal Point Application Name (X'41') subfield are always also present in the X'62' subvector. The X'40' and X'41' subfields specify the entry point's current focal point for the requested category. X'03' Authorization rejected: the receiving node is an end node and does not support maintaining a focal point for the requested MS category.

Authorization Revoked (X'30') Focal Point Authorization Data Subfield

This subfield indicates that focal-point authorization has been revoked. It specifies the value X'0000' or X'FFFF', depending on the value in the MS Category (X'01') subfield of the Focal Point Identification (X'21') subvector, and the reason for the revocation.

Authorization Revoked (X'30') Focal Point Authorization Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Authorization Revoked subfield
1		Key: X'30'

Authorization Revoked (X'30') Focal Point Authorization Data Subfield

Byte	Bit	Content
2–3		A value, for migration, loosely coupled with the MS category code in the (X'01') subfield of the Focal Point Identification (X'21') subvector, as defined below: X'0000' if MS category = X'23F0F3F1' X'FFFF' if MS category ≠ X'23F0F3F1'
4		Reserved
5(=q)		Authorization status code: indicates the reason authorization for the specified category was revoked. One of the following code points is present: X'60' Authorization revoked: the focal point receiving this subfield was replaced as this node's focal point. When this code point is used, the Current Focal Point NAU Name (X'40') subfield and the Current Focal Point Application Name (X'41') subfield must also be present in the X'62' subvector. The X'40' and X'41' subfields identify the currently active focal point. X'61' Authorization revoked: the focal point receiving this subfield is revoked due to an explicit command from an operator at the entry point. The X'40' and X'41' subfields are not present for this authorization status code.

Current Focal Point NAU Name (X'40') Focal Point Authorization Data Subfield

This subfield specifies the name of the current focal point for the node sending the subfield. The MS category for which the focal point is authorized to receive unsolicited data is identified in either the Authorization Request Rejected (X'20') or Authorization Revoked (X'30') subfield also present in this X'62' subvector.

Current Focal Point NAU Name (X'40') Focal Point Authorization Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Current Focal Point NAU Name subfield
1		Key: X'40'
2		Length of network-qualified NAU name (values 3 to 17 are valid)
3–q		Network-qualified NAU name (NETID always present). The NETID and NAU name are defined using Coded Graphic Character Set 01134–00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," and they are separated by a period (X'4B') Trailing space characters may be present, but are not significant. Leading or embedded space characters may not be present.

Current Focal Point Application Name (X'41') Focal Point Authorization Data Subfield

This subfield specifies the application program name of the current focal point for the node sending the subfield.

Current Focal Point Application Name (X'41') Focal Point Authorization Data Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Current Focal Point Application Name subfield
1		Key: X'41'
2−q		Either an 8-byte installation-defined application name, defined using Coded Graphic Character Set 01134−00500, documented in Appendix A, "SNA Character Sets and Symbol-String Types," or one of the 4-byte architecturally-defined values for management services application programs, listed in *SNA/Management Services Reference*. Trailing space characters may be present, but are not significant. Leading or embedded space characters may not be present.

Entry Point Authorization Request (X'63') MS Common Subvector

The Entry Point Authorization Request subvector carries a request from an entry point to a focal point to request primary focal-point services for the unsolicited management service units (MSUs) associated with a specified MS category.

Entry Point Authorization Request (X'63') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Entry Point Authorization Request subvector
1		Key: X'63'
2−p		One of the following subfields: X'10' Primary Authorization Request X'30' Backup Authorization Request

Primary Authorization Request (X'10') Entry Point Authorization Request Subfield

This subfield specifies that the request is for services of a primary focal point.

Primary Authorization Request (X'10') Entry Point Authorization Request Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Primary Authorization Request subfield
1(=q)		Key: X'10'

Backup Authorization Request (X'30') Entry Point Authorization Request Subfield

This subfield specifies that the request is for services of a backup focal point.

Backup Authorization Request (X'30') Entry Point Authorization Request Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Backup Authorization Request subfield
1(=q)		Key: X'30'

Entry Point Authorization Reply (X'64') MS Common Subvector

The Entry Point Authorization Reply subvector flows from a focal point to an entry point to indicate the authorization status of the entry point for a specified MS category.

Entry Point Authorization Reply (X'64') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Entry Point Authorization Reply subvector
1		Key: X'64'
2−p		One of the following subfields: X'10' Authorization Request Accepted X'20' Authorization Request Rejected

Authorization Request Accepted (X'10') Entry Point Authorization Reply Subfield

This subfield specifies that request for authorization has been accepted.

Authorization Request Accepted (X'10') Entry Point Authorization Reply Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Authorization Request Accepted subfield
1(=q)		Key: X'10'

Authorization Request Rejected (X'20') Entry Point Authorization Reply Subfield

This subfield specifies that the request for authorization has been rejected, and indicates the reason for that rejection.

Authorization Request Rejected (X'20') Entry Point Authorization Reply Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Authorization Request Rejected subfield
1		Key: X'20'

Authorization Request Rejected (X'20') Entry Point Authorization Reply Subfield

Byte	Bit	Content
2(=q)		Authorization status code: indicates the reason authorization for the specified category was not granted. One of the following code points is present: X'01' Authorization rejected: function not supported X'02' Authorization rejected: this node not a focal point for that category X'03' Authorization rejected: not able to support request at this time

Sense Data (X'7D') MS Common Subvector

This MS common subvector transports error information back to the control point that initiated an MS request. The subvector contains a 4-byte field for sense data.

Sense Data (X'7D') MS Common Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Sense Data subvector
1		Key: X'7D'
2 – 5(=p)		Sense data, as defined in Chapter 10, "Sense Data"

Chapter 10. Sense Data

The sense data included with an EXCEPTION REQUEST (EXR), a negative response, an UNBIND request, a Sense Data (X'7D') MS common subvector, a function management header type 7 (FMH-7), an extended sense data control vector (X'35'), or a SNA report code is a 4-byte field (see Figure 10-1) that includes a 1-byte category value, a 1-byte modifier value, and two bytes of sense code specific information, whose format is defined along with the sense code definition, below.

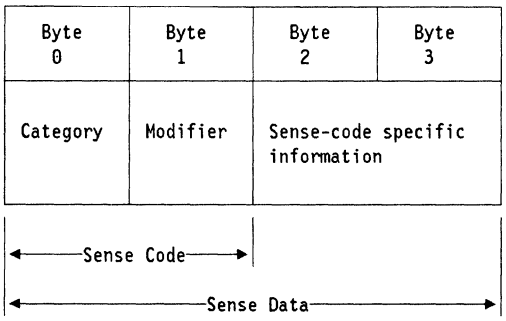


Figure 10-1. Sense Data Format

Together, the category byte 0, the modifier byte 1, and the sense code specific bytes 2 and 3 hold the sense data defined for the exception condition that has occurred.

The following categories are defined; all others are reserved:

VALUE	CATEGORY
X'00'	User Sense Data Only
X'08'	Request Reject
X'10'	Request Error
X'20'	State Error
X'40'	Request Header (RH) Usage Error
X'80'	Path Error

The category User Sense Data Only (X'00') allows the end users to exchange sense data in bytes 2−3 for conditions not defined by SNA within the other categories (and perhaps unique to the end users involved). The modifier value is also X'00'. User Sense Data may not be sent on LU 6.2 sessions.

In earlier versions of SNA, user data (as well as implementation-specific data) generally could be carried in bytes 2−3 for all categories. This is no longer the case. Bytes 2−3 are used generally only for SNA-defined conditions for nonzero categories; exceptions for implementation-specific use are documented in the appropriate product publications.

The sense codes for the other categories are discussed below.

Request Reject (Category Code = X'08')

This category indicates that the request was delivered to the intended component and was understood and supported, but not executed.

Category and modifier (in hexadecimal):

0801 Resource Not Available: The LU, PU, link station, or link specified in an RU is not available.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

- 0000 No specific code applies.
- 0001 Independent LU Does Not Receive ACTLU: An ACTLU has been sent by the SSCP to an independent LU (sent by BF).
- 0003 Name aliasing cannot be performed because the name alias function is not available.
- 0006 The line is not associated with a line adapter, or is associated with a line adapter that is not valid for the usage tier specified in the receiving node's system definition.
- 0007 The line is associated with a line adapter that is not installed or not attached to a communications processor.
- 0008 The line is associated with a line adapter that is inoperative.
- 0009 The LU is not available because it is not ready to accept sessions.
- 000A The PLU is not available because it is being taken down, and is therefore not accepting new sessions. The initiation request should not be retried.
- 000B The PLU is not available because it is unable to comply with the PLU-SLU role specification.
- 000C The SLU is not available because it is unable to comply with the PLU-SLU role specification.
- 000D The LU is not available because its SSCP is in the process of being taken down, and is therefore not allowing new sessions to be started. The initiation request should not be retried.
- 000E The LU is not available because an intermediate gateway SSCP is in the process of being taken down, and is therefore not allowing new sessions to be started.
- 000F The SLU is not available because it is being taken down, and is therefore not accepting new sessions. The initiation request should not be retried.

0010 Switched subarea connection cannot be established because no switched subarea links have been defined.

0011 Switched subarea connection to another network cannot be established because no switched subarea links have been defined within the gateway PU.

0012 An APPN connection cannot be established because this node has no available integers to represent a new TG.

0013 A switched connection cannot be established because no short-hold mode capable link is defined.

0014 A switched connection cannot be established. Call Request Verification was requested, but is not supported for this configuration. This condition will result from conflicting system definition.

001A-001E Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

4001 – 4002 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0802 Intervention Required: Forms or cards are required at an output device, or a device is temporarily in local mode, or other conditions require intervention.

0803 Missing Password: The required password was not supplied.

0804 Invalid Password: Password was not valid.

0805 Session Limit Exceeded: The requested session cannot be activated, as one of the NAUs is at its session limit, for example, the LU-LU session limit or the (LU, mode) session limit. This sense code applies to ACTCDRM, INIT, BIND, and CINIT requests.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 If accepted, the BIND request would prevent either the receiving LU or the sending LU from activating the number of contention winner sessions to the partner LU that were agreed upon during a change-number-of-sessions procedure.

0002 If accepted, the BIND request would cause the XRF-backup session limit to be exceeded.

0003 If accepted, the BIND request would cause the XRF-active session limit to be exceeded.

Note: The session limit for XRF-active sessions is 1. An XRF-active BIND is valid only if there are no XRF-active or XRF-backup sessions with the receiving SLU.

0005 The intermediate session router is unable to create a session connector control block. The pool of session connectors is saturated with active sessions and with pending-active sessions for which the Queue bit was set in the BIND; the BIND should not be retried.

0006 The intermediate session router is unable to create a session connector control block. The pool of session connectors is saturated with active sessions and with pending-active sessions for which the Queue bit was not set in the BIND; the BIND should be retried.

0008 For a dependent LU, if accepted, the BIND request would cause the session limit to be exceeded.

0009 If accepted, the request would cause the PLU session limit to be exceeded.

000A If accepted, the request would cause the SLU session limit to be exceeded.

000B The request was rejected because a session already exists between the same LU pair, and at least one of the LUs does not support parallel sessions.

000C An LU-LU session was not established because a session already exists between the SLU and the session-controller PLU.

0806 Resource Unknown: For example, the request contained a name or address not identifying a PU, LU, SSCP, link, or link station known to the receiver or the sender.

Note: In an interconnected network environment, this sense code may be set by an SSCP in whose subnetwork and domain the LU was expected to reside; it is not set by an SSCP that is only an intermediary on the session-setup path. A gateway SSCP examines the Resource Identifier control vector in a session setup request (for example, CDINIT), to determine whether the LU is in the SSCP's subnetwork and domain.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The resources identified in an SNA Address List (X'04') MS common subvector are unknown to the PU receiving the request.

Note: When this sense data flows in a -RSP(NMVT), the referenced X'04' subvector is the one that was present in the corresponding request NMVT. When this sense data flows in a Sense Data (X'7D') MS common subvector, the referenced X'04' subvector is present with the X'7D' subvector in the same major vector.

0002	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0009	Reserved
000A	The configuration identifier specified in a management services command is not recognized by
000B	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0011	An unknown OLU name was specified in the request.
0012	An unknown DLU name was specified in the request.
0013	An unknown SLU name was specified in the request.
0014	An unknown PLU name was specified in the request.
0015	An unknown OLU address was specified in the request.
0016	An unknown DLU address was specified in the request.
0017	An unknown SLU address was specified in the request.
0018	An unknown PLU address was specified in the request.
0021	The session-initiation request specified that the receiving SSCP is the SSCP having the DLU in its domain, but the DLU is unknown to the receiving SSCP.
0022	The originator of the request is unknown to the receiver.
0023	The destination of the request or response is unknown to the sender.
0024	An unknown LU1 name was specified in the request.
0025	An unknown LU2 name was specified in the request.
0026	The SSCP does not have a session with the boundary function PU of an independent LU.
0027	The PU associated with a switched SLU is unknown. Session setup processing for the switched SLU cannot proceed.
0028	NAU1 network address is unknown.
0029	NAU2 network address is unknown.
002A	The NAU name in the CONTACT or ACTLU does not correspond to the resource at the target address.
002B	The TG being activated is unknown.
002C	The identification supplied by the adjacent node in its XID3 differed from the identification that the receiving node was configured to expect.

002D,002E Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

002F The destination resource was not found on this node during a session activation attempt.

0030 The adjacent node was not identified during CP-CP session activation.

0807 Resource Not Available—LUSTAT Forthcoming: A subsidiary device will be unavailable for an indeterminate period of time. LUSTAT will be sent when the device becomes available.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0808 Invalid Contents ID: The contents ID contained on the ACTCDRM request was found to be invalid.

0809 Mode Inconsistency: The requested function cannot be performed in the present state of the receiver.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 – 000D Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

000F An RNAA received from an SSCP is rejected because it specifies a resource (adjacent link station or LU) that currently has an address assigned as a result of another SSCP's RNAA; or an ACTLU, FNA, or SETCV received from an SSCP is rejected because it specifies a resource address that is not assigned to an existing resource or is assigned as a result of another SSCP's RNAA.

0010 – 0013 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0014 ANS mismatch discovered.

0015 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0016 The PU type on SETCV does not match the actual PU type.

0017,0018	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0019	A SETCV was received containing a value for the SDLC BTU send limit that conflicts with the previous value received.
001A,001B	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
001C	The RNAA request contains a network ID that is not known to the gateway PU.
001D	An address pair session key in a Network-Qualified Address Pair (X'15') control vector is not known to the gateway PU.
001E	A gateway PU received an RNAA request for a cross-network session and all possible address transforms for the named resource are allocated.
001F	Retired
0020	The gateway node receiving an RNAA request cannot support another session between the named resource pair.
0021 – 0023	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0024	A PU received an ACTPU request with the SSCP-PU Session Capabilities control vector (X'0B') indicating that the sending SSCP does not support ENA, but the PU does not know the SSCP's maximum subarea address value.
0025	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0026	A SETCV was received containing an SDLC BTU send limit of 0.
0027	A request for a function was received by a component but the function was not enabled or activated.
0028	Cleanup termination of an LU-LU session has been converted to a forced termination by the LU. The SSCP must wait for session ended signals before deleting its session awareness records of the session.
0031	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0032 A BFSESSINFO was received when the subject LU was not pending BFSESSINFO; the reported sessions will be terminated, and the associated network addresses will be freed. This sense data is also included in the BFCLEANUP when the sessions are terminated.

0033 A BIND with the same LFSID used for an existing pending-reset session has been received by a boundary function from a peripheral PLU.

0034 A termination request has been received for a resource that has been taken over by an SSCP. The termination type is not strong enough to apply to the resources. The termination type needs to be Forced or Cleanup.

0035 An other-domain resource that was expected to be active is inactive.

0036 The short-hold mode logical connection selected does not exist.

0037 A non-short-hold mode connection was attempted on a port that is dedicated to short-hold mode operation.

0038 There is an inconsistency of mode between the XID sender and receiver. The XID receiver is operating in short-hold mode. Examples include inconsistent settings of the Short-Hold indicator (SHI) and the Short-Hold Status indicator (SHSI).

0039 CP Transaction Error: CP Capabilities (X'12C1') GDS variable request sent indicating conversation complete or without change direction (i.e., CEB or ¬ CD) or CP Capabilities reply sent indicating conversation not yet complete (i.e., ¬ CEB).

003A A null XID was received when an XID3 with its Exchange State indicators set to "prenegotiation," "negotiation proceeding,"or "Exchange State indicators not supported" was expected.

003B A null XID was received when a nonactivation XID3 was expected.

003C An XID3 with the Exchange State indicators set to "prenegotiation"was received when either of the values "negotiation proceeding" or "Exchange State indicators not supported" was expected.

003D A nonactivation XID3 was received when a null XID or link-activation XID3 was expected.

003E A link activation XID3 was received when a null XID or nonactivation XID3 was expected.

003F	The node with a secondary link station attempted to initiate a nonactivation exchange when secondary-initiated nonactivation exchanges are not supported on the connection.
0040	A mode-setting command was received and was either not expected or invalid for the receiving node; e.g., SNRME was received when SNRM was expected.
0041	An XID3 with the Exchange State indicators specifying a negotiation-proceeding exchange was received when an XID3 indicating a prenegotiation exchange was expected. If prenegotiation XID3s are used in a link activation XID exchange, each node must send and receive one.
0042	On an ABM TG on which secondary-initiated nonactivation XID exchanges are supported, the adjacent link station has initiated a nonactivation exchange by sending a nonactivation XID3 in which the ABM Nonactivation XID Exchange Initiator indicator specifies that the sending node is not initiating a nonactivation exchange. On such TGs, the initiator of a nonactivation exchange always explicitly indicates that it is initiating a nonactivation exchange.
0046	An XID3 indicating that the sender supports the Exchange State indicators was received when the sender had previously indicated that it does not support this field in XID3.
0047	An XID has been received after receipt of a mode-setting command but before the completion of the mode-setting sequence, i.e., before RR, RNR, or an I-frame with the Poll bit set has been sent by the node with the primary link station after it has received UA in response to its mode-setting command.
0048	A node with an NRM primary link station has received an XID3 when it has no outstanding commands. NRM secondary link stations send XIDs only in response to XID commands.
0049	The XID3 received from the adjacent node had an XID Negotiation Error (X'22') control vector appended. The XID exchange will therefore terminate unsuccessfully.
004A	The request cannot be accepted because DR (dynamic reconfiguration) is in process for the target resource.
004B-004D	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
004E	A node with a secondary NRM link station has attempted to initiate a nonactivation XID exchange with an XID3. Nodes with secondary NRM link stations may solicit a

nonactivation XID3 exchange only by means of sending a null XID at a response opportunity.

0050 An UNBIND request was received on behalf of a resource for which a previous UNBIND is in progress. The second UNBIND does not indicate an override of the first, and is therefore a duplicate request.

0054 The adjacent node is not the node type that the receiving node was configured to expect. The received negotiation-proceeding XID3 indicated that the adjacent node is an end node while this node expected the adjacent node to be a network node, or *vice versa.*

080A Permission Rejected: The receiver has denied an implicit or explicit request of the sender.

When sent in response to BIND, it implies either that the secondary LU will not notify the SSCP when a BIND can be accepted, or that the SSCP does not recognize the NOTIFY vector key X'0C'. (See the X'0845' sense code for a contrasting response.)

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 An SSCP has denied permission to establish a session through its resources; the receiving SSCP should not attempt to reroute the request to another SSCP.

0002 An SSCP has denied permission to establish a session through its resources; the receiving SSCP should attempt to reroute the request to another SSCP.

0005 An SSCP has denied permission to continue session setup. A DLU name was presented to an alias selection function (authorization application) that rejected it. The receiving SSCP should not attempt to reroute the request to another SSCP.

0006 An SSCP has denied permission to continue session setup. A DLU name was presented to an alias selection function (authorization application) that rejected it. The receiving SSCP should attempt to reroute the request to another SSCP.

0007 An activation request was received for a resource that has a NETID different from that of the requesting SSCP, and the requesting SSCP indicated previously that it does not support this configuration.

0008 The request specified in the Request Change Control MS major vector was rejected because it did not originate from a valid focal point.

0009 The request specified in the Request Change Control MS major vector was rejected because the ability to support it has been disabled at the receiver.

000A The request was rejected because it would prohibit compliance with the status-reporting requirements specified in the Reporting Level MS Common subvector.

000B The request was rejected because the second-level application, though recognized, operates under the control of a program other than that which has received and is to forward the request to that second-level application program.

000C The request was rejected because the timer/clock at the receiver is protected and cannot be set by the request sender.

080B Bracket Race Error: Loss of contention within the bracket protocol. This error can arise when bracket initiation/termination by both NAUs is allowed

080C Procedure Not Supported: A procedure (Test, Trace, IPL, REQMS type, MS major vector key) specified in an RU is not supported by the receiver.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 – 0003 Set aside for implementation specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0005 The MS major vector key is not supported by the receiver.

0006 The MS major vector is identified as one that contains a command, but the receiver does not recognize or support the command subvector. (See the X'086C' sense code for the case in which the command subvector is identified, but an additional required subvector is missing.)

0007 A request for a function is supported by the receiver, but the resource identified in the request does not support that function (no function is specifically indicated).

0009 A request for session information retrieval for an independent LU was received in an REQMS; such requests are permitted only in an NMVT.

000A A request was received containing an address list MS subvector with multiple entries, but the receiver supports only a single entry in such a subvector.

000B An MS Request Change Control major vector was received requesting automatic delayed acceptance, but the receiver does not support that function.

000C Set aside for implementation specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

000D An MS Request Change Control major vector was received requesting post-test, but the receiver does not support that function.

000E An MS Request Change Control major vector was received prohibiting automatic removal of a change, but the receiver does not support that function.

000F An Activate MS major vector was received specifying use of changes installed in production only, but the receiver supports such a request only when it is received locally.

0010 Reserved

0011 Set aside for implementation specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0014 An MS major vector was received requesting execution window timing, but the receiver does not support that function.

0015 An MS Activate major vector was received specifying change management activation use, but the receiver does not support that function.

0016 An MS Request Change Control major vector was received requesting Activate with Force Delay, but the receiver does not support that function.

0017 The changes referred to in a Request Change Control MS major vector are already installed on trial and the receiver does not support the transfer from trial to production with removability = yes.

0018 An MS Request Change Control major vector was received requesting pre-test, but the receiver does not support that function.

4001, 4003 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

080D NAU Contention: A request to activate a session was received while the receiving half-session was awaiting a response to a previously sent activation request for the same session; for example, the SSCP receives an ACTCDRM from the other SSCP before it receives the response for an ACTCDRM that it sent to the other SSCP and the SSCP ID in the received ACTCDRM was less than or equal to the SSCP ID in the ACTCDRM previously sent.

080E NAU Not Authorized: The requesting NAU does not have access to the requested resource.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The PU, according to its system definition, does not accept an ACTPU from any SSCP having the network ID of the sending SSCP.

0002 A gateway T4 node received a dynamic dump request from an SSCP that is not in the native network of the gateway T4 node.

0006 A network node received a Register from an unauthorized end node.

0009 A network node received a Delete from an unauthorized end node.

000B A Locate/CD-Initiate was received from a node that is not defined as a client end node. This can be detected by either DS or SS.

000C A gateway T4 node received a dynamic dump request from an SSCP that is not in the native network of the gateway T4 node.

080F End User or LU Not Authorized: The requesting end user or LU does not have the proper security authorization to access the requested resource.

0000 No specific code applies.

0983 Access Denied: The request specifies a resource that the requester is not permitted to access. Access to a resource is controlled by evaluation of the requester's identity, profile, or location. This sense data is sent in FMH-7.

6051 Access Security Information Invalid: The request specifies an Access Security Information field that is unacceptable to the receiver; for security reasons, no further detail on the error is provided. This sense data is sent in FMH-7 or UNBIND.

0810 Missing Requester ID: The required requester ID was missing.

0811 Break: Asks the receiver of this sense code to terminate the present chain with CANCEL or with an FMD request carrying EC. The half-session sending the Break sense code enters chain-purge state when Break is sent; the half-session receiving the Break sense code discards the terminated chain without ever retransmitting it.

0812 Insufficient Resource: Receiver cannot act on the request because of a temporary lack of resources.

Bytes 2 and 3 may contain the following sense code specific information:

0000 No specific code applies.

0001 More PUs or LUs are requested by RNAA than are present in the pool.

0003 Resources are not currently available to support an XRF session.

0004 The RNAA request indicates that the requested address must be pre-ENA compatible, but no pre-ENA compatible address is available.

0006 Unsuccessful Allocation: The intermediate session router or boundary function lacks resources to support a session connector. The RU being rejected is a BIND.

0007 Insufficient resources are available for LU address allocation.

000A A network node does not have adequate resources to honor a Register request (the available directory capacity has already been reached).

000D Insufficient buffers exist to activate a session.

000F Insufficient buffer space exists to build a BFINIT.

0010 The CP does not have adequate resources to process a GDS variable request; it will deactivate its CP-CP sessions with the partner CP.

0011 Insufficient storage is available to the SNA component to satisfy the request at this time.

0014 This session has failed because of storage depletion at an intermediate node.

0015 Insufficient resources are available to initiate a short-hold mode connection.

0016 Unknown network identifier

0017 Insufficient buffer space exists to process a nonimmediate UNBIND.

0018 All LFSIDs this node is allowed to assign on the TG are in use at this time; the request is rejected.

0019 Insufficient storage is available to conduct an XID exchange.

001A Insufficient storage is available to activate a TG.

001B Insufficient resources to activate a token-ring connection.

001C Insufficient storage exists to respond precisely to an error condition.

Note: This sense data is returned when node buffer resources are in critical depletion and storage cannot be obtained to build a more specific error response.

001D The receiving T4 node does not have sufficient disk space to perform the requested dump.

001E A session has failed because depletion of pooled buffer storage has exceeded a critical threshold resulting from that session's monopolizing usage.

0813 Bracket Bid Reject—No RTR Forthcoming: BID (or BB) was received while the first speaker was in the in-bracket state, or while the first speaker was in the between-brackets state and the first speaker denied permission. RTR will not be sent.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

Note: For LU 6.2, this is the only setting defined.

0001 Bracket Bid Reject: The component was in the in-bracket state when a bracket request was received.

0002 Bracket Bid Reject: The component was in the between-bracket state when a bracket request was received.

0814 Bracket Bid Reject—RTR Forthcoming: BID (or BB) was received while the first speaker was in the in-bracket state, or while the first speaker was in the between-brackets state and the first speaker denied permission. RTR will be sent.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

Note: For LU 6.2, this is the only setting defined.

0815 Function Active: A request to activate a network element or procedure was received, but the element or procedure was already active.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 A session activation request was received by a boundary function to activate a session that was already active.

0002 A session activation request was received by a gateway function to activate a cross-network session that was already active.

0003 Processing for another management services request in progress. Sender should retry the request.

Note: This sense data is sent only by a type 2 node, which may lack sufficient queuing space.

0004 A BIND was received from a T2.1 node when the session is already active; i.e., the LFSID is in use. The receiver rejects the BIND.

0005 An IPL function (the loading or storing of a load module) is in progress.

0006 The short-hold mode connection selected has been recalled on another port.

0007 A session activation request was received by a network node or an end node to activate a CP-CP session that was already active.

0816 Function Inactive: A request to deactivate a network element or procedure was received, but the element or procedure was not active.

0817 Link or Link Resource Inactive: A request requires the use of a link or link resource that is not active.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Link inactive.

0002 Link station inactive.

0003 Switched link connection inactive.

0004 The TG number of the desired link has been renegotiated to a new value; the route cannot be activated.

0006 The link between an SNA node and an attached processor is inactive; for example, the connection between the main processor and its attached service processor goes down.

4001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0818 Link Procedure in Process: CONTACT, DISCONTACT, IPL, or other link procedure in progress when a conflicting request was received.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001,0002 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0003 CONTACT Not Serialized, Retry: An initial CONTACT procedure is in progress and a nonactivation CONTACT was received by the PU. The nonactivation CONTACT is rejected until the initial CONTACT procedure is completed.

0004 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0005 Link problem determination test for a modem in progress.

0006 On-line terminal test in progress.

0007 SDLC link test, level 2, in progress.

0009 The requested test was not initiated because another test was already in progress.

0819 RTR Not Required: Receiver of Ready To Receive has nothing to send.

081A Request Sequence Error: Invalid sequence of requests.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 An ACTLU was received and no SSCP-PU session exists.

0002 An IPL or DUMP RU sequence error has occurred.

0004 An NC-ER-TEST was to be sent as a result of receiving a ROUTE-TEST request. The ROUTE-TEST was sent in one subnetwork, the NC-ER-TEST was to be sent in another. The SSCP sending the ROUTE-TEST did not have a required alias address within the subnetwork where the NC-ER-TEST was to be sent. (Before sending ROUTE-TEST, the SSCP sends RNAA, or the installation predefines the alias address, so that an origin SSCP address is available within the subnetwork of the route being tested. This address is then specified in the NC-ER-TEST RU.)

0006 RNAA Rejected: If the PU of the node to which an LU is to be added was RNAA-added and a control vector has not been received, the RNAA is rejected. A SETCV for the PU has not been received and processed.

081B Receiver in Transmit Mode: A race condition exists: a normal-flow request was received while the half-duplex contention state was not-receive, (*S,¬R), or while resources (such as buffers) necessary for handling normal-flow data were unavailable. (Contrast this sense code with X'2004', which signals a protocol violation.)

081C Request Not Executable: The requested function cannot be executed, because of a permanent error condition in the receiver.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0002 The receiver has an error resulting from a software problem that prevents execution of the request.

0005 A power-on request failed.

0006 A processor microcode load request failed.

0007 An operating system load request failed.

0009 A processor microcode quiesce request failed.

000A A power-off request failed.

00B1 An SDLC error was detected during link problem determination for a modem.

00B2 A modem error (for example, modem check) was detected during link problem determination.

00B3 A timeout threshold was exceeded for a link problem determination aid modem response.

00B4 An overrun or underrun occurred in the node using the link connection during link problem determination for a modem.

00B5 Data Check was signaled during LPDA-2 test.

00B6 Format exception was signaled during LPDA-2 test.

00B7 LPDA-2 modem test was attempted and failed because of a communication controller equipment (for example, scanner) error.

00BA The receiver has an error resulting from a microcode problem that prevents execution of the request.

0n0m An error was detected by the DLC manager of the receiving node during the execution of a management services request. If n = X'A', the link connection status has not changed from the state previous to the execution; if n = X'B', the link connection status was modified from the state existing previous to the execution. The error is specified as follows: m = X'1' for volatile storage error, m = X'2' for nonvolatile storage (e.g., file access error), m = X'3' for link connection component (e.g., modem) interface error, and m = X'4' for unspecified software error conditions.

Sense code specific information settings 0004, 0008, 000C, 0010, 0014, 0018, 0020, 0028, 0030, 0034, 0038, 003C, 0040, 0072, 0098, 00AB, 0100 – 0109, 0120 – 0125, 0149, 0189 – 0191, 0200 – 0209, 0220 – 0225, 0290, 0291, 07**, and 08** are all set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

081D Invalid Network Address or Name: A node, station, or CP identifier in the request was found to be invalid. Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 The station ID or SSCP ID in the request was found to be invalid.

0001 The network ID, LU name pair in the request was found to be invalid.

0002 Invalid resource name found in the request.

081E Session Reference Error: The request contained reference to a half-session that either could not be found or was not in the expected state (generally applies to network services requests).

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0003 No session was found during the processing of a session services request.

0004 The appropriate session was found during processing of a session services request, but the session is not in the expected state.

081F Request Was Cancelled by an Operator

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The operator has either canceled the Link Activation request or has deactivated the link.

0820 Control Vector Error: Invalid data for the control vector specified by the target network address and key.

0000 No specific code applies.

0821 Invalid Session Parameters: Session parameters were not valid or not supported by the half-session whose activation was requested.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Invalid Mode Name at LU: The specified mode name was not recognized by the LU.

0002 Invalid Mode Name at CP: The specified mode name was not recognized by the CP.

0003 The primary half-session requires cryptography, but the secondary half-session does not support cryptography.

0004 The secondary half-session requires cryptography, but the primary half-session does not support cryptography.

0005 Selective or mandatory cryptography is specified, but no SLU cryptographic data key is provided.

0006 The BIND was rejected because it was nonnegotiable and specified a primary send pacing window size larger than the SSCP or BF can handle.

0822 Link Procedure Failure: A link-level procedure has failed because of a link equipment failure, a loss of contact with a link station, or an invalid response to a DLC command. (This is not a path error, since the request being rejected was delivered to its destination.)

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 An IPLINIT request was received by a T4 node, resulting in a link disconnection. The sender should reinitialize the dial connection and resend the IPLINIT request.

0823 Unknown Control Vector: The control vector specified by a network address and key is not known to the receiver.

0824 Logical Unit of Work Aborted: The current unit of work has been aborted; when sync point protocols are in use, both sync point managers are to revert to the previously committed sync point.

For LU 6.2, this sense data is sent only in FMH-7.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 For LU 6.2, Backout Initiated—No Resync In Progress: A transaction program or its LU has initiated backout. The protected resources for the distributed logical unit of work are to be restored to the previously committed sync point.

When sent in reply to a PS header, no resync in progress means that all resources in the transaction subordinate to the backout sender have backed out.

For non-LU 6.2, no specific code applies.

0001 For LU 6.2, Backout Initiated — Resync In Progress: A transaction program or its LU has initiated backout. The protected resources for the distributed logical unit of work are to be restored to the previously committed sync point.

When sent in reply to a PS header, resync in progress means that one or more resources in the transaction subordinate to the backout sender have experienced failure so it is not known whether they've backed out.

0825 Component Not Available: The LU component (a device indicated by an FM header) is not available.

0826 FM Function Not Supported: A function requested in an FMD RU is not supported by the receiver. (*Note*: X'1003' has displaced this value for reporting such conditions.)

0827 Intermittent Error—Retry Requested: An error at the receiver caused an RU to be lost. The error is not permanent, and retry of the RU (or chain) is requested.

0828 Reply Not Allowed: A request requires a normal-flow reply, but the outbound data flow for this half-session is quiesced or shut down, and there is no delayed reply capability.

0829 Change Direction Required: A request requires a normal-flow reply, but the half-duplex flip-flop state (of the receiver of the request) is not-send, and CD was not set on the request. Therefore, there is no delayed reply capability.

082A Presentation Space Alteration: Presentation space altered by the end user while the half-duplex state was not-send, (¬S,*R); request executed.

082B Presentation Space Integrity Lost: Presentation space integrity lost (for example, cleared or changed) because of a transient condition—for example, because of a transient hardware error or an end user action such as allowing presentation services to be used by the SSCP. (*Note*: The end-user action described under X'082A' and X'084A' is excluded here.)

082C Resource-Sharing Limit Reached: The request received from an SSCP was to activate a half-session, a link, or a procedure, when that resource was at its share limit.

082D LU Busy: The LU resources needed to process the request are being used; for example, the LU resources needed to process the request received from the SSCP are being used for the LU-LU session.

082E Intervention Required at LU Subsidiary Device: A condition requiring intervention, such as out-of-paper, power-off, or cover interlock open, exists at a subsidiary device.

082F Request Not Executable because of LU Subsidiary Device: The requested function cannot be executed, because of a permanent error condition in one or more of the receiver's subsidiary devices.

0830 Session-Related Identifier Not Found: The receiver could not find a session-related identifier for a specified session.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 PCID not found for the specified resources.

0002 LSID not found for the specified session.

0831 LU Component Disconnected: An LU component is not available because of power-off or some other disconnecting condition.

0832 Invalid Count Field: A count field contained in the request indicates a value too long or too short to be interpreted by the receiver, or the count field is inconsistent with the length of the remaining fields.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnnn Bytes 2 and 3 contain a binary count that indexes (0-origin) the first byte of the invalid count field.

Note: This sense code is not used for a BIND error because the displacement of fields within the BIND may not be the same at both ends of a session when the BIND was affected by name transformations—for example, after the BIND has passed through a gateway. Sense code X'0835' is used to specify a displacement for a BIND error.

0833 Invalid Parameter (with Pointer and Complemented Byte): One or more parameters contained in fixed- or variable-length fields of the request are invalid or not supported by the NAU that received the request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnmm Byte 2 contains a binary value that indexes (0-origin) the first byte that contained an invalid parameter.

Byte 3 contains a transform of the first byte that contained an invalid parameter: the bits that constitute the one or more invalid parameters are complemented, and all other bits are copied.

Note: This sense code is not used for a BIND error because the displacement of fields within the BIND may not be the same at both ends of a session when the BIND was affected by name transformations—for example, after the BIND has passed through a gateway. Sense code X'0835' is used to specify a displacement for a BIND error.

0834 RPO Not Initiated: A power-off procedure for the specified node was not initiated because one or more other SSCPs have contacted the node, or because a CONTACT, DUMP, IPL, or DISCONTACT procedure is in progress for that node.

0835 Invalid Parameter (with Pointer Only): The request contained a fixed- or variable-length field whose contents are invalid or not supported by the NAU that received the request.

nnnn Bytes 2 and 3 contain a two-byte binary count that indexes (0-origin) the first byte of the fixed- or variable-length field having invalid contents.

Note: This sense code is not used to report an invalid value in an MS major vector. If the invalid value occurs in a formatted MS subvector, sense code X'086B' is used. If it occurs in an unformatted subvector, sense code X'0870' is used.

0836 PLU/SLU Specification Mismatch: For a specified LU-LU session, both the origin LU (OLU) and the destination LU (DLU) have only the primary capability or have only the secondary capability.

0837 Queuing Limit Exceeded: For an LU-LU session initiation request (INIT, CDINIT, or INIT-OTHER-CD) specifying (1) Initiate or Queue (if Initiate not possible) or (2) Queue Only, the queuing limit of either the OLU or the DLU, or both, was exceeded.

0838 Request Not Executable Because of Resource or Component State Incompatibility: The request is not executable because it is not compatible with the state of a resource or component in the receiver.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be deleted or replaced because it is installed marked removable.

0002 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed, removed, or accepted because they are in back-level state (see Note).

0003 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked on-trial because they are already installed marked on-trial (see Note).

0004 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked on-trial or in-production because they are already installed marked in-production removably (see Note).

0005 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked on-trial or in-production because they are already installed marked in-production and nonremovable (see Note).

0006 One or more of the changes referred to in a Request Change Control MS major vector cannot be removed or accepted because they are installed marked nonremovable (see Note).

0007 One or more of the changes referred to in a Request Change Control MS major vector cannot be removed or accepted because they are not installed (see Note).

0008 Pre-test is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

0009 Execution window timing is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000A Automatic removal is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000B Post-test is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000C Automatic delayed acceptance is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000D One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked in-production because they are installed marked on-trial with a set of corequisites different from those requested on this install request.

One or more reported-on token strings are used to identify the corequisite changes currently installed when the report code is carried in an SNA condition report.

000E One or more of the changes referred to in a Request Change Control MS major vector cannot be accepted because they are installed marked on-trial (see Note).

000F One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be replaced or deleted because they are critical system components that must always have an installed instance. The only possibility is to perform data object renewal using Send-and-Install with removability prohibited or desired (but not required) (see Note).

0010 One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be stored or installed because an implementation-defined limit on the number of changes has been exceeded (see Note).

0011 One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be deleted or replaced because they are required in order to maintain removability of other changes. They may be in backup state or installed marked in-production (see Note).

0012 One or more of the corequisite changes referred to in a Request Change Control MS major vector are missing or are in a state incompatible with the request (see Note).

0013 The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be replaced because it is installed marked in-production and non-removable and another change is not being installed in this operation (see Note).

0014 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed because a precluded combination of values in the Removability, Automatic Removal, Automatic Acceptance, or Activation Use subfields was specified (see Note).

0015 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed because one or more changes already installed are still removable for one or more components to be altered by these changes (see Note).

0016 One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be replaced because they would be required for removable installation, and removability is required (see Note).

0017 Execution of the request referred to in an MS Cancel major vector has proceeded too far to cancel.

001A The request will not be honored because it was either queued or active at a node at the time a local operator assumed control of the node, thus effecting its cancellation.

001B The request will not be honored because it was submitted to a node at a time when a local operator or other application was in control of the node.

001C One or more of the changes referred to in a Request Change Control MS major vector cannot be installed removably because the implementation does not support removability for certain classes of data objects (see Note).

001D One or more of the changes referred to in a Request Change Control MS major vector cannot be installed because the implementation precludes corequisite specification for certain classes of data objects (see Note).

001E One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be installed or stored because the implementation has identified a prerequisite change that is either not installed or is installed at an incompatible level (see Note).

001F The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be stored because the implementation has identified a missing larger entity that must first be stored before the subentity may be stored (see Note).

0020 One or more of the changes referred to in a Request Change Control MS major vector are installed in production but cannot be removed or accepted because they are required in order to maintain removability of other changes (see Note).

0021 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed, removed, or accepted because the implementation does not support certain classes of data objects (see Note).

0022 The change referred to in a Report-FS-Action command cannot be stored because another change having the same data object class already exists in sent state and the implementation prohibits more than one change of the same object class to exist in sent state for certain classes of objects. The previous change must first be installed or deleted.

A reported-on token string is used to identify the change currently in sent state when the report code is carried in an SNA condition report.

0023 The request will not be honored because a system resource file was locked at the time execution was attempted.

0024 One or more of the changes referred to in a Request Change Control MS major vector could not be installed because an unexpected error was encountered while performing the installation process (see Note).

0025 The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be stored because a reactivation of the node must first be performed (see Note).

0026 The target group object of one or more changes referred to in a Request Change Control MS major vector does not exist (see Note).

0027 The target group object of one or more changes referred to in a Request Change Control MS major vector is not accessible (see Note).

0028 The class code of one or more changes referred to in a Request Change Control MS major vector is inconsistent with the class code of the target group object for the affected component (see Note).

0029 The installation of one or more changes referred to in a Request Change Control MS major vector would require nesting of group objects, which is not supported (see Note).

002A The class code of one or more changes referred to in a Request Change Control MS major vector is inconsistent with the class code of installed changes affecting the same component (see Note).

002B The installation procedure for one or more changes referred to in a Request Change Control MS major vector was not found (see Note).

002C The command processor to execute the installation procedure for one or more changes referred to in a Request Change Control MS major vector could not be found (see Note).

002D The name of one or more changes referred to in a Request Change Control MS major vector does not contain an architecturally defined subtree as required by the receiver (see Note).

002E The specification of Alter_Active_Components = No is not supported by the receiver for the class code of one or more changes referred to in a Request Change Control MS major vector (see Note).

002F No group object is defined, and either insufficient or illegal default group information was provided for one or more changes referred to in a Request Change Control MS major vector (see Note).

0030 One or more of the changes referred to in a Request Change Control MS major vector cannot be removed (or accepted) because a previous Accept (or Remove) request has failed and its effects could not be backed out. Only a retry of the same request can be attempted (see Note).

0031 One or more of the changes referred to in a Request Change Control MS major vector are not independent and can not be installed as corequisites (see Note).

0032 The modification level of one or more changes referred to in a Request Change Control MS major vector is inconsistent with the modification level of installed changes affecting the same component (see Note).

0033 The version tokens of one or more changes referred to in a Request Change Control MS major vector are inconsistent with the version tokens of installed changes affecting the same component (see Note).

0034 The request will not be honored because it was canceled by the operating system at the node to which it was sent.

0035 The class code specified in a Request Change Control MS major vector for one or more changes is inconsistent with the class code specified in the local FS catalog for the same object (see Note).

0036 Access to one or more local files associated with one or more changes referred to in a Request Change Control MS major vector was denied (see Note).

Note: One or more reported-on token strings are used to identify these changes when the report code is carried in an SNA condition report.

0839 LU-LU Session Being Taken Down or LU Being Deactivated.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 During session-initiation processing, a session-termination request has caused the LU-LU session to be taken down.

0002 RNAA(Type 3) received for a session during the process of session deactivation. The RNAA should be retried.

0003 SSCP detected that this session should no longer exist and requested its termination. For example, a BFSESSINFO was received reporting a subject LU address that the SSCP believed already belonged to an other-domain resource.

083A LU Not Enabled: At the time an LU-LU session initiation request is received at the SSCP, at least one of the two LUs, although having an active session with its SSCP, is not ready to accept CINIT or BIND requests.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The PLU is not enabled.

0002 The SLU is not enabled.

083B Invalid PCID: the received PCID for a new session duplicated the PCID assigned to another session, or the received PCID intended as an identifier for an existing session could not be associated with such an

existing session, or an error was detected in the format of the received PCID.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The PCID contained in CDINIT(Initiate or Queue), INIT-OTHER-CD, or CDTAKED duplicates a PCID received previously in one of these requests.

0002 The received fully qualified PCID duplicated one assigned to another session.

0003 The received fully qualified PCID contains a network-qualified CP name identical to that of the receiving node.

083C Domain Takedown Contention: While waiting for a response to a CDTAKED, a CDTAKED request is received by the SSCP containing the SSCP-SSCP primary half-session. Contention is resolved by giving preference to the CDTAKED sent by the primary half-session.

083D Dequeue Retry Unsuccessful—Removed from Queue: The SSCP cannot successfully honor a CDINIT(Dequeue) request (which specifies "leave on queue if dequeue-retry is unsuccessful") to dequeue and process a previously queued CDINIT request (for example, because the LU in its domain is still not available for the specified session), and removes the queued CDINIT request from its queue.

083E Implementation-Defined Retry Limit Exhausted.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The implementation-defined limit on XID exchanges was exceeded before link activation completed.

0002 The implementation-defined limit on XID exchanges was exceeded before a nonactivation exchange completed.

0004 The implementation-defined limit on contention-winner CP-CP session activation attempts has been exceeded.

083F Terminate Contention: While waiting for a response to a CDTERM, a CDTERM is received by the SSCP of the SLU. Contention is resolved by giving preference to the CDTERM sent by the SSCP of the SLU.

0840 Procedure Invalid for Resource: The received RU is not supported in the receiver for this type of resource (for example, (1) SETCV specifies boundary function support for a type 1 node but the capability is not supported by the receiving node, or (2) the PU receiving an EXECTEST or TESTMODE is not the primary PU for the target link.)

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0007 Resource Not Found: A Delete or Find could not be satisfied because the specified entry does not exist in the receiver's directory.

000A Procedure Invalid for Resource: A T2.1 node supporting independent LUs has dialed into a boundary node that does not support sessions with independent LUs.

000C Conflicting Entry Type on Delete: The Delete request attempted to delete a home entry, i.e., one defined at the receiver by its own network operator facility (NOF).

0011 A dynamically added or a switched resource has not yet been activated.

0841 Duplicate Network Address: In an LU-LU session initiation request, one of the specified LUs has a duplicate network address already in use.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 The SSCP of the DLU determines that the OLU network address specified in the CDINIT request is a duplicate of an LU network address assigned to a different LU name.

0001 A duplicate SLU address is found during session initiation.

0002 A duplicate PLU address is found during session initiation.

0003 An SSCP finds a duplicate network address for the DLU on the OLU side of the gateway.

0004 An SSCP finds a duplicate network address for the DLU on the DLU side of the gateway.

0005 An SSCP finds a duplicate network address for the OLU on the OLU side of the gateway.

0006 An SSCP finds a duplicate network address for the OLU on the DLU side of the gateway.

0008 An ACTCDRM request was received that contained a network address already in use.

0842 Session Not Active.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 SSCP-SSCP Session Not Active: The SSCP-SSCP session, which is required for the processing of a network services request, is not active; for example, at the time an LU-LU session initiation or termination request is received, at least one of the following conditions exists:

- The SSCP of the ILU and the SSCP of the OLU do not have an active session with each other, and therefore INIT-OTHER-CD cannot flow.

- The SSCP of the OLU and the SSCP of the DLU do not have an active session with each other, and therefore CDINIT or CDTERM cannot flow.

Note: This value is used if there is not enough data to select one of the more specific codes listed below.

0001 The session between T2.1 CPs is not active.

0002 For a session-initiation request, an SSCP does not have an SSCP-SSCP session with an SSCP in the direction of the DLU.

0003 For a session-initiation request, an SSCP does not have an SSCP-SSCP session with an SSCP in the direction of the OLU.

0004 An intermediate SSCP has lost connectivity with an SSCP in the session setup path for an LU-LU session. This sense data is used when the SSCP previously lost connectivity with one or more participating gateway nodes so that it cannot learn that the LU-LU session is ended by receiving a NOTIFY RU from a gateway node.

FFFF The session is not active because the session initiation request has been transferred to another PLU.

0843 Required Synchronization Not Supplied: For example, a secondary LU (LU type 2 or 3) received a request with Write Control Code = Start Print, along with RQE and ¬CD.

0844 Initiation Dequeue Contention: While waiting for a response to a CDINIT(Dequeue), a CDINIT(Dequeue) is received by the SSCP of the SLU. Contention is resolved by giving preference to the CDINIT(Dequeue) sent by the SSCP of the SLU.

0845 Permission Rejected—SSCP Will Be Notified: The receiver has denied an implicit or explicit request of the sender; when sent in response to BIND, it implies that the secondary LU will notify the SSCP (via NOTIFY vector key X'0C') when a BIND can be accepted, and the SSCP of the SLU supports the notification. (See the X'080A' sense code for a contrasting response.)

0846 ERP Message Forthcoming: The received request was rejected for a reason to be specified in a forthcoming request.

0847 Restart Mismatch: Sent in response to STSN, SDT, or BIND to indicate that the secondary half-session is trying to execute a resynchronizing restart but has received insufficient or incorrect information.

0848 Cryptography Function Inoperative: The receiver of a request was not able to decipher the request because of a malfunction in its cryptography facility.

0849 User Names Lost: An exception condition has resulted in the loss of user names associated with the identified message unit.

084A Presentation Space Alteration: The presentation space was altered by the end user while the half-duplex state was not-send, (¬S,*R); request not executed.

084B Requested Resources Not Available: Resources named in the request, and required to honor it, are not currently available. It is not known when the resources will be made available.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0003 The application transaction program specified in the request is not available.

0005 Controller resource is not available.

6002 The resource identified by the destination program name (DPN) is not supported.

6003 The resource identified by the primary resource name (PRN) is not supported.

6031 Transaction Program Not Available—Retry Allowed: The FMH-5 Attach command specifies a transaction program that the receiver is unable to start. Either the program is not authorized to run or the resources to run it are not available at this time. The condition is temporary. The sender is responsible for subsequent retry. This sense data is sent only in FMH-7.

084C Permanent Insufficient Resource: Receiver cannot act on the request because resources required to honor the request are permanently unavailable. The sender should not retry immediately because the situation is not transient.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 For LU 6.2, Transaction Program Not Available—No Retry: The FMH-5 Attach command specifies a transaction program that the receiver is unable to start. The condition is not temporary. The sender should not retry immediately. This sense data is sent only in FMH-7.

For non-LU 6.2, no additional information is specified.

0001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0002 Creating Allocation Exception: The receiver is unable to create the specified data object as a result of an insufficient storage condition that occurred at allocation time. When this SNA report code is used in an SNA condition report, it is optionally accompanied by one or more structure reports that identify the allocation requests that failed.

0003 Replacing Allocation Exception: The receiver is unable to replace the specified data object as a result of an insufficient storage condition that occurred at allocation time. When this SNA report code is used in an SNA condition report, it is

optionally accompanied by one or more structure reports that identify the allocation requests that failed.

0004 Reserved

0005 Reserved

0006 Data-Object Storing Exception: The receiver is unable to store the specified data object as a result of an insufficient storage condition that occurred during the storing process. When this SNA report code is used in an SNA condition report, it is optionally accompanied by one or more structure reports that identify containing the allocation requests that failed.

0007 Data-Object Classification Code Not Supported: The receiver is unable to satisfy the allocation requirements of the specified data-object classification code. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report containing the data-object classification code that failed.

0008 Volume Not Mounted: The receiver is unable to perform the requested allocation/storing operation because the required volume is not mounted. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report identifying the volume that was not mounted.

hnnn where h≥8, i.e., the high-order bit in byte 2 is set to 1. The 15 low-order bits of bytes 2 and 3 contain a binary count that indexes (zero-origin) the first byte of the field found to be in error.

084D Retired

084E Invalid Session Parameters—PRI: A positive response to an activation request (for example, BIND) was received and was changed to a negative response because of invalid session parameters carried in the response. The LU receiving the response will send a deactivation request for the corresponding session.

084F Resource Not Available: A requested resource is not available to service the given request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The receiver's disk is full; therefore, a received load module cannot be stored.

0002 Security component not available: The security component required to process the request is currently not available. This sense data is sent only in UNBIND and -RSP(BIND) records.

0003 A coded graphics character set ID (CGCSID) needed to interpret the request is not supported by the receiver. When this report code is used in an SNA condition report, it is accompanied by a

supplemental report containing the 2-byte CGCSID not supported.

0850 Link-Level Operation Cannot Be Performed: An IPL, dump, or RPO cannot be performed through the addressed link station because the system definition or current state of the hardware configuration does not allow it.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Link Activation Limit Reached: The specified TG was not activated because the maximum number of active link stations allowed on this port has already been reached.

0851 Session Busy: Another session that is needed to complete the function being requested on this session is temporarily unavailable.

0852 Duplicate Session Activation Request: Two session activation requests have been received with related identifiers. The relationship of the identifiers and the resultant action varies by request.

If the RU is an ACTPU or ACTCDRM, it means that a session has already been activated for the subject destination-origin pair by a session activation request that carried a larger activation request identifier than the current request; the current request is refused.

If the RU is a BIND, it means that the BIND request was received with the same session instance identifier (in the structured subfield X'03' of the User Data field) as an active session's; the current request is refused.

0853 TERMINATE(Cleanup) Required: The SSCP cannot process the termination request, as it requires cross-domain SSCP-SSCP services that are not available. (The corresponding SSCP-SSCP session is not active.) TERMINATE(Cleanup) is required.

0854 Retired

0856 SSCP-SSCP Session Lost: Carried in the Sense Data field in a NOTIFY (Third-Party Notification vector, X'03') or -RSP(INIT_OTHER) sent to an ILU to indicate that the activation of the LU-LU session is uncertain because the SSCP(ILU)-SSCP(OLU) session has been lost. (Another sense code, X'0842', is used when it is known that the LU-LU session activation cannot be completed.)

0857 SSCP-LU Session Not Active: The SSCP-LU session, required for the processing of a request, is not active; for example, in processing REQECHO, the SSCP did not have an active session with the target LU named in the REQECHO RU.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The SSCP-SLU session is in the process of being reactivated.

0002 The SSCP-PLU session is inactive.

0003 The SSCP-SLU session is inactive.

0004 The SSCP-PLU session is in the process of being reactivated.

0005 The SSCP lost connectivity with the PLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0006 The SSCP lost connectivity with the SLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0007 The selected ALS for the OLU is not in a state permitting LU-LU sessions to be established using it. The condition is detected when the session request (BFINIT) was received, but, when the request was processed, the ALS was no longer in an active state. The session request is rejected.

0008 The selected ALS for the DLU is not in a state permitting LU-LU sessions to be established using it. The condition is detected when the session request was being processed in the DLU domain and the ALS selected for the DLU is no longer in an active state. The session request is rejected.

0859 REQECHO Data Length Error: The specified length of data to be echoed (in REQECHO) violates the maximum RU size limit for the target LU.

085A Specific Server Exception: An architecturally defined or user-defined server that is sensitive to data object contents, has detected an exception.

085B Unknown Resource Name: The identified resource, required to complete the requested unit-of-work, is not known to the SNA node.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Unknown server name. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report containing the server name.

0002 Unknown agent.

0003 The clock identifier specified in an MS Set Clock major vector is unknown to the receiver.

0004 The timing source name specified in an MS Set Clock major vector is unknown to the receiver.

0005 The agent unit-of-work correlator referred to by an MS Cancel major vector is unknown to the receiver, or represents a unit of work already completed.

0006 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

085C System Exception: The node experiences an exception condition within a resident system or subsystem that inhibits subsequent processing by the SNA component.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The exception is identifiable as a system-related problem.

0002 The exception is identifiable as a permanent system-related problem.

085D MU-ID Could Not Be Accepted in the MU-ID Registry.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0001 The MU-ID is a duplicate. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify information about the receiver's MU-ID registry: supplemental report 1 contains the lowest MU-ID the receiver would accept; supplemental report 2 contains the highest MU-ID the receiver would accept; supplemental report 3 contains the time stamp of the receiver's MU-ID registry.

0002 The MU-ID value is greater than expected. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify information about the receiver's MU-ID registry: supplemental report 1 contains the lowest MU-ID the receiver would accept; supplemental report 2 contains the highest MU-ID the receiver would accept; supplemental report 3 contains the time stamp of the receiver's MU-ID registry.

0003 A temporary condition prevents acceptance of the MU-ID.

0004 A permanent condition prevents acceptance of the MU-ID.

0005 The MU-ID registry is not initialized.

085E Operator Intervention

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The operator has suspended the transmission of the message unit.

0002 The operator has purged the message unit.

0860 Function Not Supported—Continue Session: The function requested is not supported; the function may have been specified by a request code or some other field, control character, or graphic character in an RU.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnnn Bytes 2 and 3 contain a 2-byte binary count that indexes (0-origin) the first byte in which an error was detected. This sense data is used to request that the session continue, thereby ignoring the error.

0861 Invalid COS Name: The class-of-service (COS) name, either specified by the ILU or generated by the SSCP of the SLU from the mode table is not in the "COS name to VR identifier list" table used by the SSCP of the PLU.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 COS name was generated by the SSCP.

0001 COS name was generated by the ILU.

0002 The COS name generated by the T2.1 CP local to, or the T2.1 NNCP server for, the ILU is not in the COS name definition table.

0003 The CDINIT request or response contains a Session Initiation (X'14') control vector that has Class-of-Service (COS) Name fields that have not been properly specified.

0862 Medium Presentation Space Recovery: An error has occurred on the current presentation space. Recovery consists of restarting at the top of the current presentation space. The sequence number returned is of the RU in effect at the top of the current presentation space.

nnnn Bytes 2 and 3 following the sense code contain the byte offset from the beginning of the RU to the first byte of the RU that is displayed at the top of the current presentation space.

0863 Referenced Local Character Set Identifier (LCID) Not Found: A referenced character set does not exist.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code appplies.

hnnn where h≥8, i.e., the high-order bit in byte 2 is set to 1. The 15 low-order bits of bytes 2 and 3 contain a binary count that indexes (0-origin) the first byte of the field found to be in error.

0864 Function Abort: The conversation was terminated abnormally. Other terminations may occur after repeated reexecutions; the request sender is responsible to detect such a loop.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 For LU 6.2, Premature Conversation Termination: The conversation is terminated abnormally; for example, the transaction program may have issued a DEALLOCATE_ABEND verb, or the program may have terminated (normally or abnormally) without explicitly terminating the conversation. This sense data is sent only in FMH-7 or UNBIND.

For non-LU 6.2, no additional information is specified.

0001 System Logic Error—No Retry: A system logic error has been detected. No retry of the conversation should be attempted. This sense data is sent only in FMH-7 or UNBIND.

0002 Excessive Elapsed Time—No Retry: Excessive time has elapsed while waiting for a required action or event. For example, a transaction program has failed to issue a conversation-related protocol boundary verb. No retry of the conversation should be attempted. This sense data is sent in UNBIND when there is no chain to respond to; otherwise, it is sent in FMH-7.

0865 Retired

0866 Retired

0867 Sync Event Response: Indicates a required negative response to an (RQE,CD) synchronizing request.

0868 No Panels Loaded: Referenced format not found because no panels are loaded for the display.

0869 Panel Not Loaded: The referenced panel is not loaded for the display.

086A Subfield Key Invalid: A subfield key in an MS subvector was not valid in the conditions under which it was processed.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnmm Byte 2 following the sense code contains the subvector key (nn) of the subvector containing the unrecognized subfield, and byte 3 contains the unidentified subfield key (mm).

086B Subfield Value Invalid: A value in a subfield within an MS major vector is invalid for the receiver.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnmm Byte 2 following the sense code contains the subvector key (nn) of the subvector containing the subfield with the invalid value, and byte 3 contains the subfield key (mm) of the subfield with the invalid value.

Note: See sense code X'0870' for the case in which the invalid value occurs in an unformatted subvector, that is, one not containing subfields with keys and lengths, or in the unformatted portion of a partially formatted subvector.

086C Required Control Vector or Subvector Missing: One or more control vectors or MS subvectors that are required by the receiver to perform some function are missing from the received message, or are not present in the required position.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nn00 Byte 2 following the sense code contains the key (nn) of one of the control vectors or subvectors that is missing, or improperly positioned. Byte 3 is reserved (00).

Note: See the X'080C0006' sense data for the case in which the major vector key is recognized but a subvector representing the function to be performed cannot be identified.

086D Required Subfield Missing: A control vector or MS subvector lacks one or more subfield keys that are required by the receiver to perform the function requested.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnmm Byte 2 following the sense code contains the key (nn) of the subvector or control vector lacking a required subfield, and byte 3 contains the subfield key (mm) of a missing subfield.

086E Invalid Subvector Combination: Two or more subvectors, each permissible by itself, are present in a combination that is not allowed.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnmm Bytes 2 and 3 following the sense code contain the subvector keys (nn) and (mm) of two of the subvectors that should not be jointly present.

086F Length Error: A length field within a structure is invalid, or two or more length fields are incompatible.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The MS major vector length is incompatible with the RU length.

0002 The sum of the MS subvector lengths is incompatible with the MS major vector length.

nn03 The sum of the subfield lengths in a MS subvector is incompatible with the subvector length. Byte 2 following the sense code contains the subvector key (nn).

nn05 MS subvector length invalid. Byte 2 following the sense code contains the relevant subvector key (nn). (This is specified only if the sum of the subvector lengths is compatible with the major vector length.)

nn06 Subfield length invalid. Byte 2 following the sense code contains the subvector key (nn) of the MS subvector containing the invalid subfield length. (This is specified only if the sum of the subfield lengths is compatible with the subvector length.)

0007 The length field of an MDS_MU is incompatible with the sum of the lengths of the imbedded GDS variables or an invalid length was found in an imbedded structure (or GDS variable).

0008 The length field of a CP-MSU is incompatible with the sum of the lengths of the imbedded structures.

0870 Unformatted Subvector Value Invalid: A value in an unformatted MS subvector, or in an unformatted portion of a partially formatted MS subvector, is invalid.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnxx Byte 2 following the sense code contains the subvector key (nn) of the MS subvector containing the invalid value. Byte 3 contains a one-byte binary count that indexes the first byte in which the invalid value falls. The indexing is zero-origin, from the beginning of the subvector.

Note: See sense code X'086B' for the case in which the invalid value occurs in a formatted MS subvector, that is, one containing subfields with keys and lengths, or in the formatted portion of a partially formatted subvector.

0871 Read Partition State Error: A Read Partition structured field was received while the display was in the retry state.

0872 Explicit or Implied Orderly Deactivation Refused

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 An NC_DACTVR(Orderly) request has been received, but sessions are assigned to the VR and it will not be deactivated.

0001 An MS major vector specifying or implying orderly deactivation of the receiving node has been received, but the node is not in a quiesced state and deactivation is not allowed; the requested action will not proceed.

0002 An MS major vector specifying or implying orderly deactivation of the receiving node has been received, but the receiver cannot determine if a quiesced state has been attained; the requested action will not proceed.

0873 Virtual Route Not Defined: No ERN is designated to support this VRN.

0874 ER Not in a Valid State: The ER supporting the requested VR is not in a state allowing VR activation.

0875 Incorrect or Undefined Explicit Route Requested: The reverse ERNs specified in the NC-ACTVR do not contain the ERN defined to be used for the VR requested, or the ERN designated to be used for the VR is not defined.

0876 Nonreversible Explicit Route Requested: The ERN used by the NC-ACTVR does not use the same sequence of transmission groups (in reverse order) as the ERN that should be used for the RSP(NC-ACTVR).

0877 Resource Mismatch: The receiver of a request has detected a mismatch between two of the following: (1) its definition of an affected resource, (2) the actual configuration, and (3) the definition of the resource as implied in the request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Link Defined as Switched Is Nonswitched: A link defined to an ACTLINK receiver as being switched was found to be nonswitched during the activation attempt.

0002 Link Defined as SDLC Is Non-SDLC: A link defined to an ACTLINK receiver as being SDLC was found to be non-SDLC during the activation attempt.

0003 Link Defined as Having Automatic Connect-Out Capability Does Not: A link defined to an ACTLINK receiver as having automatic connect-out capability was found to lack it during the activation attempt.

0004 ACTLINK Received for a Resource Other Than a Link: An ACTLINK was received that resolved to a local device address representing a device other than a link.

0005 Link defined as X.21 is not X.21.

0006 Link defined as LPDA-capable is configured in NRZI mode.

0007 A request that is allowed only for a primary link station was received for a link station that is defined to the receiver as secondary.

0008 A request for link problem determination for modems was received for a link that is defined to the receiver as not supporting link problem determination for modems.

0009 A request for link problem determination for modems was received for a link that is defined to the receiver as supporting link problem determination for modems, but no link station supporting link problem determination for modems was found on the link.

000A A request that is allowed only for a nonswitched link was received for a link that is defined to the receiver as switched.

000B A request that is allowed only for a link with a modem not using the multiplexed links feature was received for a link that is defined to the receiver as having a modem using the multiplexed links feature.

000C Resource Definition Mismatch for Modems: A request that is allowed only for a link with a non tailed modem was received for a link that is defined to the receiver as having a tailed modem. LU.

0028 An RNAA(Move) was received for a link station, and the link station's primary-secondary role is incompatible with the target link.

0029 The RU refers to a resource, and the sender and receiver disagree about its status. One considers it a static resource, the other a dynamic resource.

002A A session cannot be activated because the node does not support segment generation and the maximum link BTU size is too small to satisfy a requirement on the minimum send RU size as defined for the session mode.

002B A session cannot be activated because the node does not support segment reassembly and the maximum link BTU size is too small to satisfy a requirement on the minimum receive RU size as defined for the session mode.

002C A BFINIT session request was received from a PLU that is not in the same network as this SSCP, or a BFSESSINFO was received reporting a subject LU in another network.

002D BFSESSINFO was received for an independent subject LU, but the reported LU is considered by the receiver as a dependent LU.

002E BFSESSINFO was received reporting a dynamic subject LU that the receiver considers to be located under a different ALS from that reported in the BFSESSINFO. The SSCP will attempt to correct this configuration mismatch.

002F BFSESSINFO was received reporting a subject LU that the receiver considers to be located under a different ALS from that reported in the BFSESSINFO. The SSCP cannot correct this configuration mismatch.

0030 BFSESSINFO was received for a subject LU, but the receiver has the address associated with a different LU, which it considers to be static.

0031 BFSESSINFO was received for a subject LU, but the receiver has the address associated with anything other than a static LU or an other-domain resource.

0032 BFSESSINFO was received for a subject LU that is verified, but, for a given session, either the partner LU is reported as the primary and the receiver does not consider that LU to be primary-capable, or the partner LU is reported as the secondary

and the receiver does not consider that LU to be secondary-capable.

0033 Upon receipt of BFSESSINFO, the receiver considers the control block associated with a partner LU to be an other-domain resource that is not active or an application program that is not active.

0034 Upon receipt of BFSESSINFO, an SSCP is unable to associate the information received about a partner LU to be associated with an LU, an other-domain resource, or an application program.

0035 A network address was returned in RSP(RNAA) that the receiver believes is already associated with a different resource.

0036 BFSESSINFO received containing an invalid ALS address. For example, the ALS does not represent a T2.1 node.

0037 BFSESSINFO received for a subject LU, where the secondary address specified in the BFSESSINFO does not match the secondary address the SSCP believes is associated with the LU.

0038 The subject LU specified in the BFSESSINFO RU is not defined to the SSCP as an LU or an other-domain resource.

0039 A request that is valid only for a switched subarea link was received for a link that is not subarea-capable.

003A A request that is valid only for a non-switched subarea link was received for a subarea dial link.

003B An RNAA(Add) was received for an LU; however, an LU with the same name but a different local address already exists under the specified ALS.

0041 Takeover processing completed, but the SSCP did not receive a BFSESSINFO for a resource that the SSCP believed to be a static, independent LU.

0042 A BFINIT session request was received from a PLU that is not controlled by this SSCP.

0043 A request was received for a nonswitched resource that is valid only for a switched resource.

0044 A CONNOUT requested X.21 dial and auto-call capability was not present; resource mismatch.

0045 The DLU for a session request specified a network ID that did not match the network ID for the ALS providing services for the session.

0047 The OLU for a session request or subject LU for a BFSESSINFO specified a network ID that did not match the network ID for the ALS providing services for the session.

0048 The DLU is an independent LU but the boundary function selected to provide its services is not capable of supporting independent LUs. The condition detected during session initiation processing after the ALS was selected for a switched resource.

0049 During processing of a BFSESSINFO for a subject LU, that LU was found to be inactive.

0878 Insufficient Storage: The storage resource required for a data format is not available.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 CONNOUT contained more dial digits than can be stored by the receiving product.

0879 Storage Medium Exception: An exception has occurred involving a storage medium.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Disk I/O error.

0002 A nonrecoverable I/O exception has been encountered.

087A Format Processing Error: A processing error occurred during data formatting.

087B Resource Unknown: The request contains a session key that does not identify a session known to some gateway node; for example, a session activation request arrives at a gateway node after it has released the address transform for the intended session.

087C SSCP-PU Session Not Active: For example, a gateway SSCP-PU session that is needed to establish an address transform for a requested cross-network LU-LU session was not active.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 An SSCP in the session setup path for an LU-LU session has lost connectivity with a gateway node traversed by the session, and has no other way to learn that the session has ended. An intermediate SSCP sends this sense data to one adjacent SSCP when it had previously lost connectivity with the other adjacent SSCP on the same session setup path. An endpoint SSCP sends this sense data to its adjacent SSCP when it had previously lost connectivity to a dependent LU or the boundary function of an independent LU.

0002 The SSCP lost connectivity with the boundary function of an independent PLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0003 The SSCP lost connectivity with the boundary function of an independent SLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

087D Session Services Path Error: A session services request cannot be rerouted along a path of SSCP-SSCP sessions. This capability is required, for example, to set up a cross-network LU-LU session.

Bytes 2 and 3 contain sense code specific information that indicates the specific reason for not rerouting the request. Settings allowed are:

0000 No specific code applies.

0001 An SSCP has attempted unsuccessfully to reroute a session services request to its destination via one or more adjacent SSCPs; this value is sent by a gateway SSCP when it has exhausted trial-and-error rerouting.

Note: This code is used when SSCP rerouting fails completely. The remaining codes are used for failures to reroute to a particular SSCP. For example, they are associated with specific SSCPs when information about a rerouting failure is displayed in the node that was trying to reroute.

0002 An SSCP is unable to reroute a session services request because a necessary routing table is not available, that is, there is no adjacent SSCP table corresponding to the rerouting key in the Resource Identifier control vector. The receiver of this value will, if possible, try rerouting to another SSCP.

0003 This SSCP has no predefinition for an LU, but an adjacent SSCP does not support dynamic definition in partner SSCPs. As a result, this SSCP cannot both dynamically define the LU and reroute to that adjacent SSCP.

0004 A conflict in gateway definition or capabilities has been detected during cross-network session establishment.

0005 Retired.

0006 Retired.

0008 The adjacent SSCP does not support the requested CDINIT function (for example, notification of resource availability or XRF).

000A An SSCP is unable to reroute a session services request because the request has been routed through the same SSCP twice.

000B The DLU specified in the CDINIT is unknown to the receiving SSCP, and the receiving SSCP cannot reroute the CDINIT.

087E SSCP Visit Count Exceeds Limit: The SSCP visit count specified in the session services request – CDINIT, INIT_OTHER_CD, or DSRLST – has been decremented to 0. The session services request has been routed through an excessive number of SSCPs. (The SSCPs are not necessarily distinct.)

0881 ACTCDRM Failure–REQACTCDRM Sent: An SSCP-SSCP session-activation request, ACTCDRM, cannot be rerouted to a gateway SSCP because, at some gateway PU, the necessary transform is not complete and the gateway PU has sent REQACTCDRM to the gateway SSCP.

0884 ACTCDRM Failure–No REQACTCDRM Sent: An SSCP-SSCP session activation request, ACTCDRM, cannot be rerouted to the destination SSCP because, at some gateway node PU, the necessary transform is not complete and REQACTCDRM cannot be sent to the destination SSCP because the gateway SSCP-PU session is not active or the intended SSCP session partner does not provide gateway services.

0885 Reserved

0886 Subnetwork Rerouting Not Supported: An SSCP received a session services request–CDINIT, INIT_OTHER_CD, NOTIFY(Vector Key=X'01'), or DSRLST–from an SSCP in its subnetwork that, if rerouted, would not cross a subnetwork boundary. The SSCP does not support rerouting within a subnetwork.

0887 Dequeue Retry Unsuccessful–Session Remains Queued: The SSCP cannot successfully honor a CDINIT(Dequeue) request. The request specifies "leave on queue if dequeue-retry is unsuccessful." The SSCP has left the queued session on its queue.

0888 Name Conflict: A name specified in an RU is unknown, or is known and does not have the required capabilities, or is a duplicate resource for the specified resource type. When a name conflict is detected, further name checking ceases; multiple name conflicts are not reported or detected.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The specified DLU real network name is known, but identifies a resource that is not LU-LU session capable.

0002 The specified DLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0003 The specified OLU real network name is known, but identifies a resource that is not LU-LU session capable.

0004 The specified OLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0005 Name translation was invalid; that is, a different LU name was returned with the same network ID as the original LU name.

0006 The specified DLU real network name is known, but is a duplicate resource.

0007 The specified DLU alias network name is known, but is a duplicate resource.

0008 The specified OLU real network name is known, but is a duplicate resource.

0009 The specified OLU alias network name is known, but is a duplicate resource.

000B A cross-network DLU name is defined as a shadow resource, but shadow resources are not supported for cross-network sessions.

000C Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

000D When processing a session initiation RU, an SSCP has found two different resource definitions for the OLU, one with the real OLU name and one with the alias OLU name.

000E When processing a session initiation RU, an SSCP has found two different resource definitions for the DLU, one with the real DLU name and one with the alias DLU name.

000F The specified DLU network name is defined as a generic resource. The session should be reinitiated using the name of an LU.

0889 Transaction Program Error: The transaction program has detected an error.

This sense code is sent only in FMH-7.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 Program Error—No Data Truncation: The transaction program *sending* data detected an error but did not truncate a logical record.

Program Error—Purging: The transaction program *receiving* data detected an error. All remaining information, if any, that the receiving program had not yet received, and that the sending program had sent prior to being notified of the error, is discarded.

0001 Program Error—Data Truncation: The transaction program *sending* data detected an error and truncated the logical record it was sending.

0100 Service Transaction Program Error—No Data Truncation: The service transaction program *sending* data detected an error and did not truncate a logical record.

Service Transaction Program Error—Purging: The service transaction program *receiving* data detected an error. All remaining information, if any, that the receiving service transaction program had not yet received, and that the sending service transaction program had sent prior to being notified of the error, is discarded.

0101 Service Transaction Program Error—Data Truncation: The service transaction program *sending* data detected an error and truncated the logical record it was sending.

088A Resource Unavailable—NOTIFY Forthcoming: The SSCP cannot satisfy the request because a required resource is temporarily unavailable. When the required resource becomes available, NOTIFY NS(s) key X'07' or X'08' will be sent.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 SSCP-SSCP Session Not Active: A SSCP-SSCP session required to reroute the cross-network request was not active.

0003 SSCP-LU session not active: The SSCP(DLU) is currently not in session with the DLU.

0004 LU session limit exceeded: The DLU is currently at its session limit and the requested session would cause the limit to be exceeded.

088B BB Not Accepted—BIS Reply Requested: Sent in response to a BB (either an LUSTAT bid or an Attach) to indicate that the receiver has sent a BIS request and wishes to terminate the session without processing any more conversations, but without sending an UNBIND. A BIS reply is requested so that the negative response sender may send a normal UNBIND. This sense code is sent only by LUs not supporting change-number-of-session (CNOS) protocols.

088C Missing Control Vector: The RU or XID did not contain a required control vector.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnyy Byte 2 contains the key (nn) of the required control vector that is missing and byte 3 (yy) contains the control vector's type, if appropriate.

088D Duplicate Network Name: An SSCP has detected a violation of the requirement that network names used across multiple domains be unique within the multiple-domain network. For example, the SSCP(DLU) has detected that the OLU name received in CDINIT is currently also defined in the domain of the SSCP(DLU).

088E Capability Mismatch: A network component detected a capability mismatch between different resources involved in the same network function. For example, an SSCP detects that an LU has been assigned a

subarea address too large for one of the other resources involved in the session initiation to support.

Bytes 2 and 3 following the sense code contains sense code specific information. Settings allowed are:

0000 A resource encountered during LU-LU session initiation is not ENA-capable; the session initiation request may be rerouted.

0001 A resource encountered during LU-LU session initiation is not ENA-capable; the session initiation request should not be rerouted.

0002 An SSCP has requested a "pre-ENA compatible" SLU address for an SLU that already has an ENA address.

0003 The gateway node selected by the gateway SSCP from the gateway node list is not ENA-capable when an ENA-capable gateway node is required. Another gateway node may be tried.

0004 During a dynamic path update, the SSCP detected that the update contained a path definition with an ER number greater than 7 and that the target node does not support extended subarea addresses. Therefore, the dynamic path update information for this destination subarea was not forwarded to the target node.

0005 The session could not be established because a specified extended subarea address exceeded that allowed at a node along the selected session setup path. The gateway SSCP doing gateway node selection may retry the session setup by selecting another gateway node having a larger subarea address limit in the network containing the DLU.

0006 The session could not be established because a specified extended subarea address exceeded that allowed at a node along the selected session setup path. The gateway SSCP doing gateway node selection may retry the session setup by selecting another gateway node that uses a smaller subarea address in the network containing the DLU.

0007 During a dynamic path update, the SSCP detected that the update contained a path definition with a subarea address above 255 and that the target node does not support extended subarea addresses. Therefore, the dynamic path update information for the destination subarea was not forwarded to the target node.

088F XRF Procedure Error: A request was received for an XRF-active or XRF-backup session and was not acted on.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0004 A SWITCH request was received that was invalid.

0006 An INITIATE request for an XRF-backup session was received that allowed queuing (XRF-backup and session queuing are mutually exclusive functions.)

0007 A CDINIT or INITIATE request was received specifying an XRF-backup session, and the DLU does not support XRF sessions.

0008 An XRF-active BIND was received with a session correlation identifier that duplicates a session correlation identifier associated with an existing XRF session.

0009 An XRF-backup BIND was received for an LU that currently does not have an XRF session.

000A Cryptography Not Supported: An XRF BIND was received indicating cryptography.

000B An INITIATE request was received specifying an XRF-backup session, and the OLU does not support XRF sessions. This is a definition mismatch between the OLU and the SSCP(OLU).

0010 An XRF-backup BIND was received with a session correlation identifier that does not match the session correlation identifier associated with the existing XRF session with that LU.

0012 An XRF-backup BIND associated with the existing XRF session supporting data compression was received that did not support compression.

0013 The existing session was negotiated using an extended BIND carrying the Length-Checked Compression (X'66') control vector, but the XRF-backup BIND is nonextended.

0890 Search Failure.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0010 Routing Error during a Directed Search: A Locate GDS variable for a directed search was received by an intermediate NNCP and could not be successfully routed to the destination control point.

0020 Resource Not Found during a Directed Search: A Locate GDS variable for a directed search was received by the named destination CP and the search argument resource is not a local resource.

0022 Destination of search not served by this CP.

0024 A search request or BIND was received from an unauthorized end node identifying an origin LU not represented in the network node server's directory, and thus could not be authenticated.

0030 Resource Deleted, No Broadcast Required: A Locate GDS variable for a directed search was received by the named destination CP and the search argument resource has been deleted.

0038 Too Many Directed Search Subprocedures: A Locate search exceeded the maximum height of the search tree; too many directed search subprocedures were tried; no retry.

0040 Resource Not Found during a Broadcast Search: A Locate GDS variable for a broadcast search was received by a CP that does not provide network services for the search argument resource and neither do any of the CPs searched in its broadcast subtree. This condition is detected by crossing search requests (a CP sends and receives a search request with the same FQPCID and the same search argument resource) or by a local search failure and all CPs in the broadcast subtree returning this sense data.

0048 Neutral Reply Received from an End Node: A Locate reply with no Found and no Extended Sense Data (X'35') control vector was received from an APPN end node.

0050 Quiesced CP: A CP in the broadcast search tree is in a quiescent state and, therefore, not receiving Locate GDS variables. This condition is detected when a CP in the search subtree is quiesced and no other CP in the subtree found the requested resource.

0060 Storage Not Available: A CP in the broadcast search tree does not have sufficient storage to participate in the search and no other CP in the search subtree found the requested resource.

0070 Session Outage: A CP in the search tree has lost its CP-CP session with a CP that had been sent a Locate GDS variable and no reply had been received.

0080 Duplicate Fully Qualified PCID: A CP in the search tree detected a duplicate fully qualified PCID for a different session request from the session request that first used the fully qualified PCID.

0081 PCID Modifier Too Long: A PCID Modifier List was received that had a length greater than 10 bytes.

0082 PCID Modifier Space Exhausted: A PCID Modifier List was received that contained the maximum of 10 bytes. As the maximum list size has been reached, another list entry cannot be made that was longer than 10 bytes.

0891 Invalid or Missing Network ID (NETID).

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0004 The Network Name control vector appended to the received XID3 does not contain a valid network ID. The network ID, preceding the CP name, must be greater than 0 and less than 8 bytes in length.

Alternatively, a network ID was received as an entry in a Register GDS variable without an accompanying resource name, resulting

in an invalid resource name at the receiver; the entry was not registered.

0005 The Network Name control vector appended to the received XID3 does not contain a valid CP name. The CP name, following the network ID, must be greater than 0 and less than 8 bytes in length.

0007 Invalid NETID: Establishment of a switched link connection failed because the NETID of the destination PU was not equal to that of the requesting SSCP.

0892 Automatic network shutdown (ANS) has occurred.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Session Reset After Loss of an SSCP: The SSCP controlling an LU has been lost. The session will be terminated because the T4 node, by system definition, terminates such sessions for this LU upon loss of the SSCP.

0002 The LU-LU session was in pending-active state when the SSCP failed. Although the T4 node, by system definition, continues an active LU-LU session upon the loss of the SSCP, the session was not completely set up, and thus it was reset.

0003 XRF-backup Session Reset. The XRF-backup session was reset because the T4 node resets the session upon loss of the SSCP.

0893 Takeover Not Complete

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 PLU Lacking a Control Point, Retry: The PLU is not currently receiving network services from a control point. The BIND is rejected because the session cannot be established. This sense data is returned by the boundary function of the PLU.

0002 SLU Lacking a Control Point, Retry: The SLU is not currently receiving network services from a control point. The BIND is rejected because the session cannot be established. This sense data is returned by the boundary function of the SLU.

0003 Sequence Error: The SSCP should not send an RNAA for an independent LU until the takeover sequence is complete for the link station, that is, until all BFSESSINFOs for that LU have been received and accepted.

0894 Migration Support Error: The sender of the request is relying on migration support that is not available.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 BIND cannot be extended: A BIND that is not an LU6.2 BIND was received and cannot be extended by the receiver.

0895 Control Vector Error: The RU or XID contained a control vector that was in error.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

xxyy Byte 2 (xx) contains the key of the control vector first detected in error. If more than one control vector is in error, only the first erroneous one is reported. Byte 3 (yy) of the sense code specific data contains the (0-origin) byte offset of the error within the control vector.

0896 Control vector too long.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Network Name (X'0E') control vector is too long; the vector data portion is greater than 18 bytes long.

0897 System Definition Mismatch: The requested function is not supported by the receiver, or there is a mismatch between the sending and receiving system definitions.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0006 The SSCP has no predefinition for an LU and does not support dynamic resource definition.

0007 The receiving SSCP has a system-defined name for the SSCP(DLU) that differs from the SSCP(DLU) name in the session initiation request.

0008 In a gateway with three gateway SSCPs, a gateway SSCP on the OLU side of the gateway was specified as having predesignated control in the CDINIT. In this configuration, only the middle gateway SSCP may have predesignated control.

0009 In a gateway with three gateway SSCPs, none of which is predesignated, the gateway node believes that one is predesignated. As a result, the gateway node receives gateway control RUs such as RNAA from an unexpected SSCP.

000B An SSCP has detected a specification of gateway responsibility in the CDINIT request that is not consistent with its own definition. For example, two gateway SSCPs in the same gateway are both predefined to be predesignated.

000C The receiver is unable to interpret the DLU name.

000D Resource type not defined in receiver.

000E Reserved

000F Reserved

0010 An adjacent SSCP has the same SSCP name as the SSCP that controls the DLU, but a different network identifier from the DLU.

0012 The receiving SSCP has a system-defined name for the SSCP(OLU) that differs from the SSCP(OLU) name in the session initiation request.

0013 A CDINIT was received that indicated that the receiving SSCP controls the OLU.

0014 The receiving T4 node (though capable of supporting the function) was not defined by local system-definition option to support the requested dump type.

0015 The OLU is represented using a dynamically defined resource but the ALS selected to provide its services does not permit dynamic definitions. The condition is detected when a session initiation request is received for an independent LU and no predefinition is found for the OLU resource. The session initiation is rejected.

0016 The DLU is represented using a dynamically defined resource but the ALS selected to provide its services does not permit dynamic definitions. The condition is detected when a session initiation request is being processed for an independent destination LU and no predefinition is found for the DLU resource. The session initiation request is rejected.

0017 The request was received for an independent LU over a specific ALS but that ALS is not defined to provide services for the subject LU. The condition is detected when a session initiation request is recieved and the ALS for which the request was received was not predefined to provide service for that independent LU. The session initiation request is rejected.

0018 Session Initiation Status Not Supported: A session initiation request was received that contained a session initiation status field invalid for the receiving node.

0898 Session Reset: The XRF session is being reset.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The XRF-active session has been reset because the XRF-backup PLU forced a takeover.

0002 XRF-backup Hierarchical Reset: The identified XRF-backup LU-LU session is being deactivated because the related XRF-active session terminated normally. The LU sending this sense data is resetting its half-session before receiving the response from the partner LU. (See UNBIND type X'12'.)

0003 XRF-active Hierarchical Reset: The identified XRF-active LU-LU session is being deactivated because the related XRF-backup session performed a forced takeover of this session (via SWITCH). The LU sending this sense data is resetting its half-session before receiving the response from the partner LU. (See UNBIND type X'13'.)

089A Invalid File or File Not Found: The requested file was not found, or was found to be an invalid file.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Requested file not found.

0002 The specified load module already exists and, therefore, cannot be added.

089B Session Correlation Exception: The session correlation procedure detected an exceptional condition at the SLU.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 RUs Out of Order: A BIND request with the correlating Fully Qualified PCID (X'5F') control vector arrived before UNBIND(Type X'02') was received for the correlated session. This sense data is sent in an UNBIND that terminates the correlated session.

0002 Correlator Not Found: A BIND request with the correlating Fully Qualified PCID (X'5F') control vector cannot be correlated with any previous session.

089C Duplicate Session-Related Identifier.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Invalid URC: The URC received in a BFINIT duplicates a URC for an outstanding session initiation attempt from the same BF.

089D Gateway Node Error Detected during Cross-Network Session Initiation.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The gateway node list used to select a gateway node to cross a network boundary is exhausted.

0003 RNAA has failed; another gateway node should be tried.

0004 Address conversion based on the subarea/element address split was unsuccessful.

0005 The gateway node selected by one gateway SSCP is not known to another gateway SSCP in the same gateway. This can be a system definition error in the gateway SSCP that does not recognize the gateway node.

0006 A gateway SSCP has found that a gateway node has assigned duplicate addresses.

089E Identified Data Object Already Exists.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 A request to create a new data object has failed because the identified data-object already exists at the target node.

0002 A request to replace a data object has failed because it specifies a to-be-deleted data object different from the to-be-stored data object; however, the to-be-stored data object already exists.

08A0 Session Reset: An LU or PU is resetting an LU-LU session.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0004 Reversed FRSN Values: The value in the Last FRSN Sent field is greater than the value in the Current FRSN field in a received TDU GDS variable (no retry). The CP will send an UNBIND with a reason code of X'0F' (Cleanup)

0005 TDU Sent Out of Order: The value in the Last FRSN Sent field of the current TDU GDS variable is less than the value of the Current FRSN field in the TDU GDS variable that immediately preceded it, or is greater than it and the receiver cannot store the out-of-sequence value (no retry). The CP will send an UNBIND with a reason code of X'0F' (Cleanup)

08A2 Resource Active. The requested function must be performed on an inactive resource, and the resource is active.

08A4 Token-Match Exception: Partial name matching is unsuccessful during the required find or store operation. The canonical identifier involved in the exception is reported in the FS server report.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 One or more must-match tokens were not specified. When this report code is used in an SNA condition report, it is accompanied by a structure report containing the token-match indicators, as specified in the request plus a supplemental report containing the token attributes, as they appear in the report's directory.

0002 Specified token-match indicators yield multiple directory matches. When this report code is used in an SNA condition report, it is accompanied by a structure report containing the token-match indicators, as specified in the request plus a supplemental report containing the token attributes, as they appear in the report's directory.

08A6 Object Not Found: An exception has occurred when the general server attempted to process the server object, but the server object could not be found.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0001 Server object not found.

08A8 Multiple-Domain Support Routing Exception: The MDS router in the reporting NAU is unable to perform the required routing for an MDS-MU.

When this SNA report code is used in an SNA condition report (X'1532') GDS variable, the destination NAU name is included in the Reported on Location Name (X'09') subvector and the destination MS application name is included in the Reported On Agent (X'04') subvector of the condition report.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Destination NAU name unknown. Directory services could not locate the requested destination name.

0002 Directory services unavailable. No routing possible.

0003 MS application program name not recognized.

0004 Use of CPSVCMG session not permitted. The reporting network node has received an MDS-MU over a CPSVCMG session from another network node. These sessions are used for MDS-MUs only between a network node and its served end nodes.

0005 Function not supported by EN destination. The back-level end node destination does not support receipt of MS messages (reported by serving network node).

0006 Function not supported by destination. The back-level destination does not support receipt of MS messages other than MS Capabilities and Alert

0007 Function not supported by serving NN. The serving network node of the end node destination does not support routing of MS messages (reported by network node performing routing).

0008 Function not supported by EN. The reporting end node has received an MDS-MU with a destination other than itself.

0009 Destination not supported by reporting NN. A network node has received an MDS-MU from another network node that cannot be routed. The destination is not the reporting network node itself nor is it one of the served end nodes.

000A Unrecoverable session failure. The MDS_SEND TP in the reporting node was unable to send the message because of an allocation error. Retries have been exhausted.

000B Unrecoverable TP failure in remote node. The MDS_SEND TP in the reporting node was unable to send the message because of a TP failure in a remote node. Retries have been exhausted.

000C MS Application program failure. The MDS router in the destination NAU is unable to communicate with the destination MS application program.

000D Unrecoverable TP failure in reporting node. The MDS router in the reporting node was unable to send the message because of a local TP failure.

000E Correlation error. An MDS-MU has been received that is not the first for a unit of work (First MDS Message indicator in the MDS Routing Information Message is 0), but the agent unit of work correlator is unknown (does not match any active MDS transaction). Also used to report the receipt of a duplicate correlator (MDS-MU with first MDS message indicator is 1, but the agent unit of work correlator matches one currently in use).

000F MS application program congestion. The MDS router in the destination NAU is unable to communicate with the destination MS application program because of local congestion (implementation buffer space for queuing additional MDS-MUs has been exhausted).

0013 Session UNBIND notification. The last session to the indicated destination has been deactivated. Refer to product documentation for additional information.

08A9 Multiple-Domain Support Transaction Failure: The reporting MDS router or MS application program has detected a condition that has impacted an outstanding unit of work (identified by the agent unit of work correlator of the MDS error message).

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Failure caused by outage of a CPSVCMG session.

0002 Failure caused by outage of an SNASVCMG session. All retries have been exhausted.

0003 Unit of work canceled by reporting MS application program. The unit of work has been canceled because of a timeout in the reporting MS application program.

0004 Unit of work canceled by reporting MDS Router. The unit of work has been canceled by a garbage-collection timeout in the reporting MDS router.

0005 MDS router internal failure. The unit of work has been canceled because of an internal failure in the reporting MDS router.

0006 MS Application internal error. The unit of work has been canceled either because the reporting MS application program was terminated or because another application program served by it was terminated. The type of program termination (normal or abnormal) is not indicated.

0007 MS Application router re-initialization. The unit of work has been canceled by the reporting MDS router because of a re-initialization of the application-level router.

08AA Required GDS Variable Missing: The MS Multiple-Domain Support Message Unit (MDS-MU) is missing a required GDS variable.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nnnn Bytes 2 and 3 following the sense code contain the ID of the missing GDS variable.

08B2 Data Transmission Failure: The data transmission between an application program in an SNA MS entry point and an application program in a subentry point was incomplete, causing abnormal termination of the function.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 A timeout has occurred while waiting for transmission of data between the two application programs. For example, a service processor has timed out while waiting to receive data from the main processor.

08B3 DS DTMU Build Exception: Building of the DS Distribution Transport MU was unsuccessful.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 A maximum-sized agent object in the MU being built is insufficient to contain all the data required, and segmented agent objects are not supported. The MU is built to include as much data in the agent object as possible.

08B4 CP-CP Sessions Not Required: Sent from one APPN control point to another to deactivate CP-CP sessions between them, because the sender does not currently need CP-CP sessions, and the TG carrying the sessions is a limited resource.

Note: This sense data value is carried within the X'35' control vector on an UNBIND(Type = X'01').

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

08B5 Network Node Server Not Required: Sent by an APPN end node control point to a network node control point (1) to deactivate CP-CP sessions with the NNCP, or (2) to reject a CP-CP session BIND from the NNCP. The end node no longer requires network node services from the receiver.

Note: This sense data value is carried within the X'35' control vector on an UNBIND(Type = X'01') for case (1) above, or on an UNBIND(Type = X'FE') for case (2).

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

08B6 CP-CP Sessions Not Supported: Sent by a network node control point to reject a CP-CP session BIND from another APPN control point; support for CP-CP sessions on that TG was removed since the time when the TG was first activated.

Note: This sense data value is carried within the X'35' control vector on an UNBIND(Type = X'01').

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

Request Error (Category Code = X'10')

This category indicates that the request was delivered to the intended NAU component, but could not be interpreted or processed. This condition represents a mismatch of NAU capabilities.

Category and modifier (in hexadecimal):

1001 RU Data Error: Data in the request RU is not acceptable to the receiving component; for example, a character code is not in the set supported, a formatted data field is not acceptable to presentation services,

or a value specified in the length field (LL) of a structured field is invalid.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The request contains a subarea address of 0 or a subarea address greater than the maximum subarea value within the specified or implied network.

0002 The network ID specified in the ACTPU is unknown, or is not valid on the link over which the ACTPU was received.

0003 Isolated Pacing Message (IPM) Format Error: An incorrectly formatted IPM was received.

0008 An invalid character code was found.

0009 The formatted data field is unacceptable to presentation services.

000A An invalid length field for a structured field was found.

000B The value in the name length field is too great.

000C The value in the cryptography key length field is too great.

000D The URC field length is invalid.

000E The control vector length field is inconsistent with the control vector data.

000F A PLU or SLU role specification encoding is invalid.

0020 Too many session keys are present.

0021 A control vector or session key data is invalid.

0022 A BIND image in a session services RU is invalid.

0023 A device characteristics field is invalid.

hnnn where h≥8, i.e., the high-order bit in byte 2 is set to 1. The 15 low-order bits of bytes 2 and 3 contain a binary count that indexes (0-origin) the first byte of the field found to be in error.

1002 RU Length Error: The request RU was too long or too short.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

1003 Function Not Supported: The function requested is not supported. The function may have been specified by a formatted request code, a field in an RU, or a control character.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The half-session receiving the request did not perform the function because it is not capable of doing so. The requesting half-session requested a function that the receiver does not support and the receiver did not specify that it was capable of supporting the function at session activation; consequently, there is an apparent mismatch of half-session capabilities.

Note: This is to cover a system error. For example, if the PU receiving a SETCV(Vector Key=X'15') is not a gateway PU, that is, the PU did not indicate in the ACTPU response that it is a gateway PU, the PU reports to the SSCP that sent the SETCV that there is an apparent mismatch of half-session capabilities.

0002 The half-session receiving the request did not perform the function, though it is capable of doing so. The requesting half-session did not specify at session activation that it was capable of supporting the function; consequently, there is an apparent mismatch of half-session capabilities.

Note: This is to cover a system error. For example, if the SSCP sending a SETCV(Vector Key=X'15') is not known to the receiving PU as a gateway SSCP, that is, the SSCP did not indicate in ACTPU that it is a gateway SSCP, the PU reports a mismatch of capabilities.

Note: 0001 and 0002 are also assigned for implementation-specific use; see implementation documentation for details of usage.

0003 The component received an unsupported normal-flow DFC command.

0004 The component received an unsupported expedited-flow DFC command.

0005 The component received a network control command during an LU-SSCP session.

0006 The component received an unsupported session control command during an LU-SSCP session.

0007 The component received an unsupported data flow control command with LU-SSCP session specified.

000D The function identified in the request is not supported by the processing application transaction program.

0010 The RU is not known to session services.

0011 A session key is not supported.

0012 A control vector is not supported.

0014 Cryptography is not supported but a nonzero length was specified for the cryptography key.

0015 Queuing not supported for a controller session.

0016 Service parameter not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report identifying the service parameter triplet (or triplets) that was not supported.

0017 Service parameter level not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report identifying the service parameter triplet (or triplets) that was not supported.

0018 Destination-role function not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying the structure and containing the contents that specified the unsupported function. Whenever the structure report is not sufficient to identify the unsupported functions, the supplemental report may also be present.

0019 All-role function not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying the structure and containing the contents that specified the unsupported function. Whenever the structure report is not sufficient to identify the unsupported functions, the supplemental report may also be present.

001A Reserved.

001B Unable to initiate Agent.

001C Function conflicts with Format Set 1 encodings. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying the structure and containing the contents that specified the conflicting function.

001D Reserved

001E Reserved

001F Multiple-destination traffic not supported. The reporting location is a specialized, end-only role implementation that supports single-destination traffic only.

0020 A session initiation request specified an OLU and DLU that are the same LU. A dependent LU cannot establish a session with itself.

0027 LU type not supported.

0028 Nonnegotiable BIND not supported by the receiver.

0029 Transmission service (TS) profile of BIND not supported by the receiver.

0030 Normal-flow send/receive mode conflicts with the mode specified in the transmission services (TS) profile of a received BIND.

0031 The primary LU cannot support being first speaker; secondary LU must be first speaker.

0032 In BIND, the specified bracket termination is not supported by the receiver.

0033 Definite response mode is not supported by the receiver.

0034 Secondary LU cannot send EB when normal-flow send/receive mode is full-duplex.

0035 Bracket error resulting from failure of sender to enforce bracket rules for session.

6002 The resource identified by the destination program name (DPN) is not supported.

6003 The resource identified by the primary resource name (PRN) is not supported.

Note: This sense code can also be used instead of sense code X'0826'.

1004 Reserved

1005 Parameter Error: A parameter modifying a control function is invalid, or outside the range allowed by the receiver.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 For NMVT, the address type field in an SNA Address List subvector does not match the address type required by the command subvector.

0002 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0004 Invalid display type was requested.

0005 Invalid storage length for display type requested.

0006 Invalid storage address; out of specified range.

0007 The command in a Request Change Control MS major vector is incompatible with the SNA/FS server instruction.

0008 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0010 Incorrect setting of backup focal point flag. The MS Capabilities (X'80F0') major vector received from the focal point contains a backup focal point flag with a value of 1 (indicating that the entry point is to keep its current focal point), but the Focal Point Identification (X'21') subvector in the same major vector names a new backup focal point.

0121 – 0229 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

1006 Required field or parameter is missing.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 One or more required COS names were omitted.

0002 A required name was omitted.

0003 A required network identifier was omitted.

0004 A required session key was omitted.

0005 A required control vector was omitted.

0006 A required subfield of a control vector was omitted.

0007 The TG number field was omitted.

0008 The system-defined ID number, used within the Node Identification field of an XID, was omitted.

0009 A required GDS variable is missing.

1007 Category Not Supported: DFC, SC, NC, or FMD request was received by a half-session not supporting any requests in that category; or an NS request byte 0 was not set to a defined value, or byte 1 was not set to an NS category supported by the receiver.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Invalid NS header received. An NS request byte 0 was not set to a defined value.

1008 Invalid FM Header: The FM header was not understood or translatable by the receiver, or an FM header was expected but not present. For LU 6.2, this sense code is sent in FMH-7 or UNBIND.

Bytes 2 and 3 following the sense code contain sense code specific information. Figure 10-2 on page 10-70 shows the usage of the allowed values by LU type. Settings allowed are:

0000 No specific code applies.

0801 The function code parameters are invalid.

0803 The forms functions cannot be performed.

0805 The copy function cannot be performed.

0806 Compaction table outside the supported set: The number of master characters is not within the valid range.

0807 The PDIR (peripheral data information record) identifier is invalid.

0808 The printer train function cannot be performed.

0809	The FCB (forms control block) load function cannot be performed.
080A	The FCB (forms control block) load function is not supported.
080B	The compaction table name is invalid.
080C	The ACCESS is invalid.
080D	The RECLEN is invalid.
080E	The NUMRECS is invalid.
080F	The data set is in use.
0810	The data set cannot be found.
0811	The password is invalid.
0812	The function is not allowed for the destination or for the data set.
0813	The record is too long.
0814	The data set is full.
0815	The RECID is invalid.
0816	Reserved
0817	The VOLID format is invalid.
0818	The maximum number of logical records per chain is exceeded.
0819	The data set exists.
081A	No space is available.
081B	The VOLID is invalid.
081C	The DSACCESS is invalid.
081D	The RECTYPE is invalid *or* the data set cannot be found.
081E	The resolution space is insufficient.
081F	The key technique is invalid.
0820	The key displacement is invalid.
0821	The key is invalid.
0822	There is an Invalid N (number of records.)
0823	The KEYIND is invalid.
0824	The SERID is invalid.
0825	Disk Error: An error was detected while reading from, or writing on, the disk.
0826	The RECID format is invalid.
0827	The password has not been supplied.

0828	The record ID has not been supplied.
0829	The Volume ID has not been supplied.
082A	The PGMNAME is invalid.
1204	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
2001	The destination (active) is invalid.
2002	The destination (inactive) is invalid.
2003	The destination (suspended) is invalid.
2004	The suspend-resume sequence is invalid.
2005	There has been an interruption level violation.
2006	The resume properties are invalid.
2007	The destination is not available.
2008	The end sequence is invalid.
2009	The FM header length is invalid.
200A	Invalid field setting: The reserved field is set to 1 or the setting is not defined.
200B	Invalid destination: The destination does not exist.
200C	The ERCL is invalid.
200D	The DST is invalid.
200E	Invalid Concatenation Indicator: The concatenation indicator is *on*, but concatenation is not allowed.
200F	FM data is not allowed for the header.
2010	The FM header set specified in the BIND has been violated.
2011 – 2013	Reserved
2014	The FM header was not sent concatenated.
2015 – 2018	Reserved
2019	The stack reference indicator (SRI) is invalid.
201A	The CMI modification could not be accepted.
201B	The CPI modification could not be accepted.
201C	The ECRL modification could not be accepted.
201D	FM Header and Associated Data Mismatch: The FM header indicated associated data would or would not follow (for example, FM header 7 followed by log data, or FM header 5 followed by program initialization parameters), but this indication was in error; or a previously

	received RU (for example, -RSP(X'0846')) implied that an FM header would follow, but none was received.
4001	Invalid FM Header Type for this LU: The type of the FM header is other than 5, 7, or 12.
4002	The FMH code is invalid.
4003	Compression is not supported.
4004	Compaction is not supported.
4005	Basic exchange is not supported.
4006	Only basic exchange is supported.
4007	The medium is not supported.
4008	There has been a code selection compression violation.
4009	FMHC is not supported.
400A	Demand select is not supported.
400B	DSNAME is not supported.
400C	The media subaddress field is invalid.
400D	There are insufficient resources to perform the requested function.
400E	DSP select is not supported.
6000	FM Header Length Not Correct: The value in the FM header Length field differs from the sum of the lengths of the subfields of the FM header.
6001	The deblocking algorithm (DBA) is invalid.
6004	The queue name length is invalid.
6005	Access Security Information Length Field Not Correct: The value in the Access Security Information Length field differs from the sum of the lengths of the Access Security Information subfields.
6006	The data stream profile (DSP) is invalid.
6007	The FMH-7 is not preceded by a negative response carrying the X'0846' sense code.
6008	The Attach access code is invalid.
6009	Invalid Parameter Length: The field that specifies the length of fixed-length parameters has an invalid setting.
600A	This is not the first FMH-5, the interchange unit type is not the same as the old, and the interchange unit end indicator is *off*.

600B	Unrecogized FM Header Command Code: The partner LU received an FM header command code that it does not recognize. For LU 6.2 this sense data is sent only in FMH-7.
600C	A null sequence field is required.
600D	User to user program transition is not allowed.
600E	User to non-SNA defined program transition is not allowed.
600F	The FMH-5 reset attached program (RAP) was not sent properly.
6010	The FMH-5 reset attached program (RAP) was sent with an inactive Attach register.
6011	Invalid Logical Unit of Work (LUW): The LUW Length field (in a Compare States GDS variable or an FMH-5) is incorrect, or the length field is invalid, or a LUW ID is not present but is required by the setting of the synchronization level field.
6021	Transaction Program Name Not Recognized: The FMH-5 Attach command specifies a transaction program name that the receiver does not recognize. This sense data is sent only in FMH-7.
6031	PIP Not Allowed: The FMH-5 Attach command specifies program initialization parameter (PIP) data is present, but the receiver does not support PIP data for the specified transaction program. This sense data is sent only in FMH-7.
6032	PIP Not Specified Correctly: The FMH-5 Attach command specifies a transaction program name that requires program initialization parameter (PIP) data, and either the FMH-5 specifies PIP data is not present or the number of PIP subfields present does not agree with the number required for the program. This sense data is sent only in FMH-7.
6034	Conversation Type Mismatch: The FMH-5 Attach command specifies a conversation type that the receiver does not support for the specified transaction program. This sense data is sent only in FMH-7.
6040	Invalid Attach Parameter: A parameter in the FMH-5 Attach command conflicts with the statement of LU capability previously provided in the BIND negotiation.
6041	Synchronization Level Not Supported: The FMH-5 Attach command specifies a synchronization level that the receiver does not support for the specified transaction program. This sense data is sent only in FMH-7.

6042 Reconnection Not Supported: The FMH-5 Attach command specifies reconnection support but the receiver does not support reconnection for the specified transaction program. This sense data is sent only in FMH-7.

6043 Unable to Reconnect Transaction Program—No Retry: The FMH-5 Reconnect command specifies the conversation correlator of a transaction program to which the receiver cannot reconnect. The condition is not temporary. This sense data is sent only in FMH-7.

6044 Unable to Reconnect Transaction Program—Retry Allowed: The FMH-5 Reconnect command specifies the conversation correlator of a transaction program to which the receiver cannot reconnect. The condition is temporary. This sense data is sent only in FMH-7.

6045 Reserved

6046 An SNA/DS transaction program is unable to allocate a conversation with a SNA/DS partner.

6047 An SNA/DS transaction program in conversation with an adjacent SNA/DS transaction program has detected from LU 6.2 PS a return code of RESOURCE_FAILURE.

6048 An SNA/DS transaction program in conversation with an adjacent SNA/DS transaction program has detected from LU 6.2 PS a return code of DEALLOCATE Type(Abend).

C000 The header is not supported.

C001 The header length is invalid.

C002 There has been a logical message services block-level error.

C003 There is a version ID mismatch.

1009 Format Group Not Selected: No format group was selected before issuing a Present Absolute or Present Relative Format structured field to a display.

100A Unknown User Name.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0001 The specified user name (e.g., origin, destination, or report-to) cannot be identified with an entry in the directory.

100B Format Exception

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

Range	LU 1	LU 4	LU 6.1	LU 6.2
0801-0824	X	X		
0825	X			
0826-082A	X	X		
2001-200D	X	X		
200E	X	X	X	
200F-201C	X	X		
201D				X
4001-400E	X	X		
6000				X
6001,6004			X	
6005			X	X
6006-6008			X	
6009			X	X
600A			X	
600B			X	X
600C-6010			X	
6011-6034				X
6040			X	X
6041				X
6046				X
6047				X
6048				X
C000-C003			X	

Figure 10-2. Usage of X'1008' Sense Code Specific Information by LU Type

0001 Required structure absent. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the absent structure.

0002 Precluded structure present. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the precluded structure.

0003 Multiple occurrences of a nonrepeatable structure. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies and contains the second occurrence of the structure.

0004 Excess occurrences of a repeatable structure. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies and contains the occurrence of the structure that exceeded the maximum, plus a supplemental report that contains the allowed maximum number of occurrences.

0005 Unrecognized structure present where precluded. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies and contains the precluded unrecognized structure, plus a sibling list of all the allowed structures.

0006 Length outside specified range. This code assumes that the length arithmetic balances and that the sender intended to send the structure at that length. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies and contains the header of the excessively long structure, plus a supplemental report that contains the allowed maximum length.

0007 Length exception. Length arithmetic is out of balance. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies and contains the header of the structure that exceeded its parent's boundary.

0008 Required combination of structures absent. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures that make up the combination, indicating for each whether it was present or absent.

0009 Precluded combination of structures present. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures that make up the precluded combination.

000A Required combination of structures and data values absent. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures and data values that are present, plus structure reports that identify the absent structures needed to complete the combination.

000B Precluded combination of structures and data values present. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures and data values that make up the precluded combination.

000C Unknown or unsupported data value. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure and contains the unknown or unsupported data value.

000D Incompatible data values. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures and the incompatible data values.

000E Precluded character present. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure, indicates the byte offset of the offending byte, and includes the byte containing the precluded code point.

000F Data-value out of range. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure and contains the offending data value, plus a supplemental report that contains the maximum value allowed within the range (if a maximum range value is applicable).

0010 Segmentation present where precluded. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure that should not have been segmented.

0011 Precluded data value. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure and contains the offending data value.

0012 Recognized but unsupported structure. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure.

0013 None of several possible structures found. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the parent of the absent structure and may contain an unrecognized structure that was found in the place of the absent structure. The structure report also contains a sibling list of the possible structures.

0014 Incorrect order of child structures found. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the parent of the incorrectly ordered child structures.

100C Unrecognized Message Unit

Bytes 2 and 3 following the sense code contain sense code specific information. Specific settings allowed are:

0001 The received byte stream could not be identified by the receiving SNA component. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying and containing the unrecognized message unit, plus a sibling list of the allowed message units.

100D Request Inconsistency: The control information provided for the request is not consistent with other information in the request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Server object size is incompatible with service level. When this SNA report code is used in an SNA condition report, it is accompanied by one structure report containing the capacity service parameter triplet and one supplemental report containing the server object size.

100E Directing Exception: A node is unable to perform the required directing or redirecting function for a request as a result of insufficient directory support, or incompatibility between TP name and presence/absence of a user name.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Agent name known but not supported for specified user destination.

0002 Agent name known but not supported for specified node destination.

0003 Agent name is known at this DSU but not available.

100F Improper SNA/DS Usage of LU 6.2.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0001 An SNA/DS transaction program in conversation with an adjacent SNA/DS transaction program has detected an improper sequence of LU 6.2 basic conversation verbs.

1010 Error on Locate Search or CP Capabilities Message Detected.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 Unrecoverable error, such as a duplicate control vector, was detected.

0001 A broadcast search resulted in two or more conflicting positive replies that differ on the CP owning the target resource. Multiple positive replies are acceptable, as long as all indicate the same owning CP.

0004 Unrecoverable error on CP Capabilities GDS variable exchange prevented its initiation or completion on a contention-winner CP-CP session.

1000 Length error in CP Capabilities GDS variable.

1002 Invalid GDS variable received when CP Capabilities GDS variable was expected.

4004 Incomplete negative or neutral reply received on a search , or reservation indicated on Broadcast, or "All" specified on a directed search.

5000 Length error in CD-Initiate GDS variable.

5002 No CD-Initiate GDS variable returned on a search request.

5006 Session polarity or initiate type value received in CD-Initiate GDS variable not supported.

500A Mode name length error in CD-Initiate GDS variable.

A002 Find GDS variable not present on Locate search request.

B080 Command Parameters (X'80') control vector not present on Found GDS variable.

1012 SNA/DS Receiver Exception MU Format Exception: Parsing or building of the SNA/DS Receiver-Exception MU Format was unsuccessful.

1013 Unknown Server Parameters: The specified parameters are not recognized by the server.

1014 Control Vector Error on a Directory Services GDS Variable.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

003C Missing Associated Resource Entry (X'3C') control vector on Find or Found.

003D Missing Directory Entry (X'3D') control vector on Find or Found.

0080 Invalid control vector.

023C Conflicting directory entry or invalid Associated Resource Entry (X'3C') control vector.

502B No RSCV received from a network node server.

502C No COS/TPF control vector received in a CD-Initiate reply from a network node server.

5046 TG vectors not present in a CD-Initiate from an end node OLU or DLU.

A080 Missing Command Parameters (X'80') control vector on Find.

A082 Missing Search Argument Directory Entry (X'82') control vector on Find.

B280 A Found from an end node indicated the directory entry for a located resource was a wild-card entry.

1015 XID Length Error: The XID3 was too long or too short.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The received XID3 has fewer than 29 bytes.

0002 There is a mismatch between the number of bytes specified in the Length field of XID3 and the actual length of the received XID3.

1016 XID Format 3 Parameter Error: Data in the XID3 is not acceptable to the receiving component because the value in the received XID3 field, whose byte and bit offset is specified by the XID Negotiation Error (X'22') control vector (which also carries this sense data), is inconsistent with the corresponding field in the sent XID3.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The field in the received XID3 that specifies the maximum number of I-frames that the sender can receive before acknowledgment is set to 0.

0002 The adjacent node has been inconsistent in its request for ACTPU. In a nonactivation XID3 exchange, it has changed the value of the ACTPU Suppression indicator sent in the previous XID3 exchange.

0003 The field in the received XID3 that specifies the maximum BTU length that the sender can receive is set to less than 99 bytes, the minimum required.

0004 The received XID was not XID format 3 when XID format 3 was expected.

0005 The adjacent node does not support BIND segment generation but does support receipt of BIND segments. Any T2.1 node supporting receipt of BIND segments must also support generation of BIND segments.

0006 The adjacent node is an end node, does not support BIND segment receipt, and has a maximum BTU size of less than 265, the minimum required in this case.

0007 The adjacent node is a network node, does not support BIND segment receipt, and has a maximum BTU size of less than 521, the minimum size required in this case.

0008 The adjacent node has changed its networking capabilities in an XID3 from those declared in the previous negotiation-proceeding or nonactivation XID3. A node may not change from an end node to a network node or *vice versa* in two different negotiation-proceeding or nonactivation XID3s.

0009 The adjacent node is an APPN network node, does not provide CP services, and supports CP-CP sessions, a combination not allowed.

000A During a nonactivation XID3 exchange, the adjacent node has changed the TG number that was negotiated during the activation exchange.

000B The adjacent node is the TG number negotiation winner and designates a TG number that the receiving node cannot allocate to this connection. When parallel TGs are supported between the two nodes, 0 is always such a number.

000C The adjacent node is an APPN network node that does not support BIND segment generation, and this node has a maximum BTU receive size of less than 521. This node may, therefore, be unable to receive a BIND with RSCV from the adjacent network node.

000D The adjacent node indicates that it does not support the SDLC command/response profile in its XID3. This is the only command/response profile supported by APPN and LEN nodes.

000E Different product set IDs have been given in the Product Set ID (X'10') control vectors appended to two different received XID3s from the same adjacent node.

000F The link station roles specified in the sent and received negotiation-proceeding XID3s are not compatible. To activate a connection, one node must contain a primary link station; the other, a secondary link station.

0010 The support of combined asynchronous balanced mode link stations indicated in the sent and received negotiation-proceeding XID3s is not in agreement.

0011 A received XID3 indicates an attempt to activate multiple connections has been made when parallel transmission groups are not supported between the two nodes involved in the XID exchange.

0012 The adjacent node has sent the Network Name (X'0E', CP name) control vector in XID3 but indicates it does not support the Exchange State indicators.

0013 The DLC type indicated in the sent and received negotiation-proceeding XID3s is not in agreement.

0014 After sending a negotiation-proceeding XID3 with the Link Station Role field set to either "primary" or "secondary," the adjacent node sends a negotiation-proceeding XID3 with this field set to "negotiable."

0015 During a negotiation-proceeding XID3 exchange, the adjacent node indicated that it supports BIND pacing as a sender but not as a receiver; i.e., byte 10, bits 0−1 in XID3 are set to 10, which is not allowed.

0016 The node receiving the XID3 is attempting to activate a predefined TG, i.e., a TG that will be assigned a number in the range of 1 to 20, but the TG number sent in the adjacent node's XID3 does not agree with the number that the receiving node has assigned to the TG.

0017 After two negotiation-proceeding XID3 exchanges, the Node Identification field of the received and sent XID3s have identical values. When both nodes contain negotiable link stations, link station roles must be resolved within two exchanges of XIDs after link station role negotiation has begun.

0018 The adjacent node is an APPN node but does not support adaptive BIND pacing as a sender and receiver.

0019 The receiving node does not support CP name changes on APPN connections, but has received a nonactivation XID3 with a CP name that differs from that received during the previous XID exchange.

001A The adjacent node is inconsistent in its support of parallel TGs. Support of parallel TGs between two nodes cannot change either in link-activation XID exchanges on different TGs or in successive XID exchanges on the same TG.

001B The adjacent node provides or requests CP services but does not support CP-CP sessions; i.e., bytes 8−9, bits 10−11 of the received negotiation-proceeding XID3 were set to 10, a setting combination not allowed for T2.1 nodes.

001C The adjacent node declares that its link station role is not primary, secondary, or negotiable; i.e., byte 19, bits 2−3 of the received negotiation-proceeding XID3 were set to 10, a value not allowed for T2.1 nodes.

001D The adjacent node supports two-way alternating as its transmit-receive capability while the receiving node supports two-way simultaneous and cannot negotiate down to a two-way alternating transmit-receive capability.

001E The adjacent node has not appended its CP name in the Network Name (X'0E', CP name) control vector on XID3, but indicates that it supports CP-CP sessions and requests them from the receiver. Such a node is interpreted as a LEN end node that is requesting APPN function, which is not permitted.

001F The setting of the Intersubnet Link bit of the TG Descriptor control vector received in XID3 is inconsistent with the receiving node's system definition. This sense data value is issued only if both sender and receiver support the setting of this bit.

0020 The node type the adjacent node declares itself to be in its XID3 is one to which the receiving node cannot activate a TG.

1018 MU Sequence Exception: An SNA/DS transaction program has detected an improper sequence of SNA/DS MUs.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0001 A DMU has been received, but the MU ID has already been terminated.

0002 The MU ID state received from the partner is incompatible with the state in the MU ID registry.

0003. Reserved

0004 A previous terminate conversation indication has been ignored.

0005 An RRMU was received but was not followed by a Change Direction indicator (i.e., the RECEIVE_AND_WAIT verb issued after receiving the RRMU, returned something other than WHAT_RECEIVED = SEND).

1019 Invalid Restart Byte Position.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0001 The restart byte position value specified in the DCMU is greater than 1 plus the value of the last byte received in the CRMU. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify the invalid restart byte position in the DCMU and the values specified in the CRMU. Supplemental report 1 contains the restart byte position value in the DCMU. Supplemental report 2 contains the last structure received value in the CRMU. Supplemental report 3 contains the last byte received value in the CRMU. If this value was not specified in the CRMU, this report will be omitted.

0002 The receiver does not support the byte-count restart elective, and the restart byte position value specified in the DCMU is not the beginning of the LLID structure following the last successfully received LLID structure. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify the invalid restart byte position in the DCMU and the values specified in the CRMU: Supplemental report 1 contains the restart byte position value in the DCMU. Supplemental report 2 contains the last structure received value in the CRMU. Supplemental report 3 contains the last byte received value in the CRMU. If this value was not specified in the CRMU, the report will be omitted.

0003 The receiver supports the byte-count restart elective, and the restart byte position value specified in the DCMU is not equal to 1 and is less than or equal to the last byte received value specified in the CRMU. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify the invalid restart byte position in the DCMU and the values specified in the CRMU; Supplemental report 1 contains the restart byte position value in the DCMU. Supplemental report 2 contains the last structure received value in the CRMU. Supplemental report 3 contains the last byte received value in the CRMU. If this value was not specified in the CRMU, the report will be omitted.

101A Invalid Control Vector Sequence: A control vector was found containing a key that was invalid for the position of the control vector within a TDU.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

nnmm Byte 2 following the sense code contains the key (nn) of the vector previous to the one in error; byte 3 contains the key (mm) of the vector in error.

101C Invalid Data Received

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Alteration of input data not allowed.

101D Insufficient Length: The length of the received signal is insufficient to contain additional required fields.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 A BIND was received that was too large to be extended. The BIND was rejected.

0002 An UNBIND was received that was too large to be extended. An UNBIND cleanup is sent on both session stages.

State Error (Category Code = X'20')

This category indicates a sequence number error, or an RH or RU that is not allowed for the receiver's current session control or data flow control state. These errors prevent delivery of the request to the intended component.

For LU 6.2, this category will be indicated within UNBIND or on negative response to BIND.

Category and modifier (in hexadecimal):

2001 Sequence Number: Sequence number received on normal-flow request was not 1 greater than the last.

2002 Chaining: Error in the sequence of the chain indicator settings (BCI, ECI), such as first, middle, first.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The receiver received a middle or end-chain request when in the in-chain state.

0002 The receiver received a begin-chain request when in the in-chain state.

2003 Bracket: Error resulting from failure of sender to enforce bracket rules for session. (This error does not apply to contention or race conditions.)

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The receiver received a begin-bracket request before receiving a response to its own previously sent begin-bracket request.

0002 The receiver received a begin-bracket request not specifying begin-bracket when in the between-bracket state.

0003 The receiver received an out-of-sequence LUSTAT command.

2004 Direction: Error resulting from a normal-flow request received while the half-duplex flip-flop state was not Receive.

2005 Data Traffic Reset: An FMD or normal-flow DFC request received by a half-session whose session activation state was active, but whose data traffic state was not active.

2006 Data Traffic Quiesced: An FMD or DFC request received from a half-session that has sent QUIESCE COMPLETE or SHUTDOWN COMPLETE and has not responded to RELEASE QUIESCE.

2007 Data Traffic Not Reset: A session control request (for example, STSN), allowed only while the data traffic state is reset, was received while the data traffic state was not reset.

2008 No Begin Bracket: An FMD request specifying BBI = BB was received after the receiver had previously received a BRACKET INITIATION STOPPED request.

2009 Session Control Protocol Violation: An SC protocol has been violated; a request, allowed only after a successful exchange of an SC request and its associated positive response, has been received before such successful exchange has occurred (for example, an FMD request has preceded a required CRYPTOGRAPHY VERIFICATION request). The request code of the particular SC request or response required, or X'00' if undetermined, appears in the fourth byte of the sense data.

200A Immediate Request Mode Error: The immediate request mode protocol has been violated by the request.

200B Queued Response Error: The Queued Response protocol has been violated by a request, i.e., QRI = ¬QR when an outstanding request had QRI = QR.

200C ERP Sync Event Error: The ERP sync event protocol in DFC has been violated; for example, after receiving a negative response to a chain, a request other than a request soliciting a synchronization event response was sent to DFC_SEND and rejected.

200D Response Owed Before Sending Request: An attempt has been made in half-duplex (flip-flop or contention) send/receive mode to send a normal-flow request when a response to a previously received request has not yet been sent.

200E Response Correlation Error: A response was received that cannot be correlated to a previously sent request.

200F Response Protocol Error: A violation has occurred in the response protocol; e.g., a +RSP to an RQE chain was generated.

2010 BIS Protocol Error: A BIS protocol error was detected; for example, a BIS request was received after a previous BIS was received and processed.

2012 Invalid Sense Code Received: A negative response was received that contains an SNA-defined sense code that cannot be used for the sent request.

2013 Decompression Protocol Error: A request containing compressed data was received in error.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The decompressor received a compressed RU without an expected Reset decompression control sequence. The compressor and the decompressor are not synchronized.

0002 The decompressor received a compressed RU containing an invalid decompression control sequence. The compressor and the decompressor are not synchronized.

0003 The length of the decompressed RU did not match the length given in the compression header.

RH Usage Error (Category Code = X'40')

This category indicates that the value of a field or combination of fields in the RH violates architectural rules or previously selected BIND options. These errors prevent delivery of the request to the intended component and are independent of the current states of the session. They may result from the failure of the sender to enforce session rules. Detection by the receiver of each of these errors is optional.

Category and modifier (in hexadecimal):

4001 Invalid SC or NC RH: The RH of a session control (SC) or network control (NC) request was invalid. For example, an SC RH with pacing request indicator set to 1 is invalid.

4002 Reserved

4003 BB Not Allowed: The Begin Bracket indicator (BBI) was specified incorrectly; for example, BBI = BB with BCI = ¬BC.

4004 CEB or EB Not Allowed: The Conditional End Bracket indicator (CEBI) or End Bracket indicator (EBI) was specified incorrectly; for example, CEBI = CEB when ECI = ¬EC or EBI = EB with BCI = ¬BC, or by the primary half-session when only the secondary may send EB, or by the secondary when only the primary may send EB.

4005 Incomplete RH: Transmission shorter than full TH-RH.

4006 Exception Response Not Allowed: Exception response was requested when not permitted.

4007 Definite Response Not Allowed: Definite response was requested when not permitted.

4008 Pacing Not Supported: The Pacing indicator was set on a request, but the receiving half-session or boundary function half-session does not support pacing for this session.

4009 CD Not Allowed: The Change Direction indicator (CDI) was specified incorrectly; for example, CDI = CD with ECI = ¬EC, or CDI = CD with EBI = EB.

400A No-Response Not Allowed: No-response was specified on a request when not permitted. (Used only on EXR.)

400B Chaining Not Supported: The chaining indicators (BCI and ECI) were specified incorrectly; for example, chaining bits indicated other than (BC,EC), but multiple-request chains are not supported for the session or for the category specified in the request header.

400C Brackets Not Supported: The bracket indicators (BBI, CEBI, and EBI) were specified incorrectly; e.g., a bracket indicator was set (BBI = BB, CEBI = CEB, or EBI = EB), but brackets are not used for the session.

400D CD Not Supported: The Change-Direction indicator was set, but is not supported.

400E Reserved

400F Incorrect Use of Format Indicator: The Format indicator (FI) was specified incorrectly; for example, FI was set with BCI = ¬BC, or FI was not set on a DFC request.

4010 Alternate Code Not Supported: The Code Selection indicator (CSI) was set when not supported for the session.

4011 Incorrect Specification of RU Category: The RU Category indicator was specified incorrectly; for example, an expedited-flow request or response was specified with RU Category indicator = FMD.

4012 Incorrect Specification of Request Code: The request code on a response does not match the request code on its corresponding request.

4013 Incorrect Specification of (SDI, RTI): The Sense Data Included indicator (SDI) and the Response Type indicator (RTI) were not specified properly on a response. The proper value pairs are (SDI = SD, RTI = negative) and (SDI = ¬SD, RTI = positive).

4014 Incorrect Use of (DR1I, DR2I, ERI): The Definite Response 1 indicator (DR1I), Definite Response 2 indicator (DR2I), and Exception Response indicator (ERI) were specified incorrectly; for example, a SIGNAL request was not specified with DR1I = DR1, DR2I = ¬DR2, and ERI = ¬ER.

4015 Incorrect Use of QRI: The Queued Response indicator (QRI) was specified incorrectly; for example, QRI = QR on an expedited-flow request.

4016 Incorrect Use of EDI: The Enciphered Data indicator (EDI) was specified incorrectly; for example, EDI = ED on a DFC request.

4017 Incorrect Use of PDI: The Padded Data indicator (PDI) was specified incorrectly; for example, PDI = PD on a DFC request.

4018 Incorrect Setting of QRI with Bidder's BB: The first speaker half-session received a BB chain requesting use of a session (via LUSTAT(X'0006')), but the QRI was specified incorrectly; that is, QRI = ¬QR.

4019 Incorrect Indicators with Last-In-Chain Request: A last-in-chain request has specified incompatible RH settings; for example, RQE*, CEBI = ¬CEB, and CDI = ¬CD.

4021 QRI Setting in Response Different From That in Request: The QRI setting in the response differs from the QRI setting in the corresponding request.

Path Error (Category Code = X'80')

This category indicates that the request could not be delivered to the intended receiver, because of a path outage, an invalid sequence of activation requests, or one of the listed path information unit (PIU) errors. Some PIU errors fall into other categories; for example, sequence number errors are sense code category X'20'. A path error received while the session is active generally indicates that the path to the session partner has been lost.

Category and modifier (in hexadecimal):

8001 Intermediate Node Failure: Machine or program check in a node providing intermediate routing function. A response may or may not be possible.

8002 Link Failure: Data link failure.

8003 NAU Inoperative: The NAU is unable to process requests or responses; for example, the NAU has been disrupted by an abnormal termination.

8004 Unrecognized Destination: A node in the path has no routing information for the destination specified either by the SLU name in a BIND request or by the TH.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 A request was received by a gateway function that could not be rerouted because of invalid or incomplete routing information.

8005 No Session: No half-session is active in the receiving end node for the indicated origination-destination pair, or no boundary function session connector is active for the origin-destination pair in a node providing the boundary function. A session activation request is needed.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The receiver received a request other than session control request when no LU-LU session was active.

0002 The receiver received a request other than session control request when no LU-SSCP session was active.

0003 The receiver received a session control request other than BIND/UNBIND when no LU-LU session was active.

0004 The receiver received an UNBIND when no LU-LU session was active.

0005 The receiver received a session control request other than ACTLU/DACTLU for the LU-SSCP session when no LU-SSCP session was active.

0006 The receiver received DACTLU when no LU-SSCP session was active.

8006 Invalid FID: Invalid FID for the receiving node.

8007 Segmenting Error: First BIU segment had less than 10 bytes; or Mapping field sequencing error, such as first, last, middle; or segmenting not supported and Mapping field not set to BBIU, EBIU; or (in APPN) an expedited request or response was received segmented (see Note 2 located at the end of this section).

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The node does not support receipt of segments, and a Mapping field value other than BBIU, EBIU was received. Sent in UNBIND.

0002 Interleaved BIND Segments Not Allowed: A BIND receiver that is in the middle of receiving segments of one BIND receives a segment from a different BIND; the receiver rejects both BINDs and disconnects the link.

8008 PU Not Active: The SSCP-PU secondary half-session in the receiving node has not been activated and the request was not ACTPU for this half-session; for example, the request was ACTLU from an SSCP that does not have an active SSCP-PU session with the PU associated with the addressed LU.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 A physical unit name was specified for an independent LU session stage. The specified PU name is either unknown or in an invalid state.

0002 NO ALS is defined for use by the origin independent LU. An implementation-defined automatic logon (autolog) request was specified for the subject resource, but, when the session establishment was attempted, no ALS was found to be associated with the subject resource.

8009 LU Not Active: The destination address specifies an LU for which the SSCP-LU secondary half-session has not been activated and the request was not ACTLU.

800A Too-Long PIU: Transmission was truncated by a receiving node because the PIU exceeded a maximum length or sufficient buffering was not available.

800B Incomplete TH: Transmission received was shorter than a TH (see Note 1 located at the end of this section).

800C DCF Error: Data Count field inconsistent with transmission length.

800D Lost Contact: Contact with the link station for which the transmission was intended has been lost, but the link has not failed. If the difference between link failure and loss of contact is not detectable, link failure (X'8002') is sent.

800E Unrecognized Origin: The origin address specified in the TH was not recognized.

800F The address combination is invalid.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 The (DAF', OAF') (FID2) combination or the LSID (FID3) specified an invalid type of session, for example, a PU-LU combination.

0001 The FID2 ODAI setting in a received BIND is incorrect; the BIND is rejected.

8010 Segmented RU Length Error: An RU was found to exceed a maximum length, or required buffer allocation that might cause future buffer depletion.

8011 ER Inoperative or Undefined: A PIU was received from a subarea node that does not support ER and VR protocols, and the explicit route to the destination is inoperative or undefined.

8012 Subarea PU Not Active or Invalid Virtual Route: A session-activation request for a peripheral PU or LU cannot be satisfied because there is no active SSCP-PU session for the subarea node providing boundary

function support, or the virtual route for the specified SSCP-PU (type 1 or type 2 nodes) or SSCP-LU session is not the same as that used for the SSCP-PU session of the type 1 or type 2 node's PU or the LU's subarea PU.

8013 Route Not Available: No route is available to connect the specified OSA and DSA for the specified COS.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

Byte 2 indicates the environment in which the failure was detected:

00 Single network

01 Interconnected network: Failure was detected at a node in a subnetwork other than that of the NAU sending the activation request.

Byte 3 indicates the reason for the failure:

00 No Specific Code Applies: This means an error occurred, but none of the conditions listed below applies.

01 No Mapping Specified: A session-activation request cannot be satisfied because for each VR in the VR identifier list for the session, no VR to ER mapping is specified.

02 No Explicit Routes Defined: A session-activation request cannot be satisfied because each VR in the VR identifier list for the session maps to a corresponding ER that is not defined.

03 No VR Resource Available: A session-activation request cannot be satisfied because each VR specified in the VR identifier list for the session requires a node resource that is not available.

04 No Explicit Routes Operative: A session-activation request cannot be satisfied because no underlying ER is operative for any VR specified in the VR identifier list for the session.

05 No Explicit Route Can Be Activated: A session-activation request cannot be satisfied because no VR specified in the VR identifier list for the session mapped to a defined and operative ER that could be activated.

06 No Virtual Route Can Be Activated: A session-activation request cannot be satisfied because no VR specified in the VR identifier list for the session can be activated by the PU, though for at least one VR an underlying ER is defined, operative, and activated.

07 No Virtual Route Identifier List Available: A session-activation request cannot be satisfied because a VR identifier list is not available.

Note: If none of the virtual routes specified in the VR identifier list for the session is active or can be activated, the reported reason is set based on a hierarchy of failure events. The "highest" of the failures that occurred within the set of virtual routes is returned on the response. For

example, if the VR manager receives a negative response to an NC_ACTVR request for a VR specified in the VR identifier list and for all other VRs in the list no VR to ER mapping is specified, then reason X'06' is reported. The hierarchy of the failure reasons is in ascending numeric order, that is, reason X'02' is higher than reason X'01'.

8014 No Path Exists to the Destination Node: Route selection services in the CP has determined from the topology database that no path exists to the destination node.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 No route to the destination node exists for the specified class of service.

0002 Invalid COS name received.

0003 The topology database indicates that the destination node is not available at this time; the node either has inconsistent data or is quiescing.

0004 The topology database indicates that the endpoint resources are depleted; the node is out of either half-session control blocks or message buffers.

0005 The length of the generated RSCV exceeds the maximum allowed.

8017 PIU from Adjacent Pre-ER-VR Subarea Node Rejected: A PIU that requires intermediate path-control routing was received by a subarea node from an adjacent subarea node that does not support ER-VR protocols, but the receiving subarea node does not support intermediate path-control routing for adjacent subarea nodes that do not support ER-VR protocols.

8018 Management Services component is unable to find or recognize the name of the application transaction program specified in the request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 The application transaction program specified in the request is not recognized by PUMS.

0002 The Cascaded Resource Name List is unrecognized.

0003 The Destination Application Name is unrecognized.

0004 The Destination Instance Identifier is unrecognized.

8019 Routing Exception: A node is unable to perform the required routing function for a request.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Unknown Routing Group Name.

0002 Unknown Routing Group Name, Routing Element Name combination.

0003 Reserved

0004 No connection is available for level of service required. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report containing the the service parameter triplet (or combination of triplets) for which a connection could not be found.

0005 The Routing and Targeting Instructions GDS variable is required but is not present.

801A Confirmation of Forwarding

0001 The identified request has been successfully forwarded by the reporting node. When this SNA report code is used within an SNA condition report, the reported-on destination list identifies the list of destinations to which the request has been forwarded.

801B Confirmation of Acceptance

0001 The identified request has been successfully received by the intended destination(s) at the reporting node. When this SNA report code is used within an SNA condition report, the reported on destination list identifies the list of destinations for which the request has been accepted.

801C Hop Count Exhausted

0001 The request has been forwarded by an excessive number of nodes (e.g., the count has been decremented at each node and has reached 0) and, therefore, the request could not be delivered to one or more destinations. Typically, this exception indicates that one or more nodes have incorrectly routed or directed the request. The exception may also indicate that the routing/directing count was not appropriately initiated according to network size.

8020 Session Reset: The LU-LU session identified in the UNBIND is being deactivated because of a reset condition.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Virtual Route Inoperative: The virtual route used by the LU-LU session has become inoperative, thus forcing the deactivation of the identifed LU-LU session.

0002 Hierarchical Reset of Both XRF-active and XRF-backup Sessions: The XRF-backup session has failed; therefore, both the XRF-active and XRF-backup session are being reset.

0003 Virtual Route Deactivated: The identified LU-LU session had to be deactivated because of a forced deactivation of the virtual route being used by the LU-LU session.

0004 Route Extension Failure: The route extension used by the LU-LU session has become inoperative, thus forcing the deactivation of the identified LU-LU session.

0005 Route Extension Failure: The route extension used by the XRF-backup LU-LU session has become inoperative, thus forcing the deactivation of the identified XRF-backup LU-LU session.

0006 Virtual Route Inoperative: The virtual route used by the LU-LU session has become inoperative, thus forcing the deactivation via VR-INOP of the identifed XRF-backup LU-LU session.

0007 Third-Party Termination: The network operator caused the forced or cleanup termination of the LU-LU session.

0008 BFTERM has been received with no indication of the cause of the reset.

0009 Termination was requested by the dependent SLU with a TERMINATE SELF or character-coded logoff.

Notes:

1. It is generally not possible to send a response for this exception condition, since information (FID, addresses) required to generate a response is not available. It is logged as an error if this capability exists in the receiver.
2. If segmenting is not supported, a negative response is returned for the first segment only, since this contains the RH. Subsequent segments are discarded.

Chapter 11. Function Management (FM) Headers

For sessions that support FM headers, the request header (RH) contains a format indicator (FI) that, when *on*, indicates that an FM header is at the beginning of an FMD request unit (RU).

FM headers appear only at the beginning of an RU. An RU containing an FM header may appear anywhere within a chain. When the FM header is longer than one RU will hold, the header is continued in as many additional RUs of a chain as needed to hold it. Figure 11-1 and Figure 11-2 show the placement of FM headers within an RU:

Figure 11-1. FM Header Contained in One RU

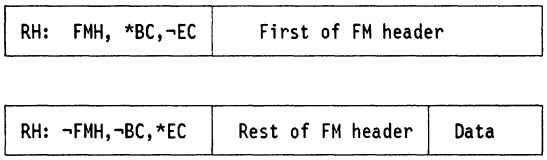
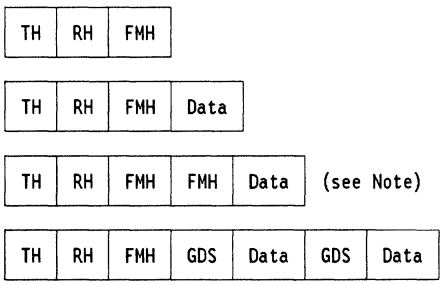


Figure 11-2. FM Header Contained in Two Contiguous RUs of a Chain

Figure 11-3 shows some instances where FM headers are used and Figure 11-4 identifies the logical unit (LU) types that use each FM Header.



FMH — Function Management (FM) Header
GDS — General Data Stream identifier
TH — Transmission Header
RH — Request/Response Header

Note: In LU type 6.2 a maximum of one FM header per RU is allowed.

Figure 11-3. Usage of FM Headers

LU Type	FM Header Type
0	None required, but may use any header
1	1, 2, 3
2	None
3	None
4	1, 2, 3
6.1	4, 5, 6, 7, 8, 10
6.2	5, 7, 12
7	None

Figure 11-4. LU Types That Support FM Headers

FM Header 1

This header is used to select a destination within a logical unit (LU). A destination may be represented by a device, a data set residing on a device, or merely a data stream. The LU initiates, interrupts, resumes, and concludes data traffic for the half-session using the FMH-1.

FM Header 1

Byte	Bit	Content
0		Length, in binary, of FMH-1, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-1 1 another FMH follows this FMH-1
	1–7	Type: 0000001

FM Header 1

Byte	Bit	Content
2	0–3	Select desired medium for data (see Notes 1 and 2): 0000 console 0001 exchange 0010 card 0011 document 0100 nonexchange disk 0101 extended document 0110 extended card 0111 data set name select destination (see Note 3) 1000 word processing (WP) media 1 1001 WP media 2 1010 WP media 3 1011 reserved 1100 WP media 4 1101 reserved 1110 reserved 1111 reserved
	4–7	Logical subaddress (see Note 2): 0000–1110 specific device in medium class 1111 any device in medium class (see Note 3)
3	0	SRI: Stack Reference indicator: 0 stack to be used is the sender's send stack 1 stack to be used is the receiver's send stack
	1	Demand select: 0 receiver may direct data to alternate medium/subaddress 1 receiver must direct data to specified medium/subaddress (spooling is prohibited)
	2–3	Reserved
	4–7	DSPs: data stream profiles: 0000 default (the DSP is implied by the Medium Select field) 0001 base 0010 general 0011 job 0100 WP raw-form text 0101 WP exchange diskette 0110 reserved 0111 Office Information Interchange level 2 1000 reserved 1001 reserved 1010 document interchange 1011 structured field 1100 reserved 1101 reserved 1110 reserved 1111 reserved

FM Header 1

Byte	Bit	Content
4		FMH-1 properties
	0–2	DSSEL: destination selection: 000 resume 001 end 010 begin 011 begin/end 100 suspend 101 end-abort 110 continue 111 reserved
	3	DST: data set transmission (see Note 6): 0 transmission exchange format 1 basic exchange format
	4	Reserved
	5	CMI: FMH-1 SCB compression indicator (see Notes 4 and 5): 0 no FMH-1 SCB compression 1 FMH-1 SCB compression (the first byte following the FMH(s) is a string control byte)
	6	CPI: compaction indicator (see Notes 4 and 5): 0 no compaction 1 compaction (the first byte following the FMH(s) is a string control byte)
	7	Reserved
5	0–7	ECRL: exchange record length if medium select = exchange or card; otherwise, reserved. For medium select = card, a hexadecimal value indicates maximum card length: 00000000 80-column length
6–7		Reserved (optional)
8		DSLEN: length of destination name (optional)
9–n		DSNAME: destination name (optional; reserved when DSSEL = continue)

FM Header 1

Byte	Bit	Content

Notes:

1. The data stream profile (DSP) defaults for the Medium Select field are:

FMH-1 MEDIUM SELECT	*DEFAULT DSP*
Console, X'0'	Base
Exchange, X'1'	DST field of FMH-1
Card, X'2'	SCS (IRS, TRN)
Document, X'3'	Subset 2 (RJE)
Nonexchange Disk, X'4'	DST field of FMH-1
Extended Document,X'5'	Subset 2 (RJE)
Extended Card, X'6'	SCS (IRS, TRN)
WP Medium 1, X'8'	WP Raw Form
WP Medium 2, X'9'	WP Raw Form
WP Medium 3, X'A'	WP Raw Form
WP Medium 4, X'C'	WP Raw Form

   An LU requiring any other DSP value associated with Medium Select does so by specifying the desired DSP in byte 3, bits 4−7 of the FMH-1. This selection adheres to those DSPs allowed on the session as specified in the BIND parameters.

2. Medium Select and Logical Subaddress fields are reserved when the Destination Selection (DSSEL) field is set to 110 (continue), 001 (end), 100 (suspend), or 101 (end-abort).

3. If Medium Select = X'7' and Logical Subaddress = X'F', the Destination Name (DSNAME) field is used to select destination.

4. CMI, CPI, and ERCL indicators are meaningful and valid only when specified in a Begin, Begin/end, or Continue FMH-1.

5. CMI, CPI, and ERCL information received when DSSEL = Continue overlays the settings of the Begin FMH-1 or the last-received Continue FMH-1.

6. When Medium Select is not equal to Exchange, this field is reserved. Receiver may do spooling and exchange-medium creation locally. When Medium Select = Exchange, specifying 0 preserves chain boundaries while spooling, but nonsequential allocation techniques may be used. Specifying 1 does not preserve chain boundaries, but uses sequential medium allocation.

FM Header 2

Once a destination has been selected using a FMH-1, this header handles the data management tasks for that destination.

FM Header 2

Byte	Bit	Content
0		Length, in binary, of FMH-2, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-2 1 another FMH follows this FMH-2
	1–7	Type: 0000010
2	0	SRI: stack reference indicator (see Note below): 0 FMH-2 pertains to the active destination of the sending half-session's send stack and the receiving half-session's receive stack 1 FMH-2 pertains to the active destination of the receiving half-session's send stack and the sending half-session's receive stack
	1–7	FMH-2 function to be performed (see Note for specific values): Identifies the function that this FMH-2 is to perform
3–n		Parameter fields (These fields provide the information needed to perform the selected function. They are different for each FMH-2 function, and are described in *SNA: Sessions Between Logical Units.*)

Note: Byte 2 of the FMH-2 contains the Stack Reference indicator (SRI) and defines the function to be performed. The valid combinations of SRI and function codes are:

Function Code	*Function*
X'01'	Peripheral data information record (PDIR)
X'02'	Compaction table
X'04'	Prime FMH-1 SCB compression character
X'07'	Execute program offline
X'20'	Create data set
X'21'	Scratch data set
X'22'	Erase data set
X'23'	Password
X'24'	Add
X'25'	Replace
X'26'	Add replicate
X'27'	Replace replicate
X'28'	Query for data set
X'29'	Note
X'2B'	Record ID
X'2C'	Erase record
X'2D'	Scratch all data sets
X'2E'	Volume ID
X'AA'	Note reply (SRI is always *on*)

FM Header 3

This header handles data management tasks that are common to all destinations in the LU-LU session.

The FMH-3 format is identical to the FMH-2 format except that an FMH-3 does not have a Stack Reference indicator (SRI) in byte 2. An FMH-3 is used when information is needed or used by all destinations managed by the half-session. By contrast, an FMH-2 is used for a specific destination.

Two functions, the compaction table and the prime FMH-1 SCB compression character, can be sent in an FMH-2 or FMH-3. They are sent in an FMH-2 when they apply to a specific destination at the half-session and in an FMH-3 when they apply to all destinations at the half-session.

The FMH-3 functions are as follows:

Function Code	*Function*
X'02'	Compaction table
X'03'	Query for compaction table
X'04'	Prime FMH-1 SCB compression character
X'05'	Status
X'06'	Series ID

FM Header 4

This header carries a logical block command and its parameters that, together with information, apply to a logical block within a logical message as defined for Logical Message Service.

FM Header 4

Byte	Bit	Content
0		Length, in binary, of FMH-4, including this Length byte
1	0 1–7	FMH concatenation (must be 0) Type: 0000100
2		FMH4FXCT: length of fixed length parameters excluding the length of FMH4FXCT. The first nonfixed parameter position is FMH4LBN. The minimum value of FMH4FXCT is 3, the maximum is 4.

FM Header 4

Byte	Bit	Content
3		FMH4TT1: block transmission type: X'00' inherit code (from MM-TT register) X'01'–X'3F' reserved X'40' FFR-FNI record X'41' FFR-FS record X'42' FFR-FS2 record X'43'–X'4F' reserved X'50'–X'FE' reserved X'FF' reserved *Note:* FFR = field formatted record, FNI = fixed fields without field separators, FS = fixed fields with field separators, FS2 = fixed fields with or without field separators.
4		FMH4TT2: block transmission type qualifier: reserved except for FMH4TT1 = X'41' or X'42', in which case it holds the separator value
5		FMH4CMD: command: X'00' CRT-NU-BLK X'02' CRT-SU-BLK X'03' CRT-SN-BLK X'10' CONT-NU-BLK X'12' CONT-SU-BLK X'13' CONT-SN-BLK X'23' DEL-SN-BLK X'32' UPD-SU-BLK X'33' UPD-SN-BLK X'42' RPL-SU-BLK X'43' RPL-SN-BLK *Note:* NU = nonshared, unnamed; SU = shared, unnamed; SN = shared, named; NN = nonshared, named
6		FMH4FLAG: flags (if omitted, X'00' is assumed):
	0–1	Reserved
	2–3	F4RDESCR: record descriptor flag: 00 no logical record headers (LRHs) in transmission block 01 LRHs present, with implicit lengths 10 reserved 11 reserved
	4–5	Reserved
	6	FMH4BDTF: block data transform flag: 0 FMH4BDT absent 1 FMH4BDT present
	7	FMH4RDTF: reserved
7		FMH4LBN: length of FMH4BN (X'00', or omitted, if unnamed block)
8–m		FMH4BN: name of block
m+1		FMH4LBDT: length of FMH4BDT (X'00' if FMH4BDTF is 0)

FM Header 4

Byte	Bit	Content
m+2−n		FMH4BDT: block data transform
n+1		FMH4LVID: length of FMH4VID
n+2−p		FMH4VID: version identifier

FM Header 5: Attach (LU 6.2)

LU type 6.2 uses this header to carry a request for a conversation to be established between two transaction programs. This header identifies the transaction program that is to be put into execution and connected to the receiving half-session.

When a transaction program issues an ALLOCATE verb naming a transaction program to be run at the other end of the conversation, an Attach FMH-5 carries the transaction program name (TPN) to the receiving half-session.

FM Header 5: Attach (LU 6.2)

Byte	Bit	Content
0		Length, in binary, of FMH-5, including this Length byte
1	0	Reserved
	1−7	Type: 0000101
2−3		Command code: X'02FF' (Attach)
4	0−2	Security Indicators
	0	Already-verified indicator: 0 user ID is not already verified 1 user ID is already verified (Password subfield not included in this Attach)
	1−2	Persistent-verification indicator: 00 persistent verification not supported or needed 01 sign-on requested 10 already signed on (Password subfield not included in this Attach) 11 reserved
	3	Reserved
	4	Program initialization parameter (PIP) presence: 0 PIP not present following this FMH-5 1 PIP present following this FMH-5 (see "PIP Variable" on page 11-11 for format)
	5−7	Reserved
5		Length (j-5), in binary, of Fixed Length Parameters field (currently 3—future expansion possible)
6−j		Fixed Length Parameters

FM Header 5: Attach (LU 6.2)

Byte	Bit	Content
6		Resource type: X'D0' basic conversation X'D1' mapped conversation
7		Reserved
8(=j)	0–1	Synchronization level: 00 none 01 confirm 10 confirm, sync point, and backout 11 reserved
	2–7	Reserved
j+1–p		Variable Length Parameters
j+1–k		Transaction Program Name Field:
j+1		Length (values 1 to 64 are valid), in binary, of transaction program name
j+2–k		Transaction program name: a symbol string identifying a transaction program name known at the receiver; receivers may constrain such names to be type A, AE, GR, or DB, depending on the implementation
k+1–m		Access Security Information Field:
k+1		Length (0 or m-k-1), in binary, of Access Security Information subfields
k+2–m		Zero or more Access Security Information subfields (see "Access Security Information Subfields" on page 11-11 for format)
m+1–n		Logical-Unit-of-Work Identifier Field:
m+1		Length (values 0 and 10 to 26 are valid), in binary, of Logical-Unit-of-Work Identifier field
m+2–n		Logical-Unit-of-Work Identifier
m+2		Length (values 1 to 17 are valid), in binary, of network-qualified LU name
m+3–w		Network-qualified LU network name
w+1–w+6		Logical-unit-of-work instance number, in binary
w+7–w+8(=n)		Logical-unit-of-work sequence number, in binary
n+1–p		Conversation Correlator Field:
n+1		Length (values 0 to 8 are valid), in binary, of conversation correlator of sender
n+2–p		Conversation correlator of the sending transaction: a 1- to 8-byte symbol-string type G identifier (unique between partner LUs) of the conversation being allocated via FMH-5 (an example construction of this field would be the composition of a transaction program instance identifier and a resource identifier)
Note:		Trailing Length fields (bytes n+1, m+1, and k+1) that have value X'00' can be omitted.

Access Security Information Subfields

The Access Security Information subfields in FMH-5 have the following formats:

Access Security Information Subfields

Byte	Bit	Content
0		Length (valid values are 1 to 11), in binary, of remainder of subfield—does not include this Length byte
1		Subfield type: X'00' profile X'01' password X'02' user ID
2−i		Data: a symbol string identifying access security information known at the receiver; receivers may constrain such information to be type A, AE, GR, DB, or 1134, depending on the implementation. *Note:* The length of the symbol string may be less than the length of the Data field; in this case, the symbol string is left-justified within the Data field and the Data field is filled out to the right with space (X'40') characters. Space characters, if present, are not part of the symbol string.
Note:		The Access Security Information subfields may appear in any order in the Access Security Information field of the FMH-5. The profile and password subfields are not present if no user ID subfield is present. For full details of the conditions of presence and the receive checks for these subfields, see the ATTACH_SECURITY_CHECK procedure described in &peerlu..

PIP Variable

The PIP GDS variable is present following FMH-5 Attach if the PIP Presence indicator (byte 4, bit 4) in the Attach is set to 1. Although not part of the Attach (i.e., the Attach Length byte does not include its length), it is shown here because of its affinity with the Attach.

PIP Variable

Byte	Bit	Content
0−1		Length (4 or n+1), in binary, of PIP variable, including this Length field
2−3		GDS indicator: X'12F5'
4−n		Zero or more PIP subfields, each of which has the following format (shown in "PIP Subfield" using 0-origin)

PIP Subfield: Zero or more of these subfields are contained in a PIP variable (see "PIP Variable").

PIP Subfield

Byte	Bit	Content
0–1		Length, in binary, of PIP subfield, including this Length field
2–3		GDS indicator: X'12E2'
4–m		PIP subfield data: type-G symbol string is valid

FM Header 5: Attach (Not LU 6.2)

This header flows from the program using the sending half-session to the attach manager of the receiving half-session. This header identifies the program at the receiving LU that it wishes to have attached. An FMH-5 can be followed by other FMHs (for example, FMH-6, FMH-8, and FMH-4), a logical record header (LRH), and FM data. Optionally, it can be sent with CD or EB.

FM Header 5: Attach (Not LU 6.2)

Byte	Bit	Content
0		Length, in binary, of FMH-5, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-5 1 another FMH follows this FMH-5
	1–7	Type: 0000101
2–3		FMH5CMD: command code: X'0202' attach transaction program X'0204' reset attached process X'0206' data descriptor
4		FMH5MOD: modifier
5		FMH5FXCT: fixed-length parameters: X'00' reset attached process X'02' attach transaction program, data descriptor
6		ATTDSP
7		ATTDBA
8–n		Resource names

FM Header 6

This header flows from a currently active transaction program using a sending half-session to a currently active transaction program using a receiving half-session.

FM Header 6

Byte	Bit	Content
0		Length, in binary, of FMH-6, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-6 1 another FMH follows this FMH-6
	1–7	Type: 0000110
2–3		Command code (CC2): For service transaction programs, the first byte of the command code identifies a transaction program and the second byte identifies a function within a transaction program.
4		FMH6MOD: modifier
	0	FMH6LNSZ: length of parameter length fields: 0 1-byte field 1 2-byte field
	1–7	Reserved
5–n		Fixed: total length of fixed length parameters (LF): This field contains the sum of the lengths of all fixed length parameters that are mandatory for the particular command code located in bytes 2 and 3. This field is either one byte or two bytes in length, based on the setting of FMH6LNSZ (0 = one byte; 1 = two bytes).
n+1–m		Fixed length parameters (FDy): the fixed length parameters are positional by command code
m+1–p		Variable: length field of first, positional variable-length parameter (LV1): This field is either one byte or two bytes in length, based on the setting of FMH6LNSZ (0 = one byte; 1 = two bytes). If the Length field (LVx) is equal to 0, then the variable parameter is omitted. The next positional variable-length parameter length (LV2) occurs in byte q+1.
p+1–q		Variable-length positional parameter (VD). The LV and VD fields are replicated to represent x number of variable-length parameters according to command code.

FM Header 7: Error Description (LU 6.2)

LU type 6.2 uses this header, following a negative response (0846), to carry information that relates to an error on the session or conversation. For example, an FMH-7 and additional error information are sent when an FMH-5 (Attach) specifies a nonexistent transaction program name.

FM Header 7: Error Description (LU 6.2)

Byte	Bit	Content
0		Length (7), in binary, of FMH-7, including this Length byte
1	0	Reserved
	1–7	Type: 0000111

FM Header 7: Error Description (LU 6.2)

Byte	Bit	Content
2–5		SNA-defined sense data listed below; the phrases following the sense data values are the symbolic return codes provided to the application program in LU 6.2 verbs (see *SNA Transaction Programmer's Reference Manual for LU Type 6.2*) when the sense data is received. See Chapter 10, "Sense Data" on page 10-1 for additional details on the sense data.

Sense Data	Return Code – Secondary Return Code (if present)
1008600B	RESOURCE_FAILURE_NO_RETRY
10086021	ALLOCATION_ERROR – TPN_NOT_RECOGNIZED
10086031	ALLOCATION_ERROR – PIP_NOT_ALLOWED
10086032	ALLOCATION_ERROR – PIP_NOT_SPECIFIED_CORRECTLY
10086034	ALLOCATION_ERROR – CONVERSATION_TYPE_MISMATCH
10086041	ALLOCATION_ERROR – SYNC_LEVEL_NOT_SUPPORTED_BY_PGM
080F0983	ALLOCATION_ERROR – ACCESS_DENIED
080F6051	ALLOCATION_ERROR – SECURITY_NOT_VALID
08240000	BACKED_OUT (resync not in progress: The state of the entire subtree headed by the sender is backed out. See Note.)
08240001	BACKED_OUT (resync in progress: The state of one or more other partners of the sender is unknown. See Note.)
084B6031	ALLOCATION_ERROR – TP_NOT_AVAIL_RETRY
084C0000	ALLOCATION_ERROR – TP_NOT_AVAIL_NO_RETRY
08640000	DEALLOCATE_ABEND_PROG
08640001	DEALLOCATE_ABEND_SVC
08640002	DEALLOCATE_ABEND_TIMER
08890000	PROG_ERROR_NO_TRUNC or PROG_ERROR_PURGING
08890001	PROG_ERROR_TRUNC
08890100	SVC_ERROR_NO_TRUNC or SVC_ERROR_PURGING
08890101	SVC_ERROR_TRUNC

Note: On a BACKOUT verb, the two X'0824' sense data values cause an OK rather than BACKED_OUT return code, since the meaning of OK is equivalent to the meaning of BACKED_OUT on this verb. The two X'0824' sense data values may cause different secondary return codes (ALL_AGREED and LUW_OUTCOME_PENDING) to be returned for the SYNCPT, (MC_)PREPARE_FOR_SYNCPT, and BACKOUT verbs. Other verbs have no secondary return code for the the BACKED_OUT return code. For the mapping of these two sense data values to secondary return codes, see the sync point chapter in *SNA LU 6.2 Reference: Peer Protocols.*

6	0	Error log variable presence: 0 no error log variable follows this FMH-7 1 error log GDS variable follows this FMH-7
	1–7	Reserved

FM Header 7: Error Description (Not LU 6.2)

This header is sent after a negative response (0846) to provide further information about an error.

FM Header 7: Error Description (Not LU 6.2)

Byte	Bit	Content
0		Length, in binary, of FMH-7, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-7 1 reserved
	1–7	Type: 0000111
2–5		ERPSENSE: SNA-defined sense data, which would appear on error response (see Chapter 10, "Sense Data" on page 10-1)
6–7		ERPSEQ: sequence number of RU chain in which error was detected

FM Header 8

This header is used only with IMS/VS logical message services that use LU type 6.1 protocols. Refer to the IMS publications for the formats and meanings of the bytes in this header.

FM Header 10

This header is sent to prepare the session for a sync point. It may be sent with data. The RU chain has CDI = CD so that the receiver may, on the next flow, request a sync point or abort the unit of work.

FM Header 10

Byte	Bit	Content
0		Length, in binary, of FMH-10, including this length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-10 1 another FMH follows this FMH-10
	1–7	Type: 0001010

FM Header 10

Byte	Bit	Content
2–3		SPCCMD: sync point command: X'0202' Prepare command
4–5		SPCMOD: sync point modifier For a Prepare command (FMH-10), the modifier indicates RH settings to be returned on the first RU chain sent by the FMH-10 receiver. X'0000' *CD, *EB: The sender of FMH-10 does not care what RH settings are returned on the reply. X'0001' EB: The sender of FMH-10 requires an EB on the reply. X'0002' CD, ¬EB: The sender of FMH-10 requires a CD on the reply.

FM Header 12: Security

LU type 6.2 uses this header during LU-LU verification. This header is used to return to the partner LU the enciphered version of the clear random data received in + RSP(BIND).

The function management header 12 (FMH-12) has the following format:

FM Header 12: Security

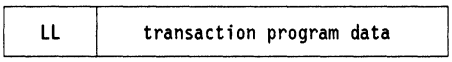
Byte	Bit	Content
0		Length (10), in binary, of FMH-12, including this Length byte.
1	0 1–7	Reserved Type: 0001100
2–9		Enciphered version of the random data received in the Random Data field of RSP(BIND) enciphered version of the clear random data received in + RSP(BIND).

Chapter 12. Presentation Services (PS) Headers

Presentation Services (PS) Headers

Presentation services (PS) headers convey sync point information between PS component sync point managers when the conversation using the session is allocated specifying the sync-point synchronization level. These headers are used only by LU type 6.2.

Typically, transaction program data exchanged over LU 6.2 sessions uses a 2-byte length field prefix called an LL. The LL specifies the number of bytes contained in the transaction program data plus 2 (the length of the LL field itself).



PS headers, however, deliberately violate this format. All PS headers are identified by an LL of X'0001' immediately preceding the header. X'0001' is an invalid LL value for use by transaction programs because the LL's value must include the length of itself, which is 2 bytes. All LLs indicating a length of less than 2 are reserved for use by the LU. The format of PS headers is shown below.

PS Header 10: Sync Point Control

Presentation services header 10 (Sync Point Control) has the following format:

PS Header 10: Sync Point Control

Byte	Bit	Content
0		Length, in binary, of PS header, including this length field
1	0	Reserved
	1–7	Type: 0001010 sync point control (only value defined)
2		Flag byte *Note:* The Flag byte is different for each sync point command type specified in byte 3. The Flag byte is reserved and set to X'00' if the cold-start Exchange Log Name GDS variable indicates the partner does not use PS header byte 2 as the Flag byte.

PS Header 10: Sync Point Control

Byte	Bit	Content
Prepare flags:		
	0–4	Reserved
	5	LOCKS parameter indicator (used when PS header modifier, bytes 4-5, is request RECEIVE; otherwise, reserved): 0 LOCKS(SHORT) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL). 1 LOCKS(LONG) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL).
	6–7	Reserved
Request Commit flags:		
	0	Support of the New LUWID PS header: 0 Not all participants in the subtree that include the sync point manager sending this Request Commit support receipt of the New LUWID PS header. 1 All participants in the subtree that include the sync point manager sending this Request Commit support receipt of the New LUWID PS header.
	1–4	Reserved
	5	LOCKS parameter indicator (used when PS header modifier, bytes 4-5, is request RECEIVE; otherwise, reserved): 0 LOCKS(SHORT) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL). 1 LOCKS(LONG) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL).
	6–7	Reserved
Committed flags:		
	0	Reserved
	1	Resync processing status (sent from last agent; otherwise, reserved): 0 not in progress: the state of the entire subtree is committed 1 in progress: the state of one or more agents of the sender is unknown
	2	Source of next LUWID: 0 to be generated locally by receiver of this Committed 1 New LUWID PS header provided by sender of this Committed
	3–7	Reserved
Forget flags:		
	0	Support of the New LUWID PS header (reserved if replying to Committed): 0 Not all participants in the subtree that includes the sync point manager sending this Forget support receipt of the New LUWID PS header. 1 All participants in the subtree that include the sync point manager sending this Forget support receipt of the New LUWID PS header.
	1	Resync processing indicator: 0 not in progress: the state of the entire subtree is committed 1 in progress: the state of one or more agents of the sender is unknown
	2–7	Reserved

PS Header 10: Sync Point Control

Byte	Bit	Content
HM flags:		
	0–1	Reserved
	2	Source of next LUWID (sent in reply to Request Commit; otherwise, reserved): 0 to be generated locally by receiver of this HM 1 New LUWID PS header provided by sender of this HM
	3–7	Reserved
New LUWID flags:		
	0–7	Reserved
3		Sync point command type: X'05' Prepare X'06' Request Commit X'07' Committed X'08' Forget X'09' Heuristic Mixed X'0A' New LUWID
4–n		Command-specific information (present only for Prepare (X'05'), Request Commit (X'06'), and New LUWID (X'0A') commands. Data in this field depends on the value of byte 3, as shown below: **05 (Prepare)** Modifier specifying next flow **06 (Request Commit)** Modifier specifying next flow **0A (New LUWID)** LUWID for next transaction
If byte 3 = X'05' or X'06':		
4–5(=n)		Modifier specifying next flow (reserved, when byte 3 = X'06' and the Request Commit is being sent from a not-last agent to its initiator): X'0000' request RECEIVE X'0001' request DEALLOCATE X'0002' request SEND *Note:* Bytes 4-5 affect the Change Direction indicator (CDI) and Conditional End Bracket indicator (CEBI) settings of the RH for the last PS header in the sync point sequence. For example, the CDI and CEBI bits on the Forget command are affected when Prepare was the first PS header received; similarly, the CDI and CEBI on the Committed command are affected when Request Commit was the first PS header received.
If byte 3 = X'0A':		
4		Length (values 10 to 26 are valid), in binary, of Logical-Unit-of-Work Identifier field (bytes 5-n). Since the value may not be 0 (the LUWID may not be omitted), the value of byte 4 is 9 greater than the value of byte 5.
5–n		Logical-Unit-of-Work Identifier for the next logical unit of work
5		Length (values 1 to 17 are valid), in binary, of network-qualified LU name

PS Header 10: Sync Point Control

Byte	Bit	Content
6 − m		Network-qualified LU name (format described in Chapter 5, "Request/Response Units (RUs)" on page 5-1)
m + 1 − m + 6		Logical-unit-of-work instance number, in binary
w + 7 − w + 8(= n)		Logical-unit-of-work sequence number, in binary

Chapter 13. GDS Variables for SNA Service Transaction Programs (STPs)

General Context

This chapter describes GDS variables that are used by SNA service transaction programs that use LU 6.2 session protocols (including over CP-CP sessions). See *SNA Transaction Programmer's Reference Manual for LU Type 6.2* for a complete list of the currently defined service TP names (TPNs); TPNs are specified in FMH-5s (Attaches).

Refer to Chapter 14, "SNA/DS FS1 Encodings" or to Chapter 15, "SNA/DS FS2 Encodings" for additional SNA/DS information and refer to Chapter 16, "GDS Variables for Application Programs" for information about GDS variables that are not specific to SNA service transaction programs.

See also Appendix B, "GDS ID Description and Assignments" on page B-1 for a discussion of the general notion of general data stream (GDS) structured fields and a comprehensive list of the block assignments of GDS identifiers by architecture (or other use).

Descriptions of GDS Variables for SNA STPs

Change Number of Sessions (X'1210') GDS Variable

Change Number of Sessions (X'1210') GDS Variable

Byte	Bit	Content
0–1		Length (17 or n + 1), in binary, of Change Number of Sessions GDS variable, including this Length field
2–3		GDS ID: X'1210'
4		Service flag:
	0–3	Reserved
	4–7	Request/reply indicator: 0010 request 1000 reply, function completed abnormal 1010 reply, function accepted but not yet completed
5		Reply modifier (reserved if byte 4, bits 4–7 = 0010): X'00' normal—no negotiation performed X'01' abnormal—command race detected X'02' abnormal—mode name not recognized X'03' reserved X'04' normal—negotiated reply X'05' abnormal—(LU,mode) session limit is 0
6		Action: X'00' set (LU,mode) session limits X'01' reserved X'02' close
7		Drain immediacy:
	0–2	Reserved
	3	Source LU drain (reserved if byte 6 ≠ 02): 0 no (send BIS at next opportunity) 1 yes
	4–6	Reserved
	7	Target LU drain (reserved if byte 6 ≠ 02): 0 no (send BIS at next opportunity) 1 yes
8		Action flags:
	0–6	Reserved
	7	Session deactivation responsibility: 0 sender of Change Number of Sessions request (source LU) 1 receiver of Change Number of Sessions request (target LU) *Note:* Bytes 9–14 are reserved if byte 6 ≠ 0.

Change Number of Sessions (X'1210') GDS Variable

Byte	Bit	Content
9–10		(LU,mode) session limit:
	0	Reserved
	1–15	Maximum (LU,mode) session count, in binary
11–12		Source LU contention winners:
	0	Reserved
	1–15	Guaranteed minimum number of contention winner sessions at source LU, in binary
13–14		Target LU contention winners:
	0	Reserved
	1–15	Guaranteed minimum number of contention winner sessions at target LU, in binary
15		Mode name selection:
	0–6	Reserved
	7	Mode names affected by this command: 0 a single mode name is affected 1 all mode names are affected (valid if byte 6 = X'02')
16		Length (values 0 to 8 are valid; reserved if byte 15, bit 7 = 1), in binary, of mode name
17–n		Mode name (omitted if byte 16 = X'00')

Exchange Log Name (X'1211') GDS Variable

Exchange Log Name (X'1211') GDS Variable

Byte	Bit	Content
0–1		Length (p+1 or r+1), in binary, of Exchange Log Name GDS variable, including this Length field
2–3		GDS ID: X'1211'
4		Service flag:
	0–3	Reserved
	4–7	Request/reply indicator: 0010 request 1000 reply, function completed abnormally: A log name or warm/cold log status mismatch was detected. 1001 reply, function completed normally

Exchange Log Name (X'1211') GDS Variable

Byte	Bit	Content
5		Sync point manager support and status flags:
	0	Reserved
	1	Presence of the LU name of the creator of the conversation correlator in Compare States: 0 not present 1 present
	2	Ability of the LU to treat byte 2 of the PS header as a Flag byte and accept the X'08240001' sense data value (Backout Initiated—Resync in Progress) in FMH-7: 0 not able: Byte 2 of the PS header is reserved; X'08240001' is not accepted in FMH-7. 1 able: Byte 2 of the PS header contains flags; X'08240001' is accepted in FMH-7.
	3–6	Reserved
	7	Log status: 0 cold 1 warm
6		Length (values 1 to 17 are valid), in binary, of network-qualified LU name
7–n		Network-qualified LU name
n+1–p		Name of the log at this LU
n+1		Length (values 1 to 64 are valid), in binary, of the local LU's log name
n+2–p		Local LU's log name: a type-AE symbol string
p+1–r		Name of the log at the partner LU (not included if log status is cold)
p+1		Length (values 1 to 64 are valid), in binary, of the partner LU's log name
p+2–r		Partner LU's log name, a type-AE symbol string

Control Point Management Services Unit (X'1212') GDS Variable

CP-MSU carries MS requests and data in general data stream (GDS) format.

Control Point Management Services Unit (X'1212') GDS Variable

Byte	Bit	Content
0–1		Length (m+1), in binary, of the CP-MSU
2–3		GDS ID: X'1212'
4–m		One or more MS major vectors, as described (using 0-origin indexing) in Chapter 9, "SNA/MS Encodings" on page 9-1, and/or one or more of the following GDS variables if appropriate: X'1532' SNA Condition Report: documented in Appendix C, "Common Structures." Present if an SNA-registered condition was recognized by the management services application program or SNA/DS agent at the sending node, except in the case of SNA/File Services errors (when the report is contained within the FS Action Summary). X'1548' FS Action Summary: defined by SNA/File Services. Present in a management services reply MU if a server object requesting SNA/FS action was present in the management services request MU. X'1549' Agent Unit Of Work: defined by SNA/File Services. Present in a management services request MU if a Request Cancellation (X'8076') major vector refers to another request MU, using its correlation value as its identifier. X'154D' Routing and Targeting Instructions: defined by SNA/Management Services. Present in a management services request MU if it requires further routing information, or if it refers to a particular agent or application name. X'1323' Context-Identified Values: defined by SNA/Management Services. Present in a management services MU if implementation-unique data is to be transported as a set of values identified in a specific implementation-defined context. *Note:* For some conditions (for example, parsing errors where the command is not recognized, or SNA/File Services errors that occur prior to MS command execution), the major vector may be omitted.

Compare States (X'1213') GDS Variable

Compare States (X'1213') GDS Variable

Byte	Bit	Content
0 – 1		Length (q + 1 or r + 1), in binary, of Compare States GDS variable, including this Length field
2 – 3		GDS ID: X'1213'
4		Service Flags
	0 – 3	Reserved
	4 – 7	Request/reply indicator: 0010 request resync 1000 reply to resync, function completed abnormally: The partner's specified LUW state was a legal value, but the partner's state along with the local state do not comprise a legal combination. 1001 reply to resync, function completed normally
5		Sync point manager's LUW state: X'01' RESET X'02' SYNC_POINT_MANAGER_PENDING X'03' IN_DOUBT X'04' COMMITTED X'05' HEURISTIC_RESET X'06' HEURISTIC_COMMITTED X'07' HEURISTIC_MIXED
6		Flag byte
	0	Reserved
	1	Resync processing status (reserved when sent from initiator): 0 resync not in progress: Byte 5 reflects the state of the entire subtree headed by the sender of this Compare States. 1 resync in progress: The state of one or more agents of the sender of this Compare States is unknown.
	2 – 7	Reserved
7		Length, in binary, of Logical-Unit-of-Work Identifier field (values 10 to 26 are valid)
8 – n		Logical-Unit-of-Work Identifier
8		Length, in binary, of network-qualified LU name (values 1 to 17 are valid)
8 – w		Network-qualified LU network name
w + 1 – w + 6		Logical-unit-of-work instance number, in binary
w + 7 – w + 8(= n)		Logical-unit-of-work sequence number, in binary
n + 1		Length (values 1 to 8 are valid), in binary, of conversation correlator
n + 2 – p		Conversation correlator of the transaction program that allocated the conversation that failed (see FMH-5 for the format of this correlator)

Compare States (X'1213') GDS Variable

Byte	Bit	Content
p+1		Length (values 2 to 8 are valid), of session-instance identifier
p+2−q		Session-instance identifier of session being used by the conversation at the time of failure (see Chapter 7, "User Data Structured Subfields" on page 7-1 for the format of this identifier)
q+1		Length (values 0 to 17 are valid), in binary, of the network-qualified name of the LU that created the conversation correlator carried in byte n+2.
q+2−r		Network-qualified name of the LU that created the conversation correlator
Note:		The network-qualified name of the conversation correlator creator is omitted if non-support of it is negotiated during the Exchange Log Name exchange. If the field is omitted, the length field (byte q+1) is also omitted.

Sign-Off (X'1220') GDS Variable

The Sign-Off (X'1220') GDS variable flows if an active session exists to the partner LU whenever:

- A user's entry has been removed from the sender's signed-on-from list.
- A sign-on Attach fails to sign on the user.
- A signed-on Attach fails because the user is not in the sender's signed-on-from list and was not already verified.

Sign-Off (X'1220') GDS Variable

Byte	Bit	Content
0–1		Length (n + 1), in binary, of the GDS variable, including the Length field.
2–3		GDS ID: X'1220'
4–n		Zero to two Sign-Off subfields, each of which has the following format:
0		Length (values 1 – 11), in binary, of remainder of subfield, not including this byte
1		Subfield type: X'00' profile X'01' reserved X'02' user ID
2–i		Profile name or user ID, depending on the subfield type: a 1- to 10-byte symbol string of a type (A, AE, GR, DB, 1134) constrained by the receiver

Notes:

1. Only one of each subfield type is included.
2. If the subfield type is X'00' (profile) and no profile name follows, only the list entries *lacking* a profile for the specified user ID are to be removed. Omission of the entire profile subfield (length and X'00' subfield type as well as the profile name) means *all* entries for the specified user ID, *regardless of* profile, are to be removed.
3. Omission of the full user ID subfield (length, X'02' subfield type, and user ID) implies *all* user IDs in the receiving LU's signed-on-to list that are signed on to the sending LU are to be removed.

Sign-On (X'1221') GDS Variable

The Sign-On (X'1221') GDS variable is used to convey user ID, password, and optional profile to a sign-on server, and to request a sign-on, or a sign-on with change password, or to reply to these requests by the server.

Sign-On (X'1221') GDS Variable

Byte	Bit	Content
0–1		Length (n+1), in binary, of the GDS variable, including this length field
2–3		GDS ID: X'1221'
4–n		GDS variable data: one of the following context-dependent GDS structured fields: **X'FF00'** Sign-On Request Data **X'FF01'** Sign-On/Change-Password Request Data **X'FF02'** Sign-On Reply Data The first two flow requester to server; the third flows server to requester. Each of these structured fields is described in zero-origin form below.

Sign-On Request Data (X'FF00') GDS Structured Field

Sign-On Request Data (X'FF00') GDS Structured Field

Byte	Bit	Content
0–1		Length, in binary, of this GDS structured field, including this length field
2–3		GDS ID: X'FF00'
4–n		The following subfields (order unspecified): **X'00'** Profile subfield (optional) **X'01'** User ID subfield (always present) **X'02'** Password subfield (always present) The formats of these subfields are shown following the Sign-On/Change-Password GDS structured field.

Sign-On/Change-Password Request Data (X'FF01') GDS Structured Field

Sign-On/Change-Password Request Data (X'FF01') GDS Structured Field

Byte	Bit	Content
0–1		Length, in binary, of this GDS structured field, including this length field
2–3		GDS ID: X'FF01'
4–n		The following subfields (order unspecified): **X'00'** Profile subfield (optional) **X'01'** User ID subfield (always present) **X'02'** Password subfield (always present) **X'06'** New Password subfield (always present) The formats of these subfields are shown below.

Profile (X'00') Subfield

Profile (X'00') Subfield

Byte	Bit	Content
0		Length (m + 1), in binary, of this subfield
1		Type: X'00'
2–m		Profile: a 1- to 10-byte symbol string of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

User ID (X'01') Subfield

User ID (X'01') Subfield

Byte	Bit	Content
0		Length (m + 1), in binary, of this subfield
1		Type: X'01'
2–m		User ID: a 1- to 10-byte symbol string of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

Password (X'02') Subfield

Password (X'02') Subfield

Byte	Bit	Content
0		Length (m + 1), in binary, of this subfield
1		Type: X'02'
2–m		Password: a 1- to 10-byte symbol string (in the clear) of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

New Password (X'06') Subfield

New Password (X'06') Subfield

Byte	Bit	Content
0		Length (m + 1), in binary, of this subfield
1		Type: X'06'
2–m		Password: a 1- to 10-byte symbol string (in the clear) of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

Sign-On Reply Data (X'FF02') GDS Structured Field

Sign-On Reply Data (X'FF02') GDS Structured Field

Byte	Bit	Content
0–1		Length, in binary, of this GDS structured field, including this length field
2–3		GDS ID: X'FF02'
4–n		One or more of the following subfields (order unspecified):
		X'00' Sign-On Completion Status subfield (always present)
		X'01' Sign-On Request Formatting Error subfield (present only when completion status [in the X'00' subfield] is set to X'06' [incorrect data format])
		X'02' Date/Time of Current Successful Sign-On subfield (optionally present only when completion status [in the X'00' subfield] is set to X'00')
		X'03' Date/Time of Last Successful Sign-On subfield (optionally present only when completion status [in the X'00' subfield] is set to X'00')
		X'04' Date/Time That Password Will Expire subfield (optionally present only when completion status [in the X'00' subfield] is set to X'00')
		X'05' Number of Unsuccessful Sign-On Requests subfield (optionally present only when completion status [in the X'00' subfield] is set to X'00')
		The formats of these subfields are shown below.

Sign-On Completion Status (X'00') Subfield

Sign-On Completion Status (X'00') Subfield

Byte	Bit	Content
0		Length (3), in binary, of this subfield
1		Type: X'00'

Sign-On Completion Status (X'00') Subfield

Byte	Bit	Content	
2		Completion Status:	
		X'00'	Successful completion: • user ID valid • optional profile valid • password valid • password not expired unless new password specified • new password valid, if specified and therefore set • persistent verification processing complete, if supported
		X'01'	user ID unknown
		X'02'	user ID valid, password incorrect
		X'03'	user ID valid, password correct but expired, requiring new password be sent
		X'04'	user ID valid, password correct, new password not acceptable to receiving security system
		X'05'	security function failure, function not performed
		X'06'	incorrect data format, subfield X'01' provides additional error information
		X'07'	general security error: user ID unknown or password or optional profile incorrect
		X'08'	password changed completed, but persistent verification sign-on failed

Sign-On Request Formatting Error (X'01') Subfield

Sign-On Request Formatting Error (X'01') Subfield

Byte	Bit	Content
0		Length (4), in binary, of this subfield
1		Type: X'01'
2–3		Error code: one of the values also defined for use in bytes 2-3 of sense code X'100B'

Date/Time (X'02', X'03', X'04') Subfields

Date/Time (X'02', X'03', X'04') Subfields

Byte	Bit	Content
0		Length (10), in binary, of this subfield
1		Type: **X'02'** date/time of current successful Sign-On **X'03'** date/time of last successful Sign-On **X'04'** date/time that password will expire
2–9		Date/Time Fields (Values formatted in hex)
2–3		Year (Example: 1989 = X'07C5')
4		Month (Example: January = X'01')
5		Day (Example: First day = X'01'; 31st day = X'1F')
6		Hour (Example: Midnight = X'01'; 23rd hour = X'17')
7		Minute (Example: On the hour = X'00'; 59th minute = X'3B')
8		Second (Example: On the minute = X'00'; 59th second = X'3B')
9		One-hundredth of a second (Example: On the second = X'00'; maximum = X'63') **Note:** On a given day, the maximum time is 23 hours, 59 minutes, and 59.99 seconds. (Midnight is zero hours, zero minutes, and zero seconds on the following day.)

Number of Unsuccessful Sign-On Requests (X'05') Subfield

Number of Unsuccessful Sign-On Requests (X'05') Subfield

Byte	Bit	Content
0		Length (4), in binary, of this subfield
1		Type: X'05'
2–3		Number, in binary, of unsuccessful Sign-On requests since the last successful one

CP Capabilities (X'12C1') GDS Variable

The CP Capabilities GDS variable is exchanged by the CP capabilities (CP_CAP_SON_TP) service transaction programs to describe the control point capabilities once a CP-CP session is activated.

CP Capabilities (X'12C1') GDS Variable

Byte	Bit	Content
0–1		Length (n + 1), in binary, of the GDS variable, including the Length field
2–3		Key: X'12C1'
4–n		GDS Variable Data
4–7		Flow reduction sequence number (defined only for NNCP-NNCP sessions; otherwise, reserved): a monotonically increasing value (within the wrap window) that identifies the latest CP Capabilities or Topology Database Update GDS variable received by the sender of this CP Capabilities GDS variable
8–11		Support indicators (bit is set to 1 if the sender supports the function):
	0–2	Retired (set to 100 by ENCPs; set to 111 by NNCPs)
	3–4	Reserved
	5	TOPOLOGY_DATABASE_UPDATE service transaction program supported: the sending CP supports receipt of topology database updates on this session (always 1 between NNCPs; otherwise, reserved)
	6	MS capabilities exchange supported: the sending CP supports the MS capabilities requests and replies.
	7	Reserved
	8	Retired (set to 1)
	9–12	Reserved
	13	This node (whether an EN or an NN) supports EN SS-initiated NN server selection and (if an NN) supports receiving, from another NN, a BIND without an RSCV.
	14–31	Reserved
12–n		Control vectors, as described in "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT. X'33' ENCP Search Control control vector (optionally included only by ENCPs)

Topology Database Update (X'12C2') GDS Variable

A Topology Database Update (TDU) GDS variable is used to transport topology data between APPN network nodes.

Each Topology Database Update GDS variable has a maximum length of 1024 bytes.

Topology Database Update (X'12C2') GDS Variable

Byte	Bit	Content
0–1		Length (n + 1, ≤ 1024), in binary, of the GDS variable, including the Length field
2–3		Key: X'12C2'
4–n		GDS Variable Data Control vectors, as described in "Control Vectors" on page 8-3 *Note:* The following control vectors may be included. They are parsed according to subfield parsing rule LT. X'80' Flow Reduction Sequence Number control vector (One Flow Reduction Sequence Number control vector is present regardless of the number of resources being reported; it always appears first.) X'44' Node Descriptor control vector (always present): All control vectors that follow a Node Descriptor (X'44') control vector are assumed to be associated with the node identified in the X'44' control vector until another X'44' control vector is encountered. *Note*: Multiple topology updates may be blocked within a single Topology Database Update GDS variable. Each block of updates (those associated with a single node) begins with a Node Descriptor (X'44') control vector. X'45' Node Characteristics control vector (present when node characteristics are being reported; when present, the Node Characteristics control vector immediately follows the associated Node Descriptor control vector) X'46' TG Descriptor control vector (present when TG characteristics are being reported) *Note:* The X'46' and X'47' control vectors always appear in ordered pairs. X'47' TG Characteristics control vector (present when TG characteristics are being reported)

Flow Reduction Sequence Numbers (X'80') TDU Control Vector

Successive Flow Reduction Sequence Number control vectors use monotonically increasing values to identify the ordered sending of Topology Database Update GDS variables. Each update includes flow reduction sequence numbers (FRSNs) in it. This allows a node that has become reconnected to a network to specify (on the CP Capabilities exchange) the last update that it received, or (on TDUs) to specify that gaps exist in the sequence of FRSNs sent. FRSN values begin at 1, are incremented by 1, and wrap to 0 when the end of the range is reached.

Flow Reduction Sequence Numbers (X'80') TDU Control Vector

Byte	Bit	Content
0		Length (n + 1), in binary, of Flow Reduction Sequence Number control vector
1		Key: X'80'
2–n		Subfield Data
2–5		Current FRSN of the sender: the binary value that the sender maintains locally for each update that was included in this TDU GDS variable
6–9(=n)		Last FRSN sent by the sender to allow the receiver to compute the numbering gap between the previously sent FRSN and the current one

Register Resource (X'12C3') GDS Variable

The Register Resource (X'12C3') GDS variable is used to request that one or more entries be added to a network node server's directory of network resources. The network node server returns a Register Resource (X'12C3') GDS variable to report an error; no reply is sent if the registration is completely successful.

Register Resource (X'12C3') GDS Variable

Byte	Bit	Content
0–1		Length (n + 1), in binary, of the GDS variable, including the Length field
2–3		Key: X'12C3'
4–n		GDS Variable Data
4–n		Control vectors, as described in "Control Vectors" on page 8-3 and in "Register Control Vectors" on page 13-19. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X'80' Command Parameters (X'80') Register control vector (always present, always first) **For a Register request:** X'37' Directory Entry Correlator (X'37') control vector (optionally present; paired with, and immediately preceding, a X'3C' or X'3D' control vector to provide error correlation of the reply data to the corresponding request data; if paired with a control vector that causes an error, the correlator is returned to provide a pointer to the data in error) X'3C' Associated Resource Entry (X'3C') control vector (optionally present to indicate a hierarchical relationship to the following X'3D' control vectors, e.g., the ENCP for the LUs to be registered) X'3D' Directory Entry (X'3D') control vector (present one or more times, not to exceed the length limit of 1024 bytes for the full Register GDS variable) **For a Register reply:** X'36' Directory Error (X'36') control vector (always present) X'37' Directory Entry Correlator (X'37') control vector (present when provided in the Register request; returned in a reply to indicate the directory entry on which the error occurred)

Register Control Vectors

Command Parameters (X'80') Register Control Vector

Command Parameters (X'80') Register Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'80' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2	0	Request/reply indicator: 0 Register request 1 Register (negative) reply
	1–2	Reserved
	3–4	Entry type (reserved on a Register reply): 10 register (only value defined)
	5–7	Reserved

Locate (X'12C4') GDS Variable

The Locate (X'12C4') GDS variable is used in conjunction with other GDS variables by the SEND_ and RECEIVE_NETWORK_SEARCH service transaction programs.

Locate (X'12C4') GDS Variable

Byte	Bit	Content
0–1		Length (n + 1), in binary, of the GDS variable, including the Length field
2–3		Key: X'12C4'
4		GDS Variable Data
	0	Locate chain indicator: 0 discard 1 keep
	1–3	Request-reply status (r = reserved): 00r request 01r incomplete reply (sent or received only by NNCPs) a complete reply (bits 1–3 = 10r|11r) will follow 10r complete reply 11r complete reply, but eligible resources may exist that could not be located because of an outage on the search route
	4–7	Reserved
5–6		Retired
7–8		Search number (sent or received only by NNCPs; otherwise, reserved): a binary value used as a secondary key, in conjunction with the Fully Qualified PCID (X'60') control vector, to uniquely identify a search subprocedure (control block) of a Locate procedure; echoed in the search reply
9–n		Control vectors, as described in "Control Vectors" on page 8-3. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X'0E' Network Name control vector: name of the destination control point (present in a request when the destination network name is known) X'2B' Route Selection control vector (present on a directed Locate search request exchanged between NNCPs to specify the CPs along a directed Locate procedure path) X'35' Extended Sense Data control vector (present on a reply to indicate a Locate error) X'60' Fully Qualified PCID control vector (always present) X'80' Search Scope control vector (present between NNCPs to define the scope of a broadcast search request) X'82' Intersubnetwork Search control vector (present when the Locate is being sent over an intersubnetwork TG)

Locate Control Vectors

Search Scope (X'80') Locate Control Vector

Search Scope (X'80') Locate Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'80' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2–n		Vector Data
2		Hop count: a binary value specifying the number of hops that may be traversed for a broadcast search (set by the broadcast origin CP and decremented, on the search request, by intermediate CPs participating in the broadcast search)

Intersubnetwork Search (X'82') Locate Control Vector

Intersubnetwork Search (X'82') Locate Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X'82' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2–n		Vector Data
2		Subnetwork controls:
	0	Search scope indicator: 0 Search may span subnetwork boundaries. 1 Search may not span subnetwork boundaries.
	1	Search boundary-crossing indicator: 0 Search has not crossed a subnetwork boundary. 1 Search has crossed a subnetwork boundary.
	2–7	Reserved

Cross-Domain Initiate (X'12C5') GDS Variable

The CD-Initiate GDS variable is used in conjunction with the Locate, Find, and Found GDS variables to provide cross-domain session-initiation services.

Cross-Domain Initiate (X'12C5') GDS Variable

Byte	Bit	Content
0–1		Length (r+1), in binary, of the GDS variable, including the Length field
2–3		Key: X'12C5'
4–r		GDS Variable Data
4	0–3	Format: X'0' format 0 (only value defined)
	4–7	Reserved
5		Reserved
6–8		Initiate Parameters
6	0–6	Reserved
	7	Session polarity: 0 DLU is PLU 1 OLU is PLU (only value currently used)
7–9		Reserved
10		Length, in binary, of mode name
11–m		Mode name: 0 to 8 type-1134 symbol-string characters with optional (but not significant) trailing space (X'40') characters
m+1–m+2		Reserved

Cross-Domain Initiate (X'12C5') GDS Variable

Byte	Bit	Content
m+3−n		Control vectors, as described in "Control Vectors" on page 8-3. *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT. X'2B' Route Selection control vector (present in a reply CD-Initiate sent from the NNCP(OLU) to the ENCP(OLU) when a Locate search was successful) X'2C' COS/TPF control vector (generated by an ENCP(OLU) if it supports mode-to-COS mapping — otherwise by its NN server; returned to the ENCP(OLU) by its NN server on a Locate search reply for use in a subsequent BIND): when present, the TPF field is reserved unless a Route Selection (X'2B') control vector is also present X'46' TG Descriptor control vector (generated by APPN end nodes, or on replies by APPN network nodes for destination client LEN end nodes, but not received by end nodes; appears in ordered pairs with the TG Characteristics [X'47'] control vector; an EN(OLU) generates one pair for each active TG to a network node or a connection network; an EN(DLU) does the same, but also includes one pair for each active TG to the EN(OLU); in any case, the number of pairs cannot violate the limit of 1024 bytes on the total Locate search message) X'47' TG Characteristics control vector (appears in ordered pairs with the TG Descriptor [X'46'] control vector)

Delete Resource (X'12C9') GDS Variable

The Delete Resource (X'12C9') GDS variable requests a network node server to delete one or more entries from its directory. The network node server returns a Delete Resource (X'12C9') GDS variable reply to report an error; no reply is sent if the deletion is completely successful.

Delete Resource (X'12C9') GDS Variable

Byte	Bit	Content
0−1		Length (n+1), in binary, of the GDS variable, including the Length field
2−3		Key: X'12C9'
4−n		GDS Variable Data

Delete Resource (X'12C9') GDS Variable

Byte	Bit	Content
4−n		Control vectors, as described in "Control Vectors" on page 8-3 and in "Delete Control Vectors" on page 13-24. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X'80' Command Parameters (X'80') Delete control vector (always present, always first) **For a Delete request**: X'37' Directory Entry Correlator (X'37') control vector (optionally present; paired with, and immediately preceding, a X'3C'or X'3D', control vector to provide error correlation of the reply data to the corresponding request data; if paired with a control vector that causes an error, the correlator is returned to provide a pointer to the data in error) X'3C' Associated Resource Entry (X'3C') control vector (optionally present to indicate a hierarchical relationship to the following X'3D' control vectors, e.g., the ENCP for the LUs to be deleted) X'3D' Directory Entry (X'3D') control vector (present one or more times, not to exceed the length limit of 1024 bytes for the full Delete GDS variable **For a Delete reply**: X'36' Directory Error (X'36') control vector (always present) X'37' Directory Entry Correlator (X'37') control vector (present when provided in the Delete request; returned in a reply to indicate the directory entry on which the error occurred)

Delete Control Vectors

Command Parameters (X'80') Delete Control Vector

Command Parameters (X'80') Delete Control Vector

Byte	Bit	Content
0−1		Vector header; Key = X'80' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2	0	Request/reply indicator: 0 Delete request 1 Delete (negative) reply
	1	Delete directory entry condition (reserved on a Delete reply): 1 delete a directory entry unconditionally and any subordinate directory entries (only value defined)
	2−7	Reserved

Find Resource (X'12CA') GDS Variable

The Find Resource (X'12CA') GDS variable is used to request a node to search its directory for the search arguments provided.

Find Resource (X'12CA') GDS Variable

Byte	Bit	Content
0–1		Length (n+1), in binary, of the GDS variable, including the Length field
2–3		Key: X'12CA'
4–n		GDS Variable Data
4–n		Control vectors, as described in "Control Vectors" on page 8-3 and in "Find Control Vectors" on page 13-26. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X'80' Command Parameters (X'80') Find control vector (always present, always first) X'3C' Associated Resource Entry (X'3C') control vector: used to identify the search origin end node CP or network node server CP information to be saved at the search destination (e.g., in the server's directory as a cache entry); hierarchical associations are indicated by the order of X'3C' and X'3D' control vectors, those appearing first being hierarchically above those that follow (an EN(OLU) generates one for its own CP; an EN(DLU) receives one for the OLU's network node server CP and, if the OLU resides in an end node, one for the OLU's ENCP) X'3D' Directory Entry (X'3D') control vector: provides information about the search origin (always present) X'40' Real Associated Resource Entry (X'40') control vector: used to identify the name of the real associated resource of the resource identified in the Directory Entry (X'3D') control vector (present only when an Associated Resource Entry (X'3C') control vector in the hierarchy does not represent the real hierarchical superior of the target resource but rather is a surrogate); multiple Real Associated Resource (X'40') control vectors are arranged in hierarchical order, with an earlier-appearing one being hierarchically above one that follows. X'82' Search Argument Directory Entry (X'82') Find control vector: used to specify the search argument directory entry (always present)

Find Control Vectors

Command Parameters (X'80') Find Control Vector

Command Parameters (X'80') Find Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'80' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2(=m)	0	Origin information present indicator: 1 present (only value defined)
	1–7	Reserved

Search Argument Directory Entry (X'82') Find Control Vector

Search Argument Directory Entry (X'82') Find Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X'82' (see "Substructure Encoding/Parsing Rules" on page 8-1)
2–m		Vector Data
2–3		Resource type: X'00F3' logical unit X'00F4' ENCP X'00F6' NNCP
4–m		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted) *Note:* The network ID is always present when different from the network ID of the receiver.

Found Resource (X'12CB') GDS Variable

The Found Resource (X'12CB') GDS variable is a positive reply to a Find Resource (X'12CA') GDS variable; it provides the requested data.

Found Resource (X'12CB') GDS Variable

Byte	Bit	Content
0–1		Length (n + 1), in binary, of the GDS variable, including the Length field
2–3		Key: X'12CB'
4–n		GDS Variable Data
4–n		Control vectors, as described in "Control Vectors" on page 8-3 and in "Found Control Vectors" on page 13-28. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X'80' Command Parameters (X'80') Found control vector (always present, always first) X'3C' Associated Resource Entry (X'3C') control vector: used to identify the search destination end node CP or network node server CP to be saved at the search origin (e.g., in the server's directory as a cache entry); hierarchical associations are indicated by the order of X'3C' and X'3D' control vectors, those appearing first being hierarchically above those that follow (an EN(DLU) generates one for its own CP; an EN(OLU) receives one for the DLU's network node server CP and, if the DLU resides in an end node, one for the DLU's ENCP) target resource, and requested by a Return Associated Resource (X'84') Find control vector on the Find Resource (X'12CA') GDS variable or by default if X'84' control vectors are not present). X'3D' Directory Entry (X'3D') control vector: identifies the requested destination directory entry (always present) X'40' Real Associated Resource Entry (X'40') control vector: used to identify the name of the real associated resource of the resource identified in the Directory Entry (X'3D') control vector (present only when an Associated Resource Entry (X'3C') control vector in the hierarchy does not represent the real hierarchical superior of the target resource but rather is a surrogate); multiple Real Associated Resource (X'40') control vectors are arranged in hierarchical order, with an earlier-appearing one being hierarchically above one that follows.

Found Control Vectors

Command Parameters (X'80') Found Control Vector

Command Parameters (X'80') Found Control Vector

Byte	Bit	Content
0 – 1		Vector header; Key = X'80' (see "Substructure Encoding/Parsing Rules" on page 8-1
2(=m)	0	Target information present indicator: 1 present (only value defined)
	1 – 6	Reserved
	7	Wild-card directory entry: 0 The directory entry for this located resource is an explicit or partially specified name. 1 The directory entry for this located resource is a wild-card entry.

Multiple-Domain Support Message Unit (X'1310') GDS Variable

Multiple-Domain Support Message Unit (MDS-MU) transports routing and control information and data for management services application programs.

Multiple-Domain Support Message Unit (X'1310') GDS Variable

Byte	Bit	Content
0–1		Length (m + 1), in binary, of the MDS-MU
2–3		GDS ID: X'1310'
4–m		The following general data stream (GDS) variables as indicated:

General Data Stream (GDS) Variables	Presence in MDS-MU (X'1310') GDS variable	
MDS Routing Information (X'1311')	P	Note 1
Agent Unit of Work Correlator (X'1549')	P	Note 2
SNA Condition Report (X'1532')	CP	Note 3
CP-MSU (X'1212')	CP	Note 4

Key:

P Present one time
CP Conditionally present one time (See Notes for conditions.)

Notes:

1. MDS Routing Information is always the first structure in the MDS-MU.
2. Agent Unit of Work Correlator is always the second structure in the MDS-MU. See Appendix C, "Common Structures" for the format of this GDS variable.
3. SNA Condition Report is always present if the MDS message type is X'02' (MDS error message), as indicated in byte 2 of the Flags (X'90') MDS Routing Information subvector. It is optionally present for other MDS message types (see next Note). See Appendix C, "Common Structures" for the format of this GDS variable.
4. CP-MSU is optionally present for MDS message types X'00' (MDS request) and X'01' (MDS reply). It may not be present for message type X'02' (MDS error message).

   For MDS requests and replies, a single GDS variable may be included after the Agent Unit of Work Correlator. This GDS variable, which is supplied by the origin MS application program, may be one of the following:

   a. CP-MSU

b. SNA Condition Report
c. Some other GDS variable, not currently defined by management services architecture.

Multiple-domain support considers this GDS variable to be application data, with no restrictions except the following:

a. At most, one application GDS variable may be present.
b. The length of the application GDS variable may not exceed 31743 (X'7BFF') bytes.

MDS Routing Information (X'1311') GDS Variable

Multiple-Domain Support (MDS) Routing Information contains routing and control information for the Multiple-Domain Support Message Unit (MDS-MU) containing it.

MDS Routing Information (X'1311') GDS Variable

Byte	Bit	Content
0–1		Length (m + 1), in binary, of the MDS Routing Information GDS variable
2–3		GDS ID: X'1311'
4–m		The following MS subvectors: X'81' Origin Location Name (always first) X'82' Destination Location Name (always second) X'90' Flags (always third)

Origin Location Name (X'81') MDS Routing Information Subvector

This subvector identifies the origin NAU and application program for the Multiple-Domain Support Message Unit (MDS-MU) that contains it.

Origin Location Name (X'81') MDS Routing Information Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Origin Location Name subvector
1		Key: X'81'
2–p		Three subfields containing data identifying the origin of the record, as described below. X'01' NETID (always first) X'02' NAU Name (always second) X'03' Application ID (always third)

NETID (X'01') Origin Location Name Subfield

This subfield contains the network identifier portion of the network-qualified name of the NAU that originated the management services record containing it.

NETID (X'01') Origin Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the NETID subfield
1		Key: X'01'
2−q		NETID: a 1- to 8-byte type-1134 symbol string name; trailing space (X'40') characters may be present, but are insignificant; leading or embedded space characters are not permitted.

NAU Name (X'02') Origin Location Name Subfield

This subfield contains the unqualified name of the NAU that originated the management services record containing it. This is a CP or LU name.

NAU Name (X'02') Origin Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the NAU Name subfield
1		Key: X'02'
2−q		NAU name: a 1- to 8-byte type-1134 symbol string name; trailing space (X'40') characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Application ID (X'03') Origin Location Name Subfield

This subfield contains either a 4-byte application program name defined by the management services architecture or a 1- to 8-byte installation-defined name.

Application ID (X'03') Origin Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Application Identification subfield
1		Key: X'03'
2−q		Application identification: Either a 1- to 8-byte type-1134 symbol string name, or one of the 4-byte architecturally defined values for management services application programs, listed in *SNA/Management Services Reference.* Trailing space (X'40') characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Destination Location Name (X'82') MDS Routing Information Subvector

This subvector identifies the destination NAU and application program for the Multiple-Domain Support Message Unit (MDS-MU) that contains it.

Destination Location Name (X'82') MDS Routing Information Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Destination Location Name subvector
1		Key: X'82'
2–p		Three subfields containing data identifying the destination of the record, as described below. X'01' NETID (always first) X'02' NAU Name (always second) X'03' Application ID (always third)

NETID (X'01') Destination Location Name Subfield

This subfield contains the network identifier portion of the network-qualified name of the NAU to which the management services record containing it is being sent.

NETID (X'01') Destination Location Name Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the NETID subfield
1		Key: X'01'
2–q		NETID: a 1- to 8-byte type-1134 symbol string name; trailing space (X'40') characters may be present, but are insignificant; leading or embedded space characters are not permitted.

NAU Name (X'02') Destination Location Name Subfield

This subfield contains the unqualified name of the NAU to which the management services record containing it is being sent. This is a CP or LU name.

NAU Name (X'02') Destination Location Name Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the NAU Name subfield
1		Key: X'02'
2–q		NAU name: a 1- to 8-byte type-1134 symbol string name; trailing space (X'40') characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Application ID (X'03') Destination Location Name Subfield

This subfield contains either a 4-byte application program name defined by the management services architecture or a 1- to 8-byte installation-defined name.

Application ID (X'03') Destination Location Name Subfield

Byte	Bit	Content
0		Length (q + 1), in binary, of the Application Identification subfield
1		Key: X'03'
2–q		Application identification: Either a 1- to 8-byte type-1134 symbol string name, or one of the 4-byte architecturally defined values for management services application programs, listed in *SNA/Management Services Reference.* Trailing space (X'40') characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Flags (X'90') MDS Routing Information Subvector

This subvector contains various flags related to the transport of data between management services application programs.

Flags (X'90') MDS Routing Information Subvector

Byte	Bit	Content
0		Length (p + 1), in binary, of the Flags subvector
1		Key: X'90'

Flags (X'90') MDS Routing Information Subvector

Byte	Bit	Content
2		MDS message type: X'00' MDS request X'01' MDS reply X'02' MDS error message
3 – 4(=p)		Flags:
	0	First MDS message indicator: 0 MDS message is not the first message for the current unit of work 1 MDS message is the first message for the current unit of work. This value is required for an MDS error message. If the last MDS message indicator is also 1, then the message is the only one for the current unit of work.
	1	Last MDS message indicator: 0 MDS message is not the last message for the current unit of work 1 MDS message is the last (or only) message for the current unit of work. This value is required for an MDS error message.
	2 – 15	Reserved

Chapter 14. SNA/DS FS1 Encodings

Introduction

This chapter contains the format descriptions of the FS1 message units. The format descriptions are comprised of two parts: *header description tables* and *structure descriptions*. A header description table contains the header information for each structure associated with a particular message unit. A structure description contains a prose description of the structure, bit-level representations, any presence rules or length restrictions associated with a particular structure, and any special notes required to understand the differences between FS1 and FS2 encodings.

The definition of SNA/Distribution Services (SNA/DS) requires a byte-accurate description of the formats that must be understood by all DSUs. The SNA/DS formats are described in terms of encoded fields referred to as "structures" and the hierarchical relationship between these structures. In this chapter, the header description tables show each structure and its header.

Refer to Appendix C, "Common Structures" on page C-1 for a complete definition and classification of the encoding structures used in the following tables.

Header Description Tables for FS1 Message Units

DISTRIBUTION MESSAGE UNIT (DIST_MU)

Figure 14-1 (Page 1 of 2). Distribution Message Unit (DIST_MU)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_MU	14-6	Del-IDF	pfx	≥148	1	N	Y	3-4	—
Prefix	14-6	IDF/pfx	C00102	5-21	1	—	—	—	—
Dist_Command	14-6	IDF/seg	C10502	138-32511	1	N	Y	2-3	—
Service_Desc_Operands	14-6	Imp-IDF	idc	58-774	1	N	N	2-5	—
Dist_ID	14-6	IDF/idc	C34041	28-107	1	N	N	5-7	—
Origin_RGN	14-6	T	01	3-10	0-1	—	—	—	—
Origin_REN	14-7	T	02	3-10	1	—	—	—	—
Origin_DGN	14-7	T	03	2-10	1	—	—	—	—
Origin_DEN	14-7	T	04	2-10	1	—	—	—	—
Origin_Seqno	14-8	T	05	6	1	—	—	—	—
Origin_DTM	14-8	T	06	10	1	—	—	—	—
Agent_Correl	14-9	T	07	3-46	0-1	—	—	—	—
Dist_Gen_Options	14-9	IDF	C33D41	30-58	1	N	N	5	—
Dist_Flags (FS1)	14-9	T	01	3	1	—	—	—	—
Hop_Count	14-9	T	02	4	1	—	—	—	—
Service_Parms	14-10	T	03	11-32	1	—	—	—	—
Server_Object_Ind	14-13	T	04	4	1	—	—	—	—
Origin_Agent	14-13	T	05	3-10	1	—	—	—	—
Report-To_Address	14-13	IDF	C36041	14-45	0-1*	N	N	3-4	—
Report-To_RGN	14-13	T	01	3-10	0-1	—	—	—	—
Report-To_REN	14-14	T	02	3-10	1	—	—	—	—
Report-To_DGN	14-14	T	03	3-10	1	—	—	—	—
Report-To_DEN	14-15	T	04	3-10	1	—	—	—	—
Report-To_Options	14-15	IDF	C34341	8-47	0-1*	N	N	1-2	—
Report_Service_Parms	14-16	T	01	11-32	0-1	—	—	—	—
Report-To_Agent	14-19	T	02	3-10	0-1	—	—	—	—
Agent_Object	14-19	IDF	C32D01	6-517	0-1	—	—	—	—
Destination_Operands	14-19	Imp-IDF	idc	≥75	1	N	Y	3	—
Begin_Dest_Operands	14-21	IDF/idc	C35001	8	1	—	—	—	—
Dest_RGN_List	14-21	Imp-IDF	idc	≥62	≥1	N	Y	4	—
Dest_RGN	14-21	IDF/idc	C35201	5-13	1	—	—	—	—
Begin_REN_List	14-21	IDF	C35001	8	1	—	—	—	—
Dest_REN_List	14-21	Imp-IDF	idc	≥44	≥1	N	Y	4	—
Dest_REN	15-23	IDF/idc	C35301	6-13	1	—	—	—	—
Begin_DGN_List	14-22	IDF	C35001	8	1	—	—	—	—
Dest_DGN_List	14-22	Del-IDF	pfx	≥25	≥1	N	Y	4	—

Figure 14-1 (Page 2 of 2). Distribution Message Unit (DIST_MU)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dest_DGN	15-23	IDF/pfx	C35401	6-13	1	—	—	—	—
Begin_DEN_List	14-22	IDF	C35001	8	1	—	—	—	—
Dest_DEN	14-23	IDF	C35501	6-13	≥1	—	—	—	—
End_DEN_List	14-23	IDF/sfx	C35101	5	1	—	—	—	—
End_DGN_List	14-23	IDF	C35101	5	1	—	—	—	—
End_REN_List	14-23	IDF	C35101	5	1	—	—	—	—
End_Dest_Operands	14-23	IDF	C35101	5	1	—	—	—	—
Dist_Report_Operands	14-25	Imp-IDF	idc	≥63	0-1*	N	Y	2-4	14-4
Dist_Server_Operands	14-23	Imp-IDF	idc	≥14	0-1*	N	Y	2	—
Server_Prefix	14-23	IDF/idc	C90A41	8-280	1	N	N	1-3	—
Server_Obj_Byte_Count	14-24	T	01	10	0-1	—	—	—	—
Server	14-24	T	02	3-10	1	—	—	—	—
Server_Parms	14-24	T	03	3-255	0-1	—	—	—	—
Server_Object	14-24	IDF/seg	C90801	≥6*	1	—	—	—	—
DS_Suffix (FS1)	14-25	IDF	CF0100	5	1	—	—	—	—

Notes:

- * Refer to FS1 Structure Descriptions starting on page 14-6 for presence rules and length restrictions.
- *Dist_Report_Operands* does not occur for Dist_MU *type* TRANSPORT.
- *Agent_Correl*, *Report-To_Address*, *Report-To_Options*, *Agent_Object*, and *Dist_Server_Operands* do not occur for Dist_MU *type* REPORT.
- *Dest_RGN_List*, *Dest_REN_List*, *Dest_DGN_List*, and *Dest_DEN* occur only one time for Dist_MU *type* REPORT.

DIST REPORT OPERANDS

Figure 14-2. Distribution Report Operands

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Report_Operands	14-25	Imp-IDF	idc	≥63	0-1	N	Y	2-4	—
Report_Operands	14-25	Imp-IDF	idc	27-112	1	N	N	1-2	—
Report_Correlation	14-25	IDF/idc	C34041	27-87	1	N	N	4-5	—
Reported-On_Origin_DGN	14-25	T	03	3-10	1	—	—	—	—
Reported-On_Origin_DEN	14-26	T	04	3-10	1	—	—	—	—
Reported-On_Seqno	14-26	T	05	6	1	—	—	—	—
Reported-On_DTM	14-27	T	06	10	1	—	—	—	—
Reported-On_Agent_Correl	14-27	T	07	3-46	0-1	—	—	—	—
Receiving_DSU	14-27	IDF	C36141	8-25	0-1	N	N	1-2	—
Receiving_RGN	14-28	T	01	3-10	0-1	—	—	—	—
Receiving_REN	14-28	T	02	3-10	1	—	—	—	—
Gen_SNADS_Report	14-28	Imp-IDF	idc	16	0-1*	N	Y	2	—
Gen_SNADS_Type	14-29	IDF/idc	C35601	7	1	—	—	—	—
Gen_SNADS_Contents	14-29	IDF	C35741	9	1	N	Y	1	—
Gen_SNADS_Cond_Code	14-29	T	01	4	1	—	—	—	—
Gen_DIA_Report	14-30	Imp-IDF	idc	14-524	0-1*	N	Y	2	—
Gen_DIA_Type	14-30	IDF/idc	C35601	7	1	—	—	—	—
Gen_DIA_Contents	14-30	IDF	C35741	7-517*	1	—	—	—	—
Specific_Report	14-30	Imp-IDF	idc	≥36	1	N	Y	3	—
Begin_Report_DGN_List	14-30	IDF/idc	C35001	8	1	—	—	—	—
Report_DGN_List	14-31	Imp-IDF	idc	≥23	≥1	N	Y	4	—
Reported-On_Dest_DGN	14-31	IDF/idc	C35401	5-13	1	—	—	—	—
Begin_Report_DEN_List	14-31	IDF	C35001	8	1	—	—	—	—
Report_DEN_List	14-31	Imp-IDF	idc	5-553	≥1	N	Y	1-3	—
Reported-On_Dest_DEN	14-32	IDF/idc	C35501	5-13	1	—	—	—	—
Spec_SNADS_Report	14-32	Imp-IDF	idc	16	0-1*	N	Y	2	—
Spec_SNADS_Type	14-32	IDF/idc	C35601	7	1	—	—	—	—
Spec_SNADS_Cont	14-32	IDF	C35741	9	1	N	Y	1	—
Spec_SNADS_CC	14-33	T	01	4	1	—	—	—	—
Spec_DIA_Report	14-33	Imp-IDF	idc	14-524	0-1*	N	Y	2	—
Spec_DIA_Type	14-34	IDF/idc	C35601	7	1	—	—	—	—
Spec_DIA_Contents	14-34	IDF	C35741	7-517*	1	—	—	—	—
End_Report_DEN_List	14-34	IDF	C35101	5	1	—	—	—	—
End_Report_DGN_List	14-34	IDF	C35101	5	1	—	—	—	—

Note: * Refer to FS1 Structure Descriptions starting on page 14-6 for presence rules and length restrictions.

SENDER EXCEPTION MESSAGE UNIT (TYPE FS1)

Figure 14-3. Sender Exception Message Unit (type FS1)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Sender_Exception_MU (FS1)	14-35	IDF	CF0201	8	1	—	—	—	—

RECEIVER EXCEPTION MESSAGE UNIT (TYPE FS1)

Figure 14-4. Receiver Exception Message Unit (type FS1)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Receiver_Exception_MU (FS1)	14-35	Del-IDF	pfx	59-863	1	N	Y	3	—
Prefix	14-6	IDF/pfx	C00102	5	1	—	—	—	—
Receiver_Exception_Command	14-35	IDF	C10101	49-853	1	N	Y	2	—
Receiver_Exception_Correl	14-36	IDF	C32801	7-23	1	—	—	—	—
Exception_And_Reply_Data	14-36	Imp-IDF	idc	37-825	1	N	N	2	—
Receiver_Exception_Code	14-37	IDF/idc	C32201	8-255	1	—	—	—	—
Reply_Data	14-38	IDF	C34501	29-570	1	N	Y	2-3	—
Receiving_DSU	14-27	IDF	C36141	8-25	1	N	N	1-2	—
Receiving_RGN	14-28	T	01	3-10	0-1	—	—	—	—
Receiving_REN	14-28	T	02	3-10	1	—	—	—	—
SNADS_Report	14-38	Imp-IDF	idc	16	1	N	Y	2	—
SNADS_Report_Type	14-38	IDF/idc	C35601	7	1	—	—	—	—
SNADS_Report_Cont	14-38	IDF	C35741	9	1	N	Y	1	—
SNADS_Report_CC	14-39	T	01	4	1	—	—	—	—
DIA_Report	14-39	Imp-IDF	idc	14-524	0-1	N	Y	2	—
DIA_Report_Type	14-39	IDF/idc	C35601	7	1	—	—	—	—
DIA_Report_Cont	14-40	IDF	C35741	7-517	1	—	—	—	—
DS_Suffix (FS1)	14-25	IDF/sfx	CF0100	5	1	—	—	—	—

FS1 Structure Descriptions

Dist_MU

Description: The *distribution_message_unit* transports user information to one or more distribution service users. A Dist_MU can be one of two types based on the value of *dist_flags* (type FS1): TRANSPORT or REPORT. A Dist_MU *type* TRANSPORT transports agent and/or server objects. A Dist_MU *type* REPORT transports information reporting on the state of the distribution.

Prefix

Description: The *prefix* identifies the beginning of a message unit and may contain a message-unit identifier.

Format: Undefined byte string

Dist_Command

Description: The *distribution_command* contains all information used by each DSU to transport the distribution for a Dist_MU *type* TRANSPORT. For a Dist_MU *type* REPORT, the *distribution_command* contains the control information for the distribution report.

Service_Desc_Operands

Description: The *service_description_operands* contain all the information, except for the destination list, required by each DSU to transport the distribution.

Dist_ID

Description: The *distribution_identifier* contains information corresponding to the distribution originator.

Origin_RGN

Description: The *origin_RGN* is the first part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Origin_REN

Description: The *origin_REN* is the second part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Origin_DGN

Description: The *origin_DGN* is the first part of the user name of the distribution originator.

Note: For FS1, when the Dist_MU is of type REPORT and the distribution report was generated by SNA/DS, null user names will occur.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Origin_DEN

Description: The *origin_DEN* is the second part of the user name of the distribution originator.

Note: For FS1, when the Dist_MU is of type REPORT and the distribution report was generated by SNA/DS, null user names will occur.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Origin_Seqno

Description:	The *origin_sequence_number* is the number assigned to the distribution by the *origin_DSU*. The value ranges from 1 to 9999 for a Dist_MU *type* TRANSPORT, and is always 0 for a Dist_MU *type* REPORT.
Format:	Character string; each character is the EBCDIC representation of one digit of the sequence number.

Byte	Content
0-1	LT header
2-5	Sequence number

Notes:

- For Dist_MU *type* TRANSPORT, values range from X'F0F0F0F1' to X'F9F9F9F9'.
- For Dist_MU *type* REPORT, value is X'F0F0F0F0'.

Origin_DTM

Description:	The *origin_date-time* is the date and time the distribution was originated by the origin DSU. Time is assumed to be local.
Note:	FS1 supports neither the GMT format nor the offset time format supported by FS2.
Format:	Byte string

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X'07C5')
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)

Note: Refer to "Representing Date and Time" on page 15-38 for a complete discussion of the encoding and interpretation of date and time.

Agent_Correl

Description: The *agent_correlation* is a string supplied by the origin agent. SNA/DS is not aware of its contents.

Format: Undefined byte string

Dist_Gen_Options

Description: The *distribution_general_options* contains structures used by SNA/DS to condition its processing of the distribution.

Dist_Flags (type FS1)

Description: The *distribution_flags* indicate reporting services requested by the origin agent.

Format:

Bit	Content
0	Exception Report bit: 0 SNA/DS is requested to generate a report in case of an exception. 1 A report will not be generated by SNA/DS for this distribution.
1	Distribution Message Unit type bit: 0 Distribution is of type TRANSPORT. 1 Distribution is of type REPORT.
2-7	Reserved

Byte	Content
0-1	LT header
2	X'00' Dist_MU *type* TRANSPORT with report requested X'80' Dist_MU *type* TRANSPORT with no report requested X'C0' Dist_MU *type* REPORT with no report requested Note: All other values are reserved.

Hop_Count

Description: The *hop_count* is the remaining number of hops that may be traversed by a SNA/DS distribution on its way toward its destination DSUs. The *hop_count* is set by the origin DSU in the Dist_MUs *type* TRANSPORT and by the reporting DSUs for the Dist_MUs *type* REPORT. The *hop_count* is decremented by 1 in every DSU through which the distribution passes. If the *hop_count* reaches 0 at an intermediate DSU, exception processing is invoked.

Format: Signed binary integer (1-origin)

Range of Values: Valid values range from 0 to 2^{15}-1.

Service_Parms

Description: The *service_parameters* structure describes the types and levels of service requested for the distribution. The parameters in this structure are provided by the origin agent. In FS1, the *service_parameters* are specified by the origin agent in Dist_MU *type* TRANSPORT. The specification for deriving the *service_parameters* for Dist_MU *type* REPORT is found in the description of *report_service_parameters* on page 14-16.

Note: The differences between FS1 and FS2 service parameter support are discussed below and throughout the SP descriptions.

Format: Special format consisting of ordered SP triplets of the following general structure:

Byte	Bit	Content
0		Parameter type: All parameter type byte values < X'80' are defined by or reserved for SNA/DS. In FS1, all other parameter type byte values are reserved.
1		Comparison operator: 1100 REQUIRE_LEVEL_GE 1110 REQUIRE_SUPPORT_FOR Note: All other values for bits 0-3 are reserved.

Notes:

- FS2 supports architecturally defined SP comparison operators and values beyond those defined for FS1.
- FS2 tolerates customer-defined service parameters. Customer-defined service parameters have a parameter type byte value > X'80'.
- FS2 supports defaulted service parameters. Defaults are assigned for the SP comparison operator and SP value for each architecturally defined service parameter not specified for a given message unit.
- FS2 does not restrict the combination of SP triplets to the degree that FS1 does.

Byte	Content
0-1	LT header
2-31	Up to 10 different *service_parameter* (SP) triplets may be carried in one distribution. Each triplet appears in ascending sequence of parameter type. The architecturally defined service parameters are given below:

Note:

- Service parameters beyond the four defined below have been architecturally defined for FS2.

Priority SP Triplet

Byte	Content
0	X'01'
1	X'C0' REQUIRE_LEVEL_GE
2	X'F0' FAST X'D0' CONTROL X'80' DATA_16 (can be treated as DATAHI) X'78' DATA_15 (can be treated as DATAHI) X'70' DATA_14 (can be treated as DATAHI) X'68' DATA_13 (can be treated as DATAHI) X'60' DATA_12 (DATAHI) X'58' DATA_11 (can be treated as DATAHI) X'50' DATA_10 (can be treated as DATAHI) X'48' DATA_9 (can be treated as DATAHI) X'40' DATA_8 (can be treated as DATALO) X'38' DATA_7 (can be treated as DATALO) X'30' DATA_6 (can be treated as DATALO) X'28' DATA_5 (can be treated as DATALO) X'20' DATA_4 (DATALO) X'18' DATA_3 (can be treated as DATALO) X'10' DATA_2 (can be treated as DATALO) X'08' DATA_1 (can be treated as DATALO) Note: All other values are reserved.

Protection SP Triplet

Byte	Content
0	X'02'
1	X'C0' REQUIRE_LEVEL_GE
2	X'10' LEVEL1: safe store may be performed. X'30' LEVEL2: safe store must be performed. Note: All other values are reserved.

Capacity SP Triplet

Byte	Content
0	X'03'
1	X'C0' REQUIRE_LEVEL_GE
1	X'E0' REQUIRE_SUPPORT_FOR
2	Capacity value is the exponent of the power of 2 that represents the value of the required capacity for the *server_object* in the DTMU: X'00' ZERO X'OC' 4KB (4 kilobytes) X'FF' INDEFINITE Note: All other values are reserved.

Notes:

- The capacity requirement is for the *server_object*, and does not include the capacity needed to store and handle the other structures of the DTMU.
- FS2 implementations may accept other capacity levels (including 4KB) as long as they can route the distribution responsibly.
- Capacity SP values beyond the three defined above for FS1 have been architecturally defined for FS2 (e.g., 1MB).
- In FS2, the capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified.
- Receiving FS2 DSUs are always able to receive a capacity level of INDEFINITE; although originating FS2 DSUs never generate that capacity level. The FS2 equivalent to INDEFINITE is 16MB (X'18').

Security RSP Triplet

Byte	Content
0	X'04'
1	X'C0' REQUIRE_LEVEL_GE
2	X'01' LEVEL1: security is not required. X'20' LEVEL2: security is required. Note: All other values are reserved.

Server_Object_Ind

Description:	The *server_object_indicator* indicates whether a *server_object* is present or not. The only values supported are 0 and 1.
Presence Rule:	Contains X'0001' only for Dist_MU *type* TRANSPORT.
Format:	Hexadecimal code

Byte	Content
0-1	LT header
2-3	X'0000' no *server_object* present in this MU X'0001' a *server_object* present in this MU Note: All other values are reserved.

Origin_Agent

Description:	The *origin_agent* is the transaction program at the DSU at which the distribution originated.
Format:	Character string, except for first byte

CGCSGID	String Conventions
01130-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Report-To_Address

Description:	The *report-to_address* contains the name of the DSU and user to which any distribution reports are sent.
Presence Rule:	This information may be present only in Dist_MU *type* TRANSPORT.

Report-To_RGN

Description:	The *report-to_RGN* is the first part of the DSU name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT. This is typically, but not necessarily, the network ID.
Note:	In FS2, the *report-to_RGN* occurs in both the DTMU and DRMU.
Format:	Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_REN

Description: The *report-to_REN* is the second part of the DSU name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT. This is typically, but not necessarily, the LU name.

Note: In FS2, the *report-to_REN* occurs in both the DTMU and DRMU.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_DGN

Description: The *report-to_DGN* is the first part of the user name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT.

Note: In FS2, the *report-to_DGN* occurs in both the DTMU and DRMU.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_DEN

Description:	The *report-to_DEN* is the second part of the user name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT.
Note:	In FS2, the *report-to_DEN* occurs in both the DTMU and DRMU.
Format:	Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_Options

Description:	The *report-to_options* contains information involved in processing any reports generated as part of the distribution.
Presence Rule:	This information may be present only in Dist_MU *type* TRANSPORT.

Report_Service_Parms

Description: The *report_service_parameters* structure describes the service requested for the distribution report by the origin agent when the agent wants to override the *service_parameters* that would be routinely generated by the reporting DSU for the report MU. If *report_service_parameters* are specified, they are used as the *service_parameters* in any Dist_MU *type* REPORT that is generated as part of the distribution. If the origin agent does not specify one or more of the *report_service_parameters*, a DSU that generates a report derives appropriate *service_parameters* for the Dist_MU *type* REPORT from the *service_parameters* in the Dist_MU *type* TRANSPORT. The comparison operators and values derived for the protection, capacity, and security parameters are the same as those specified in the Dist_MU *type* TRANSPORT.

For the priority service parameter, the value derived is either FAST or CONTROL. FAST is used if the Dist_MU *type* TRANSPORT specified FAST priority; CONTROL is used if the Dist_MU *type* TRANSPORT specified a DATA_N priority. CONTROL priority is used only in Dist_MUs *type* REPORT; it may not be specified for the priority service parameter in a Dist_MU *type* TRANSPORT. If the origin agent explicitly specifies a value for the priority report service parameter, the value may be FAST, CONTROL, or DATA_N. The comparison operator for the priority service parameter is always REQUIRE_LEVEL_GE.

Notes: Following are RSP notes specific to FS2:

- For FS2, the comparison operators and values derived for the protection, security, and acceptable delay parameters are the same as those specified (explicitly or implicitly) in the DTMU. The FS2 values for the priority service parameter in the DRMU are derived using the same logic as defined above.
- FS2 tolerates customer-defined service parameters. Customer-defined service parameters have a parameter type byte value > X'80'.
- FS2 supports defaulted report service parameters. Defaults are assigned for the SP comparison operator and SP value for each architecturally defined service parameter not specified for a given message unit.
- FS2 supports SP comparison operators and SP values beyond those defined for FS1.

Format: Special format consisting of ordered, optional *report_service_parameter* triplets of the same general structure as for *service_parameters*. See *service_parameters* on page 14-10.

Byte	Content
0-1	LT header
2-31	Up to 10 different *report_service_parameter* (RSP) triplets may be carried in one distribution. Each triplet, when present, appears in ascending sequence of parameter type. The first three parameters—priority, protection, and capacity—are present if report service parameters are to be specified.

Notes:

- In FS2, all service parameters are optional in both the DTMU and DRMU.
- In FS2, the capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified. Note that the capacity RSP is specified in FS1.

Priority RSP Triplet

Byte	Content
0	X'01'
1	X'C0' REQUIRE_LEVEL_GE
2	X'F0' FAST X'D0' CONTROL X'80' DATA_16 (can be treated as DATAHI) X'78' DATA_15 (can be treated as DATAHI) X'70' DATA_14 (can be treated as DATAHI) X'68' DATA_13 (can be treated as DATAHI) X'60' DATA_12 (DATAHI) X'58' DATA_11 (can be treated as DATAHI) X'50' DATA_10 (can be treated as DATAHI) X'48' DATA_9 (can be treated as DATAHI) X'40' DATA_8 (can be treated as DATALO) X'38' DATA_7 (can be treated as DATALO) X'30' DATA_6 (can be treated as DATALO) X'28' DATA_5 (can be treated as DATALO) X'20' DATA_4 (DATALO) X'18' DATA_3 (can be treated as DATALO) X'10' DATA_2 (can be treated as DATALO) X'08' DATA_1 (can be treated as DATALO) Note: All other values are reserved.

Protection RSP Triplet

Byte	Content
0	X'02'
1	X'C0' REQUIRE_LEVEL_GE
2	X'10' LEVEL1: safe store may be performed. X'30' LEVEL2: safe store must be performed. Note: All other values are reserved.

Capacity RSP Triplet

Byte	Content
0	X'03'
1	X'C0' REQUIRE_LEVEL_GE
1	X'E0' REQUIRE_SUPPORT_FOR
2	X'00' ZERO
	Notes: All other values are reserved. Also, all FS1 implementations are able to receive distribution reports of 4KB capacity (X'0C'). New FS1 implementations always send distribution reports of ZERO capacity.

Notes:

- The capacity requirement is for the *server_object*, and does not include the capacity needed to store and handle the other structures of the DTMU.
- FS2 implementations accept other capacity levels (including 4KB) as long as they can route the distribution responsibly.
- Capacity SP values beyond the three defined above for FS1 have been architecturally defined for FS2 (e.g., 1MB).
- In FS2, the capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified.
- Receiving FS2 DSUs are always able to receive a capacity level of INDEFINITE; although originating FS2 DSUs never generate that capacity level. The FS2 equivalent to INDEFINITE is 16MB (X'18').

Security RSP Triplet

Byte	Content
0	X'04'
1	X'C0' REQUIRE_LEVEL_GE
2	X'01' LEVEL1: security is not required. X'20' LEVEL2: security is required. Note: All other values are reserved.

Report-To_Agent

Description:	The *report-to_agent* is the name of the application transaction program to be started after the report is queued for delivery. If *report-to_agent* is absent in the Dist_MU *type* TRANSPORT, the value specified in the Dist_MU *type* TRANSPORT for *origin_agent* is used in the Dist_MU *type* REPORT for *origin_agent*.
Presence Rule	This information may be present only in Dist_MU *type* TRANSPORT.
Format:	Character string, except for first byte.

CGCSGID	String Conventions
01130-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Agent_Object

Description:	The *agent_object* is directly supplied by the origin agent. It is never parsed by the distribution service and is directly delivered, unchanged, to the agent at each destination.
Format:	Undefined byte string

Destination_Operands

Description:	The *destination_operands* are the list of destinations for the distribution. Up to 256 destinations are allowed if the distribution is of type TRANSPORT; exactly one destination, if the distribution is of type REPORT. The destinations are encoded as a fully factored, partially factored, or unfactored list of users and DSUs (see the following example).

Example: The following is a list of destinations (qualified by RGN.REN.DGN.DEN):
A.K.DA.U1, A.K.DA.U2, A.K.DB.U3, A.K.DB.U4,
A.L.DC.U5, A.L.DC.U6, A.L.DD.U7, A.L.DD.U8,

B.M.DE.U9, B.M.DE.U10, B.M.DF.U11, B.M.DF.U12,
B.N.DG.U13, B.N.DG.U14, B.N.DH.U15, and B.N.DH.U16.

The list may appear factored in *destination_operands* as follows:

- Fully factored:

```
A(K(DA(U1
       U2)
    DB(U3
       U4))
  L(DC(U5
       U6)
    DD(U7
       U8)))
B(M(DE(U9
       U10)
    DF(U11
       U12))
  N(DG(U13
       U14)
    DH(U15
       U16))))
```

- Partially factored:

```
(A(K(DA(U1)
     DA(U2)
     DB(U3
        U4))
  L(DC(U5
       U6))
  L(DD(U7
       U8)))
B(M(DE(U9
       U10)
    DF(U11
       U12))
  N(DG(U13))
  N(DG(U14))
  N(DH(U15
       U16))))
```

- Unfactored, equivalent to the initial list:

```
(A(K(DA(U1)))
 A(K(DA(U2)))
 A(K(DB(U3)))
 A(K(DB(U4)))
 A(L(DC(U5)))
 A(L(DC(U6)))
 A(L(DD(U7)))
 A(L(DD(U8)))
 B(M(DE(U9)))
 B(M(DE(U10)))
 B(M(DF(U11)))
 B(M(DF(U12)))
```

B(N(DG(U13)))
B(N(DG(U14)))
B(N(DH(U15)))
B(N(DH(U16))))

In the above lists, "(" represents *begin_dest_operands*, *begin_REN_list*, *begin_DGN_list*, or *begin_DEN_list*. ")" represents *end_DEN_list*, *end_DGN_list*, *end_REN_list*, or *end_dest_operands*. (Inner parentheses have precedence over outer parentheses.)

Begin_Dest_Operands

Description: The *beginning_of_the_destination_operands* marks the beginning of the *destination_list*.

Format: Constant byte string; value is X'C35201'

Dest_RGN_List

Description: The *destination_RGN_list* associates one destination RGN with at least one destination REN.

Dest_RGN

Description: The *destination_RGN* is the first part of a destination DSU name. This is typically, but not necessarily, the network ID.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_REN_List

Description: The *beginning_of_the_destination_REN_list* marks the beginning of a list of one or more *dest_REN*(s).

Format: Constant byte string; value is X'C35301'

Dest_REN_List

Description: The *destination_REN_list* associates one destination REN with at least one destination DGN.

Dest_REN

Description: The *destination_REN* is the second part of a destination DSU name. This is typically, but not necessarily, the LU name.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_DGN_List

Description: The *beginning_of_the_destination_DGN_list* marks the beginning of a list of one or more *dest_DGN*(s).

Format: Constant byte string; value is X'C35401'

Dest_DGN_List

Description: The *destination_DGN_list* associates one *dest_DGN* with at least one *dest_DEN*.

Dest_DGN

Description: The *destination_DGN* is the first part of the name of a destination user.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_DEN_List

Description: The *beginning_of_the_destination_DEN_list* marks the beginning of a list of one or more *dest_DEN*(s).

Format: Constant byte string; value is X'C35501'

Dest_DEN

Description: The *destination_DEN* is the second part of the name of a destination user.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

End_DEN_List

Description: The *end_destination_DEN_list* marks the end of the list begun by the corresponding *begin_DEN_list*.

End_DGN_List

Description: The *end_destination_DGN_list* marks the end of the list begun by the corresponding *begin_DGN_list*.

End_REN_List

Description: The *end_destination_REN_list* marks the end of the list begun by the corresponding *begin_REN_list*.

End_Dest_Operands

Description: The *end_destination_operands* marks the end of the *destination_list*.

Dist_Server_Operands

Description: The *distribution_server_operands* structure contains the *server_prefix* and the *server_object*.

Presence Rule: This information occurs only in Dist_MU *type* TRANSPORT when *server_object_ind* = X'0001'.

Server_Prefix

Description: The *server_prefix* contains information associated with the *server_object*.

Server_Obj_Byte_Count

Description:	The *server_object_byte_count* is the number of bytes of all the segments of the *server_object*.
Note:	In FS1, the byte count need not be accurate. In FS2, the originating DSU either supplies a correct byte count or omits the field completely.
Presence Rule:	Optional when the *server_object* is present; otherwise, precluded.
Format:	Unsigned binary integer (1-origin)
Range of Values:	Valid values range from 1 to 2^{64}-2.

Server

Description:	The *server* is the name of the transaction program to be used to store the *server_object* at the destination.
Presence Rule:	Required when the *server_object* is present.
Note:	In FS2, optional when the *server_object* is present; otherwise, precluded. If optional and absent, the general server TP name is the default.
Format:	Character string, except for first byte

CGCSGID	String Conventions
01130-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Server_Parms

Description:	The *server_parameters* structure contains parameters passed by SNA/DS to the destination server.
Note:	This structure is never sent, and is retired in FS2.
Format:	Undefined byte string

Server_Object

Description:	The *server_object* is identified by the origin agent and is fetched by the origin server during transmission of the Dist_MU *type* TRANSPORT. At each destination, the *server_object* is stored by the destination server and a notification of its receipt is delivered to the destination agent.
Length Restriction:	The maximum segment size for FS1 is 32511.
Format:	Undefined byte string

DS_Suffix (FS1)

Description: The *distribution_services_suffix* contains no information and marks the end of the message unit.

Dist_Report_Operands

Description: The *distribution_report_operands* structure contains all the report information describing the condition of a particular distribution.

Presence Rule: This information occurs only when Dist_MU is of type REPORT.

Report_Operands

Description: The *report_operands* structure contains all information pertaining to the originator of the distribution and the detector of an exception.

Report_Correlation

Description: The *report_correlation* contains information that uniquely identifies a distribution being reported on.

Reported-On_Origin_DGN

Description: The *reported-on_origin_DGN* is the first part of the name of the user that originated the distribution.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Reported-On_Origin_DEN

Description: The *reported-on_origin_DEN* is the second part of the name of the user that originated the distribution.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Reported-On_Seqno

Description: The *reported-on_origin_sequence_number* is the sequence number of the distribution being reported on.

Format: Character string; each character represents the EBCDIC representation of one digit of the sequence number.

Byte	Content
0-1	LT header
2-5	Sequence number Note: Values range from X'F0F0F0F1' to X'F9F9F9F9'.

Reported-On_DTM

Description: The *reported-on_date-time* is the date and time the distribution was originated.

Note: FS1 supports neither the GMT format nor the offset time format supported by FS2.

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X'07C5')
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)

Note: Refer to "Representing Date and Time" on page 15-38 for a complete discussion of the encoding and interpretation of date and time.

Reported-On_Agent_Correl

Description: The *reported-on_agent_correlation* is a string that was supplied by the origin agent at the origin DSU.

Format: Undefined byte string

Receiving_DSU

Description: The *receiving_DSU* is the name of the DSU to which a distribution was being sent.

Receiving_RGN

Description: The *receiving_RGN* is the first part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the network ID.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Receiving_REN

Description: The *receiving_REN* is the second part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the LU name.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Gen_SNADS_Report

Description: The *general_SNADS_report* contains the SNA/DS report applicable to each user specified in *specific_report* for which a *spec_SNADS_report* is not supplied.

Note: Older DSUs may generate both *gen_SNADS_report* and *gen_DIA_report* in a single MU. All DSUs are able to receive such MUs. However, DSUs may ignore *gen_DIA_report* if *gen_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.

Presence Rule: This information occurs when *gen_SNADS_type* = X'0001'.

Gen_SNADS_Type

Description: The *general_SNADS_type* indicates that a SNA/DS condition is being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' SNA/DS report Note: Any other value indicates that this is not a *gen_SNADS_report*.

Gen_SNADS_Content

Description: The *general_SNADS_contents* contains information describing the condition being reported on.

Gen_SNADS_Cond_Code

Description: The *general_SNADS_condition_code* is the particular condition being reported on.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2-3	X'0001' routing exception X'0002' unknown user name X'0003' hop count exhausted X'0004' format exception X'0005' function not supported X'0006' specific-server exception X'0007' unknown resource name (specific server) X'0008' invalid server parameters X'0009' unknown resource name (destination agent) X'000C' operator intervention (purging) X'000D' user names lost X'000E' resource not available X'000F' system exception X'0010' insufficient resource X'0011' storage-medium exception X'0012' REMU exception X'0013' server object size incompatible with capacity level Note: All other values are reserved.

Gen_DIA_Report

Description:	The *general_DIA_report* describes an application-layer condition. The *gen_DIA_report* applies to all users specified in *specific_report*. The interaction between *gen_DIA_report* and *spec_DIA_report* is defined by DIA.
Note:	Older DSUs may generate both *gen_SNADS_report* and *gen_DIA_report* in a single MU. All DSUs can receive such MUs. However, DSUs may ignore *gen_DIA_report* if *gen_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.
Presence Rule:	This information occurs when *gen_DIA_type* ≠ X'0001'.

Gen_DIA_Type

Description:	The *general_DIA_type* indicates the type of DIA condition being reported.
Format:	Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' indicates this is not a *gen_DIA_report* X'0200' DIA application exceptions X'FEFF' reserved for 5520 migration Note: All other values are reserved.

Gen_DIA_Contents

Description:	The *general_DIA_contents* structure contains a DIA-defined byte string.
Length Restriction:	Older DSUs may generate MUs with length of up to 517. All DSUs receive such MUs without generating an exception. However, DSUs may modify such MUs to force the length to be 69 or less. For *gen_DIA_type* of X'0200' (DIA application exceptions), the truncation algorithm is given in the *DIA Transaction Programmer's Guide*. The length is at least 7, since *gen_DIA_contents* contains at least a null LT (an LT of length 2).
Format:	Undefined byte string

Specific_Report

Description:	The *specific_report* contains the portion of the destination users that are being reported on. Any specific SNA/DS and/or DIA reports are also specified within this structure.

Begin_Report_DGN_List

Description:	The *beginning_of_report_DGN_list* marks the beginning of the *specific_report*.
Format:	Constant byte string; value is X'C35401'

Report_DGN_List

Description: The *report_DGN_list* associates one *reported-on_dest_DGN* with at least one *reported-on_dest_DEN*.

Reported-On_Dest_DGN

Description: The *reported-on_destination_DGN* is the first part of the name of one of the original destination users being reported on.

Note: In FS1, for a SNA/DS condition code of X'000D' (lost user names), user names will be null.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_Report_DEN_List

Description: The *beginning_of_report_DEN_list* marks the beginning of a list of one or more *reported-on_dest_DEN*s.

Format: Constant byte string; value is X'C35501'

Report_DEN_List

Description: The *report_DEN_list* associates one *reported-on_dest_DEN* with a specific SNA/DS and/or DIA report.

Reported-On_Dest_DEN

Description: The *reported-on_destination_DEN* is the second part of the name of one of the original destination users being reported on.

Note: In FS1, for a SNA/DS condition code of X'000D' (lost user names), user names will be null.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Spec_SNADS_Report

Description: The *specific_SNADS_report* is a report on one particular user. This report overrides the *gen_SNADS_report*, if one exists, for that particular user.

Note: Older DSUs may generate both *spec_SNADS_report* and *spec_DIA_report* in a single MU. All DSUs can receive such MUs. However, DSUs may ignore *spec_DIA_report* if *spec_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.

Presence Rule: This information occurs when *spec_SNADS_type* = X'0001'.

Spec_SNADS_Type

Description: The *specific_SNADS_type* indicates that a SNA/DS condition is being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' SNA/DS report Note: Any other value indicates that this is not a *spec_SNADS_report*.

Spec_SNADS_Cont

Description: The *specific_SNADS_contents* contains information describing a condition being reported on.

Spec_SNADS_CC

Description: The *specific_SNADS_condition_code* describes the particular condition being reported on.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2-3	X'0001' routing exception X'0002' unknown user name X'0003' hop count exhausted X'0004' format exception X'0005' function not supported X'0006' specific-server exception X'0007' unknown resource name (specific server) X'0008' invalid server parameters X'0009' unknown resource name (destination agent) X'000C' operator intervention (purging) X'000D' user names lost X'000E' resource not available X'000F' system exception X'0010' insufficient resource X'0011' storage-medium exception X'0012' REMU exception X'0013' server object size incompatible with capacity level Note: All other values are reserved.

Spec_DIA_Report

Description: The *specific_DIA_report* describes a DIA-specific report on one particular user.

Note: Older DSUs may generate both *spec_SNADS_report* and *spec_DIA_report* in a single MU. All DSUs can receive such MUs. However, DSUs may ignore *spec_DIA_report* if *spec_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.

Presence Rule: This information occurs when *spec_DIA_type* $\neq$ X'0001'.

Spec_DIA_Type

Description: The *specific_DIA_type* indicates the type of DIA condition being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' indicates this is not a *spec_DIA_report* X'0200' DIA application exceptions X'FEFF' reserved for 5520 migration Note: All other values are reserved.

Spec_DIA_Contents

Description: The *specific_DIA_contents* structure contains a DIA-defined byte string.

Length Restriction: Older DSUs may generate MUs with length of up to 517. All DSUs receive such MUs without generating an exception. However, DSUs may modify such MUs to force the length to be 69 or less. For *spec_DIA_type* of X'0200' (DIA application exceptions), the truncation algorithm is given in the *DIA Transaction Programmer's Guide*. The length is at least 7, since *spec_DIA_contents* contains at least a null LT (an LT of length 2).

Format: Undefined byte string

End_Report_DEN_List

Description: The *end_report_DEN_list* marks the end of the list begun by *begin_report_DEN_list*.

End_Report_DGN_List

Description: The *end_report_DGN_list* marks the end of the *specific_report*.

Sender_Exception_MU (Type FS1)

Description: The *sender_exception_MU* (type FS1) is sent from the sender to the receiver when the sender detects an exception while sending a Dist_MU.

Format: Byte string

Byte	Bit	Content
0-4		LLIDF header
5		Severity:
	0-1	11 catastrophic
		Class:
	2-7	000101 sender
6		Exception condition code:
		X'06' execution terminated
		X'0B' I/O error
		X'0F' length invalid
		X'18' content error
7		Exception object:
		X'01' IU prefix
		X'07' command
		X'0C' document unit
		X'13' IU suffix
		X'17' unknown subfield
		X'1A' distribution object prefix
		X'1B' distribution object data

Note: Other values and their corresponding meanings are represented under *receiver_exception_code.*

Receiver_Exception_MU (Type FS1)

Description: The *receiver_exception_MU* (type FS1) is sent from the receiver to the sender when the receiver detects an exception while receiving a Dist_MU.

Receiver_Exception_Command

Description: The *receiver_exception_command* contains all information used for identifying the exception that occurred.

Receiver_Exception_Correl

Description: The *receiver_exception_correlation* contains the *prefix* ID value from the rejected Dist_MU.

Format: Byte string

Byte	Content
0-4	LLIDF header
5	Correlation field: X'00' Note: All other values are reserved.
6	Command sequence number: X'01' Note: All other values are reserved.
7-22	Correlation MU ID; value from the *prefix* of the Dist_MU

Exception_And_Reply_Data

Description: The *exception_and_reply_data* contains information pertaining to the exception causing the rejection of the Dist_MU.

Receiver_Exception_Code

Description: The *receiver_exception_code* identifies the type of exception encountered and, conditionally, the portion of the Dist_MU containing the exception.

Format: Byte string

Byte	Bit	Content
0-4		LLIDF header
5		Severity:
	0-1	11 catastrophic Note: All other values for bits 0-1 are reserved.
		Class:
	2-7	000010 syntactic 000011 semantic 000100 process Note: All other values for bits 2-7 are reserved or defined elsewhere.
6		Exception condition code (indicates reason for exception): X'01' function not supported X'02' data not supported X'04' resource not available X'06' execution terminated X'07' data not found X'08' segmentation X'0A' sequence X'0B' I/O error X'0C' ID invalid X'0E' format invalid X'0F' length invalid X'10' indicator invalid X'11' range exceeded X'15' subfield length invalid X'16' subfield type invalid X'17' invalid parameters X'18' content error Note: All other values are reserved.
7		Exception object (indicates the syntactical entity in error): X'01' IU prefix X'02' IU identifier X'07' command X'08' command operand X'09' operand value X'0C' document unit

Byte	Bit	Content
		X'0D' document unit identifier X'0E' document profile X'0F' document profile parameter X'10' document content introducer X'11' document content control X'12' document content data X'13' IU suffix X'14' segment X'16' unsupported subfield X'17' unknown subfield X'1A' distribution object prefix X'1B' distribution object data Note: All other values are reserved.
8-254		Exception data contains the Dist_MU structures in error

Reply_Data

Description: The *reply_data* describes which DSU rejected the Dist_MU and why the Dist_MU was rejected.

SNADS_Report

Description: The *SNADS_report* contains information describing the particular SNA/DS exception that caused the Dist_MU to be rejected.

SNADS_Report_Type

Description: The *SNADS_report_type* indicates that a SNA/DS exception is being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' SNA/DS report Note: Any other value indicates that this is not a *SNADS_report.*

SNADS_Report_Cont

Description: The *SNADS_report_contents* structure contains information describing the type of SNA/DS condition in the Dist_MU.

SNADS_Report_CC

Description: The *SNADS_report_condition_code* describes the particular SNA/DS condition that caused the Dist_MU to be rejected.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2-3	X'0001' routing exception X'0002' unknown user name X'0003' hop count exhausted X'0004' format exception X'0005' function not supported X'0006' specific-server exception X'0007' unknown resource name (specific server) X'0008' invalid server parameters X'0009' unknown resource name (destination agent) X'000E' resource not available X'000F' system exception X'0010' insufficient resource X'0011' storage-medium exception X'0013' server object size incompatible with capacity level Note: All other values are reserved.

DIA_Report

Description: The *DIA_report* describes a DIA condition being reported.

Note: When generating a Dist_MU *type* REPORT with report information supplied by a REMU (type FS1), the reporting DSU may ignore *DIA_report*.

Presence Rule: This information occurs when *gen_DIA_type* ≠ X'0001'.

DIA_Report_Type

Description: The *DIA_report_type* indicates the type of DIA condition being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' indicates this is not a *DIA_report* X'0200' DIA application exceptions X'FEFF' reserved for 5520 migration Note: All other values are reserved.

DIA_Report_Cont

Description:	The *DIA_report_contents* structure contains a DIA-defined byte string.
Format:	Undefined byte string

Transaction Program and Server Names

Following is a list of all transaction program and server names defined for SNA/DS in the FM header 5 (Attach), in the Distribution MU, or used internally in the distribution service unit (DSU).

Code	Meaning
X'20F0F0F0'	DIA process destination transaction program name
X'20F0F0F1'	DIA server name
X'20F0F0F2'	DIASTATUS transaction program name
X'21F0F0F1'	DS_SEND transaction program name (FS1)
X'21F0F0F2'	DS_RECEIVE transaction program name (FS1)
X'21F0F0F3'	DS_ROUTER_DIRECTOR transaction program name
X'21F0F0F6'	SNA/DS general server name
X'30F0F0F2'	Object Distribution transaction program.
X'30F0F0F3'	Object Distribution server transaction program.

Code Points Used by SNA/DS FS1

The values of the ID component of the LLIDF structure as used for SNA/DS GDS variables are shown below:[1]

ID	Structure Name
C001*	In DIA, MU PREFIX; in SNA/DS, Prefix within DIST_MU or within REMU (type FS1)
C101*	in DIA, MU CMD NO REPLY ACKNOWLEDGE; in SNA/DS, Command within REMU (type FS1)
C105	Command, DIST_MU
C322*	in DIA, MU OPERAND IMM DATA EXCEPTION-CODE; in SNA/DS, Exception Code, within REMU (type FS1)
C328*	in DIA, MU OPERAND IMM DATA DATA CORRELATION; in SNA/DS, Correlation, within REMU (type FS1)
C32D*	in DIA, MU OPERAND IMM DATA USER-DATA; in SNA/DS, Agent Object within DIST_MU
C33D*	in DIA, MU OPERAND IMM DATA STATUS-INFORMATION; in SNA/DS, Distribution General Options, within DIST_MU
C340*	in DIA, MU OPERAND IMM DATA DISTRIBUTION-IDENTIFIER; in SNA/DS, Distribution Identifier, within DIST_MU
C343*	in DIA, MU OPERAND IMM DATA GENERAL-ROUTING-DATA; in SNA/DS, Report-To Options within DIST_MU
C345*	in DIA, MU OPERAND IMM DATA REPLY DATA; in SNA/DS, Reply Data, within REMU (type FS1)
C350	Beginning of Destination Operand Lists, of the Specific Report Lists, within DIST_MU
C351	End of Destination Operands Lists, of the Specific Report Lists, within DIST_MU
C352	Routing Group Name (RGN) of Destination Operands, within DIST_MU
C353	Routing Element Name (REN) of REN List, within DIST_MU
C354	Distribution Group Name (DGN) of DGN List, within DIST_MU
C355	Distribution Element Name (DEN) of DEN List, within DIST_MU
C356	Report Type, within DIST_MU
C357	Report Contents, within DIST_MU
C360	Report-To Address, within DIST_MU

[1] The asterisk following the ID indicates that that identifier is used by both DIA (Document Interchange Architecture) and SNA/DS.

C361 Receiving DSU, within DIST_MU or within REMU (type FS1)

C908 Server Object, within DIST_MU

C90A Server Prefix, within DIST_MU

CF01* in DIA, MU SUFFIX NORMAL-TERMINATION; in SNA/DS, Suffix within DIST_MU or within REMU (type FS1)

CF02* in DIA, MU SUFFIX ABNORMAL-TERMINATION; in SNA/DS, SEMU (type FS1)

Terminology Mappings

Figure 14-5 (Page 1 of 3). Terminology Mappings

FS2 TERMINOLOGY	Current FS1 TERMINOLOGY	Old FS1 TERMINOLOGY
Dist_Transport_MU	Dist_MU (type Transport)	Dist_IU (type Data)
Transport_Prefix	Prefix	Prefix
Hop_Count	Hop_Count	Dist_Dest_Hops
MU_ID	—	—
Transport_Command	Dist_Command	Dist_CMD
Dist_Flags	Dist_Flags (FS1)	Dist_Flags
Service_Parms	Service_Parms	DSL
Server_Obj_Byte_Count	Server_Obj_Byte_Count	Data_Size
Origin_Agent	Origin_Agent	Dest_TPN
Server	Server	Server_Name
Origin_DSU	—	—
Origin_RGN	Origin_RGN	Orig_RGN
Origin_REN	Origin_REN	Orig_REN
Origin_User	—	—
Origin_DGN	Origin_DGN	Orig_DGN
Origin_DEN	Origin_DEN	Orig_DEN
Seqno_DTM	Origin_Seqno, Origin_DTM	Orig_Seqno, Orig_DTM
Ext_Net	—	—
Agent_Correl	Agent_Correl	Orig_Correl
Report-To_DSU	—	—
Report-To_RGN	Report-To_RGN	Fdbk_RGN
Report-To_REN	Report-To_REN	Fdbk_REN
Report-To_User	—	—
Report-To_DGN	Report-To_DGN	Fdbk_DGN
Report-To_DEN	Report-To_DEN	Fdbk_DEN
Report_Service_Parms	Report_Service_Parms	Fdbk_DSL
Report-To_Agent	Report-To_Agent	Fdbk_TPN
Dest_Agent	—	—
Unrecognized_Reserve	—	—
Dest_List	Destination_Operands	Destination_Operands
Dest	—	—
Dest_DSU	—	—
Dest_RGN	Dest_RGN	Dest_RGN
Dest_REN	Dest_REN	Dest_REN
Dest_User	—	—
Dest_DGN	Dest_DGN	Dest_DGN
Dest_DEN	Dest_DEN	Dest_DEN
Agent_Object	Agent_Object	Dest_Appl_Parms
Server_Object	Server_Object	Distrib_Object_Data

Figure 14-5 (Page 2 of 3). Terminology Mappings

FS2 TERMINOLOGY	Current FS1 TERMINOLOGY	Old FS1 TERMINOLOGY
Ext_Net_Correl	—	—
Ext_Net_Object	—	—
DS_Suffix	DS_Suffix	Suffix
Dist_Report_MU	Dist_MU (type Report)	Dist_IU (type Status)
Report_Prefix	—	—
Report_Command	—	—
Reporting_DSU	—	—
Reporting_RGN	—	—
Reporting_REN	—	.—
Report_DTM	—	—
Report-To_DSU_User	—	—
Report_Information	—	—
Reported-On_Origin_DSU	—	—
Reported-On_Origin_RGN	—	—
Reported-On_Origin_REN	—	—
Reported-On_Origin_User	—	—
Reported-On_Origin_DGN	Reported-On_Origin_DGN	Orig_DGN
Reported-On_Origin_DEN	Reported-On_Origin_DEN	Orig_DEN
Reported-On_Seqno_DTM	Reported-On_Seqno, Reported-On_DTM	Orig_Seqno, Orig_DTM
Reported-On_Ext_Net	—	—
Reported-On_Ext_Net_Correl	—	—
Reported-On_Agent_Correl	Reported-On_Agent_Correl	Orig_Correl
Reported-On_Dest_Agent	—	—
Reported-On_Hop_Count	—	—
SNA_Condition_Report	—	—
SNA_Report_Code	—	—
Structure_Report	—	—
Structure_State	—	—
Structure_Contents	—	—
Parent_Spec	—	—
Parent_ID_Or_T	—	—
Parent_Class	—	—
Parent_Position	—	—
Parent_Instance	—	—
Structure_Spec	—	—
Structure_ID_Or_T	—	—
Structure_Class	—	—
Structure_Position	—	—
Structure_Instance	—	—
Structure_Segment_Num	—	—
Structure_Byte_Offset	—	—

Figure 14-5 (Page 3 of 3). Terminology Mappings

FS2 TERMINOLOGY	Current FS1 TERMINOLOGY	Old FS1 TERMINOLOGY
Sibling_List	—	—
Reported-On_Dest_List	Specific_Report	Specific_Status
Reported-On_Dest_Pfx	—	—
Reported-On_Dest	—	—
Reported-On_Dest_DSU	—	—
Reported-On_Dest_RGN	—	—
Reported-On_Dest_REN	—	—
Reported-On_Dest_User	—	—
Reported-On_Dest_DGN	Reported-On_Dest_DGN	Stat_DGN
Reported-On_Dest_DEN	Reported-On_Dest_DEN	Stat_DEN
Reported-On_Dest_Sfx	—	—
Supplemental_Report	—	—
Dist_Continuation_MU	—	—
Continuation_Prefix	—	—
Restarting_Byte_Position	—	—
Sender_Exception_MU	Sender_Exception_MU	Suffix (type 2)
Receiver_Exception_MU	Receiver_Exception_MU	Ack_IU
Receiver_Exception_Command	Receiver_Exception_Command	Ack_Cmd
Sender_Retry_Action	—	—
Receiving_DSU	Receiving_DSU	Rcv_DSUN
Receiving_RGN	Receiving_RGN	Rcv_DSUN_RGN
Receiving_REN	Receiving_REN	Rcv_DSUN_REN
Completion_Query_MU	—	—
Completion_Report_MU	—	—
Indicator_Flags	—	—
Last_Structure_Received	—	—
Last_Byte_Received	—	—
Purge_Report_MU	—	—
Reset_Request_MU	—	—
Reset_DTM	—	—
Reset_Accepted_MU	—	—

Chapter 15. SNA/DS FS2 Encodings

Introduction

This chapter contains the format descriptions of the FS2 message units. The format descriptions are comprised of two parts: *header description tables* and *structure descriptions*. A header description table contains the header information for each structure associated with a particular message unit. A structure description contains a prose description of the structure, bit-level representations, and any presence rules or length restrictions associated with a particular structure.

The definition of SNA/Distribution Services (SNA/DS) requires a byte-accurate description of the formats that must be understood by all DSUs. The SNA/DS formats are described in terms of encoded fields referred to as "structures" and the hierarchical relationship between these structures. In this chapter, the header description tables show each structure and its header.

Refer to Appendix C, "Common Structures" on page C-1 for a complete definition and classification of the encoding structures used in the following tables.

Header Description Tables for FS2 Message Units

DISTRIBUTION TRANSPORT MESSAGE UNIT (DTMU)

Figure 15-1 (Page 1 of 2). Distribution Transport Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Transport_MU	15-8	Del-ID	pfx	≥53*	1	Y	Y	≥4	—
Transport_Prefix	15-8	ID/pfx	1570	8-18	1	N	Y	1-3	—
Hop_Count	15-8	T	01	4	1	—	—	—	—
MU_ID	15-8	T	03	6	0-1*	—	—	—	—
MU_Instance_Number	15-8	T	06	4	0-1*	—	—	—	—
Transport_Command	15-9	ID/seg	1571	29-4096*	1	Y	Y	≥3	—
Dist_Flags	15-9	T	01	5	0-1	—	—	—	—
Service_Parms	15-10	T	02	5-32	0-1	—	—	—	—
Server_Obj_Byte_Count	15-13	T	03	10	0-1*	—	—	—	—
Origin_Agent	15-13	T	04	3-10	1	—	—	—	—
Server	15-14	T	05	3-10	0-1*	—	—	—	—
Origin_DSU	15-14	T	06	8-22	1	N	Y	2	—
Origin_RGN	15-14	T	01	3-10	1	—	—	—	—
Origin_REN	15-14	T	02	3-10	1	—	—	—	—
Origin_User	15-14	T	07	8-22	0-1	N	Y	2	—
Origin_DGN	15-15	T	01	3-10	1	—	—	—	—
Origin_DEN	15-15	T	02	3-10	1	—	—	—	—
Seqno_DTM	15-15	T	08	14-17*	1	—	—	—	—
Supplemental_Dist_Info1	15-16	T	09	3-10	0-1	—	—	—	—
Agent_Correl	15-16	T	0A	3-130	0-1	—	—	—	—
Report-To_DSU	15-17	T	0B	8-22	0-1	N	Y	2	—
Report-To_RGN	15-17	T	01	3-10	1	—	—	—	—
Report-To_REN	15-17	T	02	3-10	1	—	—	—	—
Report-To_User	15-17	T	0C	8-22	0-1	N	Y	2	—
Report-To_DGN	15-18	T	01	3-10	1	—	—	—	—
Report-To_DEN	15-18	T	02	3-10	1	—	—	—	—
Report_Service_Parms	15-19	T	0D	5-32	0-1	—	—	—	—
Report-To_Agent	15-21	T	0E	3-10	0-1	—	—	—	—
Dest_Agent	15-22	T	0F	3-10	0-1	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-3728	—	—	—	—	—
Dest_List	15-22	ID/seg	1572	12-11268	1	N	Y	1	—
Dest	15-22	Imp-T	idc	8-5654	≥1	N	Y	1-2	—
Dest_DSU	15-22	T/idc	01	8-22	1	N	Y	2	—
Dest_RGN	15-23	T	01	3-10	1	—	—	—	—

Figure 15-1 (Page 2 of 2). Distribution Transport Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dest_REN	15-23	T	02	3-10	1	—	—	—	—
Dest_User	15-23	T	02	8-22	≥0	N	Y	2	—
Dest_DGN	15-23	T	01	3-10	1	—	—	—	—
Dest_DEN	15-24	T	02	3-10	1	—	—	—	—
Agent_Object	15-24	ID/seg	1573	5-32767	0-1	—	—	—	—
Server_Object	15-24	ID/seg	1574	≥5	0-1	—	—	—	—
Supplemental_Dist_Info2	15-24	ID/seg	1580	5-32767	0-1	—	—	—	—
Unrecognized_Reserve	15-37	ID/seg	—	4-32767	—	—	—	—	—
DS_Suffix	15-24	ID/sfx	157F	4	1	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-8 for presence rules and length restrictions.

DISTRIBUTION REPORT MESSAGE UNIT (DRMU)

Figure 15-2 (Page 1 of 2). Distribution Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Report_MU	15-25	Del-ID	pfx	≥77*	1	Y	Y	≥6	—
Report_Prefix	15-25	ID/pfx	157C	8-18	1	N	Y	1-3	—
Hop_Count	15-8	T	01	4	1	—	—	—	—
MU_ID	15-8	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-8	T	06	4	0-1*	—	—	—	—
Report_Command	15-25	ID/seg	1575	25-4096*	1	Y	Y	≥3	—
Service_Parms	15-10	T	02	5-32	0-1	—	—	—	—
Report-To_Agent	15-21	T	04	3-10	1	—	—	—	—
Reporting_DSU	15-25	T	06	8-22	1	N	Y	2	—
Reporting_RGN	15-25	T	01	3-10	1	—	—	—	—
Reporting_REN	15-25	T	02	3-10	1	—	—	—	—
Report_DTM	15-26	T	09	10-13*	1	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-4015	—	—	—	—	—
Report-To_DSU_User	15-27	ID	1583	12-48	1	N	Y	1-2	—
Report-To_DSU	15-17	T	01	8-22	1	N	Y	2	—
Report-To_RGN	15-17	T	01	3-10	1	—	—	—	—
Report-To_REN	15-17	T	02	3-10	1	—	—	—	—
Report-To_User	15-17	T	02	8-22	0-1	N	Y	2	—
Report-To_DGN	15-18	T	01	3-10	1	—	—	—	—
Report-To_DEN	15-18	T	02	3-10	1	—	—	—	—
Report_Information	15-27	ID/seg	1576	18-4096	1	Y	Y	≥1	—
Reported-On_Origin_DSU	15-27	T	06	8-22	0-1*	N	Y	2	—
Reported-On_Origin_RGN	15-27	T	01	3-10	1	—	—	—	—
Reported-On_Origin_REN	15-27	T	02	3-10	1	—	—	—	—
Reported-On_Origin_User	15-28	T	07	8-22	0-1*	N	Y	2	—
Reported-On_Origin_DGN	15-28	T	01	3-10	1	—	—	—	—
Reported-On_Origin_DEN	15-28	T	02	3-10	1	—	—	—	—
Reported-On_Seqno_DTM	15-29	T	08	14-17	1	—	—	—	—
Reported-On_Supp_Dist_Info1	15-30	T	09	3- 10	0-1	—	—	—	—
Reported-On_Agent_Correl	15-31	T	0A	3-130	0-1	—	—	—	—
Reported-On_Origin_Agent	15-31	T	0B	3-10	0-1*	—	—	—	—
Reported-On_Dest_Agent	15-31	T	0C	3-10	0-1*	—	—	—	—
Receiving_DSU	15-33	T	10	8-22	0-1	N	Y	2	—
Receiving_RGN	15-33	T	01	3-10	1	—	—	—	—
Receiving_REN	15-33	T	02	3-10	1	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-3849	—	—	—	—	—
SNA_Condition_Report	C-9	ID/seg	1532	10-32767	1	Y	Y	≥1	C-7
Reported-On_Supp_Dist_Info2	15-31	ID/seg	1582	5-32767	0-1*	—	—	—	—
Unrecognized_Reserve	15-37	ID/seg	—	4-32767	—	—	—	—	—

Figure 15-2 (Page 2 of 2). Distribution Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
DS_Suffix	15-24	ID/sfx	157F	4	1	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-8 for presence rules and length restrictions.

DISTRIBUTION CONTINUATION MESSAGE UNIT (DCMU)

Figure 15-3. Distribution Continuation Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Continuation_MU	15-32	Del-ID	pfx	≥18	1	Y	Y	≥2	—
Continuation_Prefix	15-32	ID/pfx	157B	14-24	1	N	Y	2-3	—
MU_ID	15-8	T	03	6	1	—	—	—	—
MU_Instance_Number	15-8	T	06	4	1	—	—	—	—
Restarting_Byte_Position	15-32	T	02	10	0-1	—	—	—	—
Agent_Object	15-24	ID/seg	1573	5-32767	0-1	—	—	—	—
Server_Object	15-24	ID/seg	1574	≥5	0-1	—	—	—	—
Supplemental_Dist_Info2	15-24	ID/seg	1580	5-32767	0-1	—	—	—	—
Unrecognized_Reserve	15-37	ID/seg	—	4-32767	—	—	—	—	—
DS_Suffix	15-24	ID/sfx	157F	4	1	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-8 for presence rules.

SENDER EXCEPTION MESSAGE UNIT (SEMU)

Figure 15-4. Sender Exception Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Sender_Exception_MU	15-32	ID	1578	10-256	1	Y	Y	≥1	—
SNA_Report_Code	C-9	T	7D	6	1	—	—	—	—
MU_ID	15-8	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-8	T	06	4	0-1*	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-236	—	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-8 for presence rules.

RECEIVER EXCEPTION MESSAGE UNIT (REMU)

Figure 15-5. Receiver Exception Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Receiver_Exception_MU	15-32	Del-ID	pfx	≥25	1	Y	Y	≥2	—
Receiver_Exception_Command	15-32	ID/pfx	1577	15-512	1	Y	Y	≥2	—
Sender_Retry_Action	15-33	T	01	3	1	—	—	—	—
MU_ID	15-8	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-8	T	06	4	0-1*	—	—	—	—
Receiving_DSU	15-33	T	16	8-22	1	N	Y	2	—
Receiving_RGN	15-33	T	01	3-10	1	—	—	—	—
Receiving_REN	15-33	T	02	3-10	1	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-473	—	—	—	—	—
Unrecognized_Reserve	15-37	ID	—	≥4	—	—	—	—	—
SNA_Condition_Report	C-9	ID/sfx	1532	10-1024	1	Y	Y	≥1	C-7

Note: * Refer to FS2 Structure Descriptions starting on page 15-8 for presence rules.

COMPLETION QUERY MESSAGE UNIT (CQMU)

Figure 15-6. Completion Query Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Completion_Query_MU	15-34	ID	1579	14-256	1	Y	Y	≥2	—
MU_ID	15-8	T	03	6	1	—	—	—	—
MU_Instance_Number	15-8	T	06	4	1	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-242	—	—	—	—	—

COMPLETION REPORT MESSAGE UNIT (CRMU)

Figure 15-7. Completion Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Completion_Report_MU	15-34	ID	157A	7-256	1	Y	Y	≥1	—
Indicator_Flags	15-34	T	01	3	1	—	—	—	—
MU_ID	15-8	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-8	T	06	4	0-1*	—	—	—	—
Last_Structure_Received	15-34	T	04	4	0-1*	—	—	—	—
Last_Byte_Received	15-35	T	05	10	0-1*	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-225	—	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-8 for presence rules.

PURGE REPORT MESSAGE UNIT (PRMU)

Figure 15-8. Purge Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Purge_Report_MU	15-35	ID	157E	10-256	1	Y	Y	≥1	—
MU_ID	15-8	T	03	6	1	—	—	—	—
Unrecognized_Reserve	15-37	T	—	2-246	—	—	—	—	—

RESET REQUEST MESSAGE UNIT (RRMU)

Figure 15-9. Reset Request Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Reset_Request_MU	15-35	ID	1585	21-4096	1	—	Y	≥2	—
MU_ID	15-8	T	03	6	1	—	—	—	—
Reset_DTM	15-35	T	09	11-13	1	—	—	—	—

RESET ACCEPTED MESSAGE UNIT (RAMU)

Figure 15-10. Reset Accepted Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Reset_Accepted_MU	15-36	ID	1586	21-4096	1	—	Y	≥2	—
MU_ID	15-8	T	03	6	1	—	—	—	—
Reset_DTM	15-35	T	09	11-13	1	—	—	—	—

FS2 Structure Descriptions

Dist_Transport_MU

Description:	The *distribution_transport_message_unit* transports agent and/or server objects for distribution to one or more users or application programs.
Length Restriction:	The minimum length of a *dist_transport_MU* originated by an FS2 DSU is 54 bytes. This is due to the length restriction on the *Seqno_DTM*.

Transport_Prefix

Description:	The *transport_prefix* identifies the beginning of the *dist_transport_MU*. This structure carries information that changes from DSU to DSU.

Hop_Count

Description:	The *hop_count* is the remaining number of hops that may be traversed by a SNA/DS distribution on its way toward its destination DSUs. The *hop_count* is set by the origin DSU in the DTMUs and by the reporting DSUs for the DRMUs. The *hop_count* is decremented by 1 in every DSU through which the distribution passes. If the *hop_count* reaches 0 at an intermediate DSU, exception processing is invoked.
Format:	Signed binary integer
Range of Values:	Valid values range from 1 to 2^{15}-1.

MU_ID

Description:	The *message_unit_identifier* is a number that uniquely identifies a distribution MU throughout its existence. An MU exists for only one hop, from one DSU to the adjacent DSU. In REMUs and SEMUs, the *MU_ID* refers to a distribution MU. An *MU_ID* is unique only for a particular *LU name*, *mode name* combination.
Presence Rule:	If the *MU_ID* is absent, exception reporting may not be requested.
Format:	Signed binary integer
Range of Values:	Valid values range from 1 to 2^{31}-1.

MU_Instance_Number

Description:	The *message_unit_instance_number* identifies the instance of a particular distribution message unit and its corresponding *MU_ID*.
Presence Rule:	Precluded if an *MU_ID* is not present; otherwise, required.
Format:	Signed binary integer
Range of Values:	Valid values range from 1 to 2^{15}-1.

Transport_Command

Description:	The *transport_command* contains the control information used by the distribution service to transport the distribution.
Length Restriction:	The minimum length of a *transport_command* originated by an FS2 DSU is 30 bytes. This is due to the length restriction on the *seqno_DTM*.

Dist_Flags

Description:	The *distribution_flags* indicate services requested by the origin agent.
Note:	If exception reporting is requested, the *MU_ID* is always present.
Format:	Bit string

Byte	Bit	Content
0-1		LT header
2		Flags (bits 0-7) that must be understood and honored by all DSUs
	0	Exception report flag indicating whether an exception report is to be sent if the distribution is aborted: 0 no exception report to be sent (default) 1 exception report to be sent
	1 -7	Reserved
3		Flags (bits 0-7) that must be understood and honored by destination DSUs, but that can be ignored by intermediate DSUs
	0 -7	Reserved
4		Flags (bits 0-7) that are ignored by DSUs if not understood
	0- 7	Reserved

Service_Parms

Description: The *service_parameters* structure describes the types and levels of service requested for the distribution. The parameters in this structure are provided by the origin agent. The *service_parameters* used in the DTMU and the DRMU are similar; the differences in such usage and the default values used for absent *service_parameter* (SP) triplets are discussed under the individual triplets below. The default values specified below are assumed for absent *service_parameter* (SP) triplets. The specification for deriving the *service_parameters* for the DRMU is found in the description of *report_service_parameters* on page 15-19.

Format: Special format consisting of ordered, optional, SP triplets of the following general structure:

Byte	Bit	Content
0		Parameter type: All parameter type byte values less than X'80' are defined by or reserved for SNA/DS. All parameter type byte values greater than X'80' may be customer defined.
1		Comparison operator:
	0-3	1100 REQUIRE_LEVEL_GE 1110 REQUIRE_SUPPORT_FOR Note: All other values for bits 0-3 are reserved.
	4-7	Reserved
2		Value: The meaning of this byte depends on the parameter type.

Byte	Content
0-1	LT header
2-31	Up to 10 different *service_parameter* (SP) triplets may be carried in one distribution. Each triplet, when present, appears in ascending sequence of parameter type. The capacity triplet is not used in the DRMU. All FS2 service parameters are optional in both the DTMU and the DRMU. The architecturally defined service parameters are given below:

Priority SP Triplet

Byte	Content	
0	X'01'	
1	X'C0' REQUIRE_LEVEL_GE	
2	X'F0' FAST	(default)
	X'D0' CONTROL	
	X'80' DATA_16	(can be treated as DATAHI)
	X'78' DATA_15	(can be treated as DATAHI)
	X'70' DATA_14	(can be treated as DATAHI)
	X'68' DATA_13	(can be treated as DATAHI)
	X'60' DATA_12	(DATAHI)
	X'58' DATA_11	(can be treated as DATAHI)
	X'50' DATA_10	(can be treated as DATAHI)
	X'48' DATA_9	(can be treated as DATAHI)
	X'40' DATA_8	(can be treated as DATALO)
	X'38' DATA_7	(can be treated as DATALO)
	X'30' DATA_6	(can be treated as DATALO)
	X'28' DATA_5	(can be treated as DATALO)
	X'20' DATA_4	(DATALO)
	X'18' DATA_3	(can be treated as DATALO)
	X'10' DATA_2	(can be treated as DATALO)
	X'08' DATA_1	(can be treated as DATALO)
	Note: All other values are reserved.	

Notes:

1. The Priority SP value X'D0' (CONTROL) occurs in a DRMU only.
2. The Priority SP range for DATALO is X'01' to X'40'. The Priority SP range for DATAHI is X'41' to X'80'.
3. Implementations may accept other priority levels as long as they can route the distribution responsibly.

Protection SP Triplet

Byte	Content	
0	X'02'	
1	X'C0' REQUIRE_LEVEL_GE	
2	X'10' LEVEL1	(default when Priority SP is GE X'E0'): safe store may be performed.
	X'30' LEVEL2	(default when Priority SP is LT X'E0'): safe store must be performed.
	Note: All other values are reserved.	

Capacity SP Triplet

Byte	Content
0	X'03'
1	X'C0' REQUIRE_LEVEL_GE
2	Capacity value is the exponent of the power of 2 that represents the value of the required capacity for the *server_object* in the DTMU: X'00' ZERO (default when Priority SP is GE X'E0') used if there is no *server_object* in *dist_transport_MU*. X'14' 1MB X'16' 4MB X'18' 16MB (default when Priority SP is LT X'E0') Note: All other values are reserved.

Notes:

1. The Capacity SP triplet occurs only in a DTMU.
2. Receiving FS2 DSUs are always able to receive a capacity level of INDEFINITE (designated by X'E0FF' in bytes 1-2). Originating FS2 DSUs never generate the capacity level of INDEFINITE. The level replacing INDEFINITE is 16MB (X'C018').
3. The capacity requirement is for the *server_object*, and does not include the capacity needed to store and handle the other structures of the DTMU.
4. Implementations may accept other capacity levels as long as they can route the distribution responsibly.

Security SP Triplet

Byte	Content
0	X'04'
1	X'C0' REQUIRE_LEVEL_GE
2	X'01' LEVEL1 (default): security is not required. X'20' LEVEL2: security is required. Note: All other values are reserved.

Acceptable Delay SP Triplet

Byte	Content
0	X'05'
1	X'A0' REQUIRE_LEVEL_LE
2	X'FF' INDEFINITE Note: All other values are reserved.

Server_Obj_Byte_Count

Description:	The *server_object_byte_count* is the number of bytes of all the segments of the *server_object*. An FS2-capable DSU originating a distribution either supplies a correct byte count, or omits the field completely.
Presence Rule:	Optional when the *server_object* is present; otherwise, precluded.
Format:	Unsigned binary integer
Range of Values:	Valid values range from 1 to 2^{64}-2.

Origin_Agent

Description:	The *origin_agent* is the transaction program at the DSU at which the distribution originated.
Format:	Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Server

Description: The *server* is the name to be used to store the *server_object* at the destination.

Presence Rule: Optional when the *server_object* is present; otherwise, precluded. If optional and absent, the general server TP name is the default.

Format: Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Origin_DSU

Description: The *origin_DSU* is the name of the DSU at which the distribution originated.

Origin_RGN

Description: The *origin_RGN* is the first part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Origin_REN

Description: The *origin_REN* is the second part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Origin_User

Description: The *origin_user* is the user name of the originator of the distribution.

Origin_DGN

Description: The *origin_DGN* is the first part of the user name of the distribution originator.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Origin_DEN

Description: The *origin_DEN* is the second part of the user name of the distribution originator.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Seqno_DTM

Description: The *sequence_number/date-time*, in combination with the *origin_agent*, *origin_user*, and *origin_DSU*, uniquely identifies the distribution. The sequence number is the number assigned to the distribution by the origin agent. The date of the distribution is assigned by the origin agent; the time of the distribution is assigned by the origin DSU. The offset from local time to GMT is included.

Length Restriction: Originating FS2 DSUs never generate a local-only time. The minimum length for *seqno_DTM* is therefore 15 bytes (including its header).

Format: Byte string

Range of Values: Valid values for the sequence number portion of the *seqno_DTM* range from 1 to $2^{31}-1$. FS2 tolerates sequence numbers with value 0. However, sequence numbers with value 0 are never originated from within an FS2 network.

Byte	Content
0-1	LT header
	SEQNO
2-5	Signed binary integer ranging from 1 to $2^{31}-1$
	DATE
6-7	Year, in binary (e.g., 1989 is encoded as X'07C5')
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 13 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
14	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X'E9' (z) TIME is GMT and no offset required. X'4E' (+) TIME is local, OFFSET VALUE is required, and encoder's local time is ahead of GMT. X'60' (-) TIME is local, OFFSET VALUE is required, and encoder's local time trails GMT. Note: All other values are reserved.
	OFFSET VALUE
15	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 13 are valid)
16	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 59 are valid)

Note: Refer to "Representing Date and Time" on page 15-38 for a complete discussion of the encoding and interpretation of date and time.

Supplemental_Dist_Info1

Description:	The *supplemental_dist_info1* structure is reserved for future use.
Format:	Undefined byte string

Agent_Correl

Description:	The *agent_correlation* is a string supplied by the origin agent. SNA/DS is not aware of its contents.
Format:	Undefined byte string

Report-To_DSU

Description: The *report-to_DSU* is the name of the DSU to which distribution reports are to be sent. If both *report-to_DSU* and *report-to_user* are absent in the DTMU, the values generated in the DRMU for these structures default to the origin. If only *report-to_DSU* is present in the DTMU, then any report is sent to that DSU. If only *report-to_user* is present in the DTMU, then the reporting DSU will refer to its directory to determine *report-to_DSU*.

Report-To_RGN

Description: The *report-to_RGN* is the first part of the DSU name to which distribution reports are to be sent. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Report-To_REN

Description: The *report-to_REN* is the second part of the DSU name to which distribution reports are to be sent. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. If a product chooses to implement DGN = REN, the enhanced character string (ECS) subset is implemented in a particular network, and a DGN exists that contains an ECS character that is not an element of CGCSGID 01134-0500, then ECS characters may occur in this structure.

Report-To_User

Description: The *report-to_user* is the name of the user to which distribution reports are to be sent. If both *report-to_user* and *report-to_DSU* are absent in the DTMU, the values generated in the DRMU for these structures default to the origin. If only *report-to_user* is present in the DTMU, the reporting DSU refers to its directory to determine *report-to_DSU*.

Report-To_DGN

Description: The *report-to_DGN* is the first part of the user name to which distribution reports are to be sent.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_DEN

Description: The *report-to_DEN* is the second part of the user name to which distribution reports are to be sent.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report_Service_Parms

Description: The *report_service_parameters* structure describes the service requested for the distribution report by the origin agent when the agent wants to override the *service_parameters* that would be routinely generated by the reporting DSU for the report MU. If *report_service_parameters* are specified, they are used as the *service_parameters* in any DRMUs that are generated as part of the distribution. If the origin agent does not specify one or more of the *report_service_parameters*, a DSU that generates a report derives appropriate *service_parameters* for the DRMU from the *service_parameters* in the DTMU.

The comparison operators and values derived for the protection, security, and acceptable delay parameters are the same as those specified (explicitly or implicitly) in the DTMU.

For the priority service parameter, the value derived is either FAST or CONTROL. FAST is used if the DTMU specified FAST priority; CONTROL is used if the DTMU specified a DATA_N priority. CONTROL priority is used only in DRMUs; it may not be specified for the priority service parameter in a DTMU. If the origin agent explicitly specifies a value for the priority report service parameter, the value may be FAST, CONTROL, or DATA_N. The comparison operator for the priority service parameter is always REQUIRE_LEVEL_GE.

Format: Special format consisting of ordered, optional *report_service_parameter* triplets of the same general structure as for *service_parameters*. See *service_parameters* on page 15-10.

Byte	Content
0-1	LT header
2-31	Up to 10 different *report_service_parameter* (RSP) triplets may be carried in one distribution. Each triplet, when present, appears in ascending sequence of parameter type. The capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified. All service parameters are optional in both the DTMU and the DRMU.

Priority RSP Triplet

Byte	Content
0	X'01'
1	X'C0' REQUIRE_LEVEL_GE
2	X'F0' FAST X'D0' CONTROL X'80' DATA_16 (can be treated as DATAHI) X'78' DATA_15 (can be treated as DATAHI) X'70' DATA_14 (can be treated as DATAHI) X'68' DATA_13 (can be treated as DATAHI) X'60' DATA_12 (DATAHI) X'58' DATA_11 (can be treated as DATAHI) X'50' DATA_10 (can be treated as DATAHI) X'48' DATA_9 (can be treated as DATAHI) X'40' DATA_8 (can be treated as DATALO) X'38' DATA_7 (can be treated as DATALO) X'30' DATA_6 (can be treated as DATALO) X'28' DATA_5 (can be treated as DATALO) X'20' DATA_4 (DATALO) X'18' DATA_3 (can be treated as DATALO) X'10' DATA_2 (can be treated as DATALO) X'08' DATA_1 (can be treated as DATALO) Note: All other values are reserved.

Protection RSP Triplet

Byte	Content
0	X'02'
1	X'C0' REQUIRE_LEVEL_GE
2	X'10' LEVEL1: safe store may be performed. X'30' LEVEL2: safe store must be performed. Note: All other values are reserved.

Security RSP Triplet

Byte	Content
0	X'04'
1	X'C0' REQUIRE_LEVEL_GE
2	X'01' LEVEL1: security is not required. X'20' LEVEL2: security is required. Note: All other values are reserved.

Acceptable Delay RSP Triplet

Byte	Content
0	X'05'
1	X'A0' REQUIRE_LEVEL_LE
2	X'FF' INDEFINITE Note: All other values are reserved.

Report-To_Agent

Description:	The *report-to_agent* is the name of the application transaction program to be started after the report is queued for delivery. If *report-to_agent* is absent in the DTMU, the value specified in the DTMU for *origin_agent* is used in the DRMU for *report-to_agent*.
Format:	Character string, except for first byte.

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Dest_Agent

Description: The *destination_agent* is the transaction program at the destination DSU to which the distribution is to be delivered. If *dest_agent* is absent in the DTMU, the value specified for *origin_agent* is assumed to be the *dest_agent.*

Format: Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Dest_List

Description: The *destination_list* is the list of destinations for the distribution, which can contain up to 256 destinations. Each destination is a *dest_DSU* with or without a *dest_user*, expressed as (*dest_DSU* (,*dest_user*)). For single-destination distributions and distribution reports, the *dest_list* contains only one destination.

Either a flat destination list, of the form

(*dest_DSU* (*dest_user*)), ..., (*dest_DSU* (*dest_user*)), ...

or a factored destination list, of the form

(*dest_DSU* (*dest_user, dest_user,* ...)), (*dest_DSU* (*dest_user,* ...))

may be present. For example, a flat destination list might contain

(DSU_A USER_1), (DSU_A USER_2), (DSU_A), (DSU_B USER_3), (DSU_B USER_4)

whereas a factored destination list would contain

(DSU_A (USER_1, USER_2)), (DSU_A), (DSU_B (USER_3, USER_4)).

Dest

Description: The *destination* associates *dest_users* with a *dest_DSU.* For flat destination lists, there are zero or one user names per *dest.* For factored destination lists, there can be multiple user names per *dest.*

Dest_DSU

Description: The *destination_DSU* is the name of one of the DSUs to which the distribution is to be sent.

Dest_RGN

Description: The *destination_RGN* is the first part of a *dest_DSU* name. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Dest_REN

Description: The *destination_REN* is the second part of a *dest_DSU* name. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. If a product chooses to implement DGN = REN, the enhanced character string (ECS) subset is implemented in a particular network, and a DGN exists that contains an ECS character that is not an element of CGCSGID 01134-0500, then ECS characters may occur in this structure.

Dest_User

Description: The *destination_user* is the name of one of the users to which the distribution is to be sent.

Dest_DGN

Description: The *destination_DGN* is the first part of the name of a *dest_user*.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Dest_DEN

Description: The *destination_DEN* is the second part of the name of a *dest_user*.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Agent_Object

Description: The *agent_object* is directly supplied by the origin agent. It is never parsed by the distribution service and is directly delivered, unchanged, to the agent at each destination.

Format: Undefined byte string

Server_Object

Description: The *server_object* is identified by the origin agent and is fetched by the origin server when sending the *dist_transport_MU*. At each destination, the *server_object* is stored by the destination server and a notification of its receipt is delivered to the destination agent.

Format: Undefined byte string

Supplemental_Dist_Info2

Description: The *supplemental_dist_info2* structure is reserved for future use.

Format: Undefined byte string

DS_Suffix

Description: The *distribution_services_suffix* contains no information and marks the end of the *dist_transport_MU*, *dist_report_MU*, or *dist_continuation_MU*.

Dist_Report_MU

Description: The *distribution_report_message_unit* carries information reporting on the state of the distribution. Typically, for a multiple destination distribution, a *dist_report_MU* will report on only a portion of the distribution. The report is delivered to the report-to destination if one was specified in the reported-on DTMU; otherwise, it is delivered to the distribution originator.

Length Restriction: The minimum length of a *dist_report_MU* originated by an FS2 DSU is 78 bytes. This is due to the length restriction on the *Report_DTM*.

Report_Prefix

Description: The *report_prefix* identifies the beginning of *dist_report_MU*. This structure carries information that changes from DSU to DSU.

Report_Command

Description: The *report_command* contains the control information for the distribution report.

Length Restriction: The minimum length of a *dist_report_MU* originated by an FS2 DSU is 26 bytes (including its header). This is due to the length restriction on the *report_DTM*.

Reporting_DSU

Description: The *reporting_DSU* is the name of the DSU that generated the report.

Reporting_RGN

Description: The *reporting_RGN* is the first part of the name of the DSU that generated the report. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reporting_REN

Description: The *reporting_REN* is the second part of the name of the DSU that generated the report. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Report_DTM

Description:	The *report_date-time* contains the date and time at which the reporting DSU generated the report. FS2 implementations support the offset from local time to GMT.
Length Restriction:	Originating FS2 DSUs never generate a local-only time (implying a minimum length of 11 bytes - including its header). However, if the value within Report_DTM had been mapped from an FS1 subnetwork, it would have a length of 10 bytes (including its header).
Format:	Byte string

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X'07C5')
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
10	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X'E9' (z) TIME is GMT and no offset required. X'4E' (+) TIME is local, OFFSET VALUE is required, and encoder's local time is ahead of GMT. X'60' (-) TIME is local, OFFSET VALUE is required, and encoder's local time trails GMT. Note: All other values are reserved.
	OFFSET VALUE
11	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 13 are valid)
12	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 59 are valid)

Note: Refer to "Representing Date and Time" on page 15-38 for a complete discussion of the encoding and interpretation of date and time.

Report-To_DSU_User

Description: The *report-to_DSU_user* is the DSU or user to which the distribution report is being sent.

Report_Information

Description: The *report_information* identifies the distribution (or portion thereof) being reported on.

Reported-On_Origin_DSU

Description: The *reported-on_origin_DSU* is the name of the DSU at which the distribution was originated.

Presence Rules: If *reported-on_origin_DSU* is present, and *reported-on_origin_user* is absent, then the distribution was originated by a DSU; if *reported-on_origin_user* is present and *reported-on_DSU* is absent, then the report either originated in or passed through an FS1 subnetwork. If both *reported-on_origin_DSU* and *reported-on_origin_user* are present, then the report is not going to the originator of the distribution; if both *reported-on_origin_DSU* and *reported-on_origin_user* are absent, then they default to *report-to_DSU* and, if applicable, *report-to_user.*

Reported-On_Origin_RGN

Description: The *reported-on_origin_RGN* is the first part of the DSU name at which the distribution originated. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reported-On_Origin_REN

Description: The *reported-on_origin_REN* is the second part of the DSU name at which the distribution originated. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reported-On_Origin_User

Description: The *reported-on_origin_user* is the name of the user that originated the distribution.

Presence Rules: If *reported-on_origin_DSU* is present, and *reported-on_origin_user* is absent, then the distribution was originated by a DSU; if *reported-on_origin_user* is present and *reported-on_DSU* is absent, then the report either originated in or passed through an FS1 subnetwork. If both *reported-on_origin_DSU* and *reported-on_origin_user* are present, then the report is not going to the originator of the distribution; if both *reported-on_origin_DSU* and *reported-on_origin_user* are absent, then they default to *report-to_DSU* and, if applicable, *report-to_user*.

Reported-On_Origin_DGN

Description: The *reported-on_origin_DGN* is the first part of the name of the user that originated the distribution.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Reported-On_Origin_DEN

Description: The *reported-on_origin_DEN* is the second part of the name of the user that originated the distribution.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reported-On_Seqno_DTM

Description:	The *reported-on_sequence_number/date-time*, in combination with the origin agent, origin DSU, and origin user, is the unique identifier of the distribution. The origin agent, origin DSU, and origin user are specified in the appropriate reported-on or report-to structures. The sequence number is the number assigned to the distribution by the origin agent. The date-time is the date and time generated at the origin of the distribution. FS2 implementations support the offset from local time to GMT.
Length Restriction:	Originating FS2 DSUs never generate a local-only time. The minimum length for *reported-on_seqno_DTM* is therefore 15 bytes (including its header).
Format:	Byte string
Range Of Values	Valid values for sequence number portion of the *reported-on_seqno_DTM* range from 1 to $2^{31}-1$.

Byte	Content
0-1	LT header
	SEQNO
2-5	Signed binary integer ranging from 1 to 2^{31}-1
	DATE
6-7	Year, in binary (e.g., 1989 is encoded as X'07C5')
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 23 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
14	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X'E9' (z) TIME is GMT and no offset required. X'4E' (+) TIME is local, OFFSET VALUE is required, and encoder's local time is ahead of GMT. X'60' (-) TIME is local, OFFSET VALUE is required, and encoder's local time trails GMT. Note: All other values are reserved.
	OFFSET VALUE
15	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 13 are valid)
16	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 59 are valid)

Note: Refer to "Representing Date and Time" on page 15-38 for a complete discussion of the encoding and interpretation of date and time.

Reported-On_Supp_Dist_Info1

Description:	The *reported-on_supp_dist_info1* structure is reserved for future use.
Format:	Character string

Reported-On_Agent_Correl

Description:	The *reported-on_agent_correlation* is a string that was supplied by the origin agent at the origin DSU.
Format:	Undefined byte string

Reported-On_Origin_Agent

Description:	The *reported-on_origin_agent* is the name of the transaction program at the origin DSU that originated the distribution that is being reported on.
Presence Rule:	Occurs when *report-to_agent* is different from *origin_agent*. If third-party reporting has been requested and a report was generated in or flowed through an FS1 sub-network, the *reported-on_origin_agent* structure is discarded.
Format:	Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Reported-On_Dest_Agent

Description:	The *reported-on_destination_agent* is the name of the transaction program at the destination DSU that was specified for the reported-on distribution.
Presence Rule:	Occurs when *dest_agent* was specified in the reported-on DTMU.
Format:	Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00' to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA registered. X'40' is not a valid first-byte value.

Reported-On_Supp_Dist_Info2

Description:	The *reported-on_supp_dist_info2* structure is reserved for future use.
Format:	Undefined byte string

Dist_Continuation_MU

Description: The *distribution_continuation_message_unit* is used by a sending DSU to continue transmission of a suspended MU.

Continuation_Prefix

Description: The *continuation_prefix* identifies the beginning of a DCMU.

Restarting_Byte_Position

Description: The *restarting_byte_position* indicates where the sender is beginning retransmission of the first structure being re-sent. The byte count begins with the first byte of atomic data (i.e., no LLs included) within the encompassing structure. Absence of this structure is equivalent to the presence of a 1 in this structure, implying that the first structure present in the DCMU is being re-sent in its entirety. 0 is not allowed.

Format: Unsigned binary integer

Range of Values: Valid values range from 1 to 2^{64}-2.

Sender_Exception_MU

Description: The *sender_exception_MU* is sent from the sender to the receiver when the sender detects an exception while sending a *dist_transport_MU*, a *dist_report_MU*, or a *dist_continuation_MU*.

Receiver_Exception_MU

Description: The *receiver_exception_MU* is sent from the receiver to the sender when the receiver detects an exception while receiving a *dist_transport_MU*, a *dist_report_MU*, or a *dist_continuation_MU*.

Receiver_Exception_Command

Description: The *receiver_exception_command* is the prefix identifying the *receiver_exception_MU*.

Sender_Retry_Action

Description: The *sender_retry_action* is the receiver's recommendation to the sender as to whether to retry the transmission of the MU.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2	X'01' RETRY_PRECLUDED X'02' RETRY_ALLOWED X'03' RETRY_EXPECTED_USING_DCMU Note: All other values are reserved.

Receiving_DSU

Description: The *receiving_DSU* is the name of the DSU to which a distribution was being sent.

Receiving_RGN

Description: The *receiving_RGN* is the first part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Receiving_REN

Description: The *receiving_REN* is the second part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed. If a product chooses to implement DGN = REN, the enhanced character string (ECS) subset is implemented in a particular network, and a DGN exists that contains an ECS character that is not an element of CGCSGID 01134-0500, then ECS characters may occur in this structure.

Completion_Query_MU

Description: The *completion_query_message_unit* is sent by the sending DSU to query the completion status of a particular MU at the receiving DSU.

Completion_Report_MU

Description: The *completion_report_message_unit* is sent by the receiving DSU to report on the completion status of a particular MU or to control traffic flow on a conversation.

Indicator_Flags

Description: The *indicator_flags* structure contains a 1-byte flag, to indicate the completion status of the *MU_ID* identified in a *completion_report_MU*, or to control traffic flow on a conversation.

Format: Bit string

Note: Conversation control flags (bits 2 and 3) may be used in conjunction with flow control flags (Not Received, In Transit, Suspended, Terminated, Completed, Purged).

Bit Map								Architecturally-Defined Value
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	
x	x	0	0	x	x	x	x	Default—Normal SNA/DS flow
x	x	0	1	x	x	x	x	Terminate Conversation
0	x	x	x	0	0	0	0	Not Received
0	x	x	x	0	0	0	1	In Transit
0	x	x	x	0	0	1	0	Suspended
0	x	x	x	0	0	1	1	Completed
0	x	x	x	0	1	0	1	Terminated
1	x	x	x	x	x	x	x	Purged

Note: x = any value

Last_Structure_Received

Description: The *last_structure_received* is the codepoint of the structure the receiving DSU identifies as the last structure received before the MU was suspended. This structure must be a length-bounded LLID structure at the highest level of the MU.

Presence Rule: If *indicator_flags* = SUSPENDED, then *last_structure_received* is present.

Format: Hexadecimal code

Last_Byte_Received

Description:	The *last_byte_received* is the last byte received by the receiving DSU before the MU was suspended. The byte count begins with the first byte of atomic data within the encompassing structure. The byte count contains only atomic data and does not contain the segmenting LLs for segmented structures. A byte count of X'0000000000000000' indicates that only the LLID of the structure was received (i.e., that any following atomic data was either not received or lost). A byte count of X'FFFFFFFFFFFFFFFF' indicates that the structure was fully received.
Presence Rules:	If *indicator_flags* = SUSPENDED, *last_structure_received* is present, and *last_byte_received* is absent, then the structure was received.
Format:	Unsigned binary integer
Range of Values:	Valid values range from 0 to 2^{64}-1, where the values 0 and 2^{64}-1 have the meanings defined above.

Purge_Report_MU

Description:	The *purge_report_message_unit* indicates to the receiving DSU that the sending DSU has marked a particular *MU_ID* PURGED, and that the receiving DSU may flag that *MU_ID* as PURGED.

Reset_Request_MU

Description:	The *reset_request_message_unit* is sent from DS_Send to DS_Receive. DS_Send issues the *reset_request_MU* to request that DS_Receive reset its *MU_ID* registry.

Reset_DTM

Description:	The *reset_date-time* contains the date and time at which the *reset_request_MU* was generated. Both sender and receiver store it as the "time of last reset" of their *MU_ID* registries.
Length Restriction:	Originating FS2 DSUs never generate a local-only time. The minimum length for *reset_DTM* is 11 bytes (including its header).
Format:	Byte string

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X'07C5')
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
10	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X'E9' (z) TIME is GMT and no offset required. X'4E' (+) TIME is local, OFFSET VALUE is required, and encoder's local time is ahead of GMT. X'60' (-) TIME is local, OFFSET VALUE is required, and encoder's local time trails GMT. local time Note: All other values are reserved.
	OFFSET VALUE
11	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 13 are valid)
12	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 59 are valid)

Note: Refer to "Representing Date and Time" on page 15-38 for a complete discussion of the encoding and interpretation of date and time.

Reset_Accepted_MU

Description: The *reset_accepted_message_unit* is sent from DS_Receive to DS_Send. DS_Receive issues the *reset_accepted_MU* in response to a *reset_request_MU* to inform DS_Send that DS_Receive has reset its MU_ID Registry.

Unrecognized_Reserve

Description:	The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure. Intermediate DSUs pass *unrecognized_reserve* structures through unchanged in outgoing DMUs.
Format:	Undefined byte string

Representing Date and Time

Following is a discussion of the date and time formats recognized and supported by SNA/DS. Definitions and examples are also provided that illustrate the encoding and interpretation of each format.

Generalized Time Building Blocks

Generalized time is a term that is used to refer to a very general representation of time. It is comprised of a calendar date, a time of day, and, optionally, an offset from that time of day to some common time of day (i.e., GMT/UTC). SNA/DS encodes and interprets generalized time in numerous message units.

Time Formats Supported by SNA/DS

SNA/DS can encode and interpret the following generalized time formats:

Local-Only Time Local-only time is encoded as a date and a base local time (e.g., date = May 31, 2001; time = 11:22:33.44 p.m.).

If the local-only time format is encoded to represent generalized time, the interpreter cannot relate that time to GMT and, hence, to its own local time unless it has awareness of the encoder's relationship to GMT/UTC.

GMT-Only Time GMT-only time is encoded as a date and base GMT time followed by a time flag (the character "Z"). A time flag with value "Z" signals the interpreter that the base time is GMT, not local time (e.g., May 31, 2001; time = 06:11:22.33; Flag = Z).

If the GMT-only time format is encoded to represent generalized time, the interpreter can successfully relate that time to its own local time, but cannot determine the encoder's local time unless it has awareness of the encoder's relationship to GMT/UTC.

Offset Time Offset time is encoded as a date and base local time followed by a *signed offset*. The signed offset indicates the time differential between the base time and GMT/UTC. The sign of the signed offset also acts as the time flag. Hence, the offset time format can be identified when the time flag is either a "+" sign or "-" sign.

The sign of the offset is based on the encoder's location relative to GMT/UTC. Those locations that are just east (e.g., ahead) of GMT use a "+" offset (time flag) whereas those locations just west of (e.g., behind) GMT use a "-" offset sign (e.g., offset for New York would be Flag="-"; Offset = 5 hours).

If the GMT-only time format is encoded to represent generalized time, the interpreter can successfully relate that time to GMT and, hence, its own local time.

Encoding Generalized Times

The encodings for generalized time are usually included in an encoding structure that has an LT header and, possibly, some other substructure. (The *Seqno_DTM* structure on page 15-15, for example, contains an LT header and a *sequence_number* substructure in addition to the generalized time substructure). The following encodings define the generalized time substructure only.

Byte	Content
	DATE
1-2	Year, in binary (e.g., 1989 is encoded as X'07C5')
3	Month of the year, in binary (values from 1 to 12 are valid)
4	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
5	Hour of the day, in binary (values from 0 to 23 are valid)
6	Minute of the hour, in binary (values from 0 to 59 are valid)
7	Second of the minute, in binary (values from 0 to 59 are valid)
8	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
9	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X'E9' (z) TIME is GMT and no offset required. X'4E' (+) TIME is local, OFFSET VALUE is required, and encoder's local time is ahead of GMT X'60' (-) TIME is local, OFFSET VALUE is required, and encoder's local time trails GMT Note: All other values are reserved.
	OFFSET VALUE
10	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 13 are valid)
11	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 59 are valid)

Interpreting Time Formats

In order to properly interpret an encoded generalized time, the interpreter must understand whether the base encoded time is local or GMT and what formula to use to convert between local and GMT times.

SNA/DS interprets encoded generalized times based on the following rules:

- If the time flag exists and has value *"Z"*, then the base time is GMT. Otherwise, the base time is local time.
- Conversion between local and GMT/UTC times is performed via the formula: *GMT_time = local_time - signed_offset.*

Examples

Following are two examples that illustrate how generalized time is encoded and interpreted by SNA/DS. For each of these examples, assume DSU A is located in New York City (5 hours behind GMT) and DSU B is in Tokyo, Japan (9 hours ahead of GMT).

1. DSU A sends an MU to DSU B at 11:22:33.44 p.m. NYC time on May 31, 2001. DSU A may encode its generalized time in either the local-only, GMT-only, or offset time formats. Following are the encodings for each generalized time format (with the character equivalent of each encoding provided to improve understanding of the applicable hex encoding):

   a. Local-only time:

   ```
   X'07D1051F1716212C'
     yyyyMMddHHmmsshh
   C'2001053123223344'
   ```

   Note: A SNA/DS FS2 DSU will not encode the local-only time format (see *Seqno_DTM* length restriction on page 15-15).

   b. GMT-only time:

   ```
   X'07D106010416212CE9'
     yyyyMMddHHmmsshhF  (June 1, 2001 at 4:22:33.44 a.m. GMT)
   C'2001060104223344Z'
   ```

   c. Offset time:

   ```
   X'07D1051F1716212C600500'
     yyyyMMddHHmmsshh- HHmm
   C'2001053123223344- 0500'
   ```

   If DSU A encoded the offset time format, DSU B can successfully relate the supplied generalized time to its own local time by:

   - converting the offset time to GMT via the formula:

     ```
     GMT_time = base_time - (signed_offset)
     ```

   - converting GMT to its relative local time via the formula:

     ```
     relative_local_time = GMT_time + (signed_offset_of_interpreter)
     ```

   Using these formulas (which are both based on the interpreter's formula discussed earlier), DSU B can interpret DSU A's local time to be equal to:

   ```
   X'07D106010D16212C'
     yyyyMMddHHmmsshh   (June 1, 2001 at 1:22:33.44 p.m. Tokyo time)
   C'2001060113223344'
   ```

2. DSU B sends an MU to DSU A on January 1, 2000 at 7:00 a.m. Tokyo time and encodes its generalized time in offset format as follows:

   ```
   X'07D0010107000000600900'
     yyyyMMddHHmmsshhF HHmm
   C'2000010107000000+ 0900'
   ```

   DSU A can interpret this time as either:

```
X'07CF0C1F16000000E9'
  yyyyMMddHHmmsshhF   (December 31, 1999 at 10:00 p.m. GMT), or as
C'19991231220000000Z'

X'07D00C1F05000000'
  yyyyMMddHHmmsshh    (December 31, 1999 at 5:00 p.m. NYC time)
C'19991231170000000'
```

Transaction Program and Server Names

Following is a list of all transaction program and server names defined for SNA/DS in the FM header 5 (Attach), in the Distribution MU, or used internally in the distribution service unit (DSU).

Code	Meaning
X'20F0F0F0'	DIA process destination transaction program name
X'20F0F0F1'	DIA server name
X'20F0F0F2'	DIASTATUS transaction program name
X'21F0F0F1'	DS_SEND transaction program name (FS1)
X'21F0F0F2'	DS_RECEIVE transaction program name (FS1)
X'21F0F0F3'	DS_ROUTER_DIRECTOR transaction program name
X'21F0F0F6'	SNA/DS general server name
X'21F0F0F7'	DS_SEND transaction program name (FS2)
X'21F0F0F8'	DS_RECEIVE transaction program name (FS2)
X'23F0F0F0'	SNA/MS Change Management agent TP name
X'24F0F0F0'	SNA/File Services server name
X'30F0F0F2'	Object Distribution transaction program.
X'30F0F0F3'	Object Distribution server transaction program.

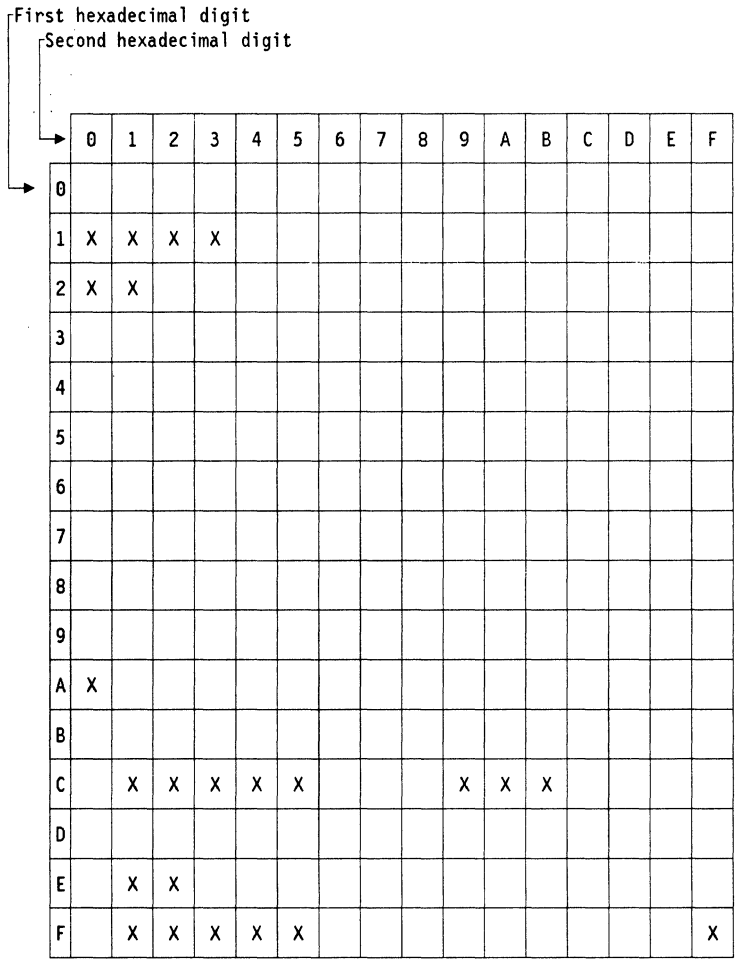
Code Points Used by SNA/DS FS2

The values of the ID component of the LLID structure as used for SNA/DS GDS variables are shown below:

ID	Structure Name	Applicable MUs
1532	SNA Condition Report	*DRMU, REMU*
1570	Transport Prefix	*DTMU*
1571	Transport Command	*DTMU*
1572	Destination List	*DTMU*
1573	Agent Object	*DTMU, DRMU, DCMU*
1574	Server Object	*DTMU, DCMU*
1575	Report Command	*DRMU*
1576	Report Information	*DRMU*
1577	Receiver Exception Command	*REMU*
1578	Sender Exception Message Unit (type FS2)	*SEMU*
1579	Completion Query Message Unit	*CQMU*
157A	Completion Report Message Unit	*CRMU*
157B	Continuation Prefix	*DCMU*
157C	Report Prefix	*DRMU*
157E	Purge Report Message Unit	*PRMU*
157F	Suffix	*DTMU, DRMU, DCMU*
1580	External Network Correlation	*DTMU, DCMU*
1581	External Network Object	*DTMU, DCMU*
1582	Reported-On External Network Correlation	*DRMU*
1583	Report-To DSU/User	*DRMU*
1585	Reset Request Message Unit	*RRMU*
1586	Reset Accepted Message Unit	*RAMU*

Chapter 16. GDS Variables for Application Programs

The following chart indicates (using an "X") each GDS variable code point (with first byte = X'12') used by LU 6.2.



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1	X	X	X	X												
2	X	X														
3																
4																
5																
6																
7																
8																
9																
A	X															
B																
C		X	X	X	X	X				X	X	X				
D																
E		X	X													
F		X	X	X	X	X										X

Figure 16-1. LU Type 6.2 GDS Variable Code Points

The code points used by LU 6.2 are:

X'1210'	Change Number of Sessions (see Note 1)
X'1211'	Exchange Log Name (see Note 1)
X'1212'	Control Point—Management Services Unit
X'1213'	Compare States (see Note 1)
X'1220'	Sign Off (see Note 1)
X'1221'	Sign On (see Note 1)
X'12A0'	Workstation Display Passthrough
X'12C1'	CP Capabilities (see Note 1)
X'12C2'	Topology Database Update (see Note 1)
X'12C3'	Register Resource (see Note 1)

X'12C4'	Locate (see Note 1)
X'12C5'	Cross-Domain Initiate (see Note 1)
X'12C9'	Delete Resource (see Note 1)
X'12CA'	Find Resource (see Note 1)
X'12CB'	Found Resource (see Note 1)
X'12E1'	Error Log
X'12E2'	PIP Subfield Data (see Note 2)
X'12F1'	Null Data
X'12F2'	User Control Data
X'12F3'	Map Name
X'12F4'	Error Data
X'12F5'	PIP Data (see Note 2)
X'12FF'	Application Data

Notes:

1. See Chapter 13, "GDS Variables for SNA Service Transaction Programs (STPs)" for the formats and meanings of these GDS variables.
2. See Chapter 11, "Function Management (FM) Headers" for the formats and meanings of these GDS variables.

Error Log (X'12E1') GDS Variable

The Error Log GDS variable, ID X'12E1', following an FMH-7 conveys implementation-specific error information to an LU, where it is added to the system error log for use in debugging and error recovery. It is not used by SNA-defined service transaction programs (other than to log it) since it contains implementation-specific data. The Error Log variable is sent as a consequence of issuing the SEND_ERROR verb, but is not passed to the receiving transaction program. Its format is:

Error Log (X'12E1') GDS Variable

Byte	Bit	Content
0 – 1		Length (n + 1), in binary, of Error Log GDS variable, including this Length field
2 – 3		GDS ID: X'12E1'
4 – m		Product Set ID
4 – 5		Length, in binary, of Product Set ID, including this Length field (values 2 to 32,767 are valid) *Note:* The Length field is always present; a value of 2 indicates no Product Set ID subvector follows.
6 – m		Product Set ID (X'10') subvector (format described in Chapter 8, "Common Fields")
m + 1 – n		Message Text

Error Log (X'12E1') GDS Variable

Byte	Bit	Content
m+1–m+2		Length, in binary, of message text, including this Length field (values 2 to 32,767 are valid) *Note:* The Length field is always present; a value of 2 indicates no message text follows.
m+3–n		Message text data: implementation-specific data

Null Data (X'12F1') GDS Variable

The Null Data GDS variable, ID X'12F1', contains no application data. This variable may optionally be generated to carry certain control information (e.g., Confirm) when no application data is available.

User Control Data (X'12F2') GDS Variable

The User Control Data GDS variable, ID X'12F2', contains user control data. The meaning of this data is known only to the LU services component programs or the transaction programs and their mapping programs. This data can be used, for example, as prefix control information for an Application Data GDS variable that follows it or to carry FM header data for a mapped conversation transaction.

Map Name (X'12F3') GDS Variable

The Map Name GDS variable, ID X'12F3', is followed by a 0- to 64-byte map name.

Error Data (X'12F4') GDS Variable

The Error Data GDS variable, ID X'12F4', is used to convey information about mapping errors. It is sent using the SEND_DATA verb following a SEND_ERROR verb. Its format is:

Error Data (X'12F4') GDS Variable

Byte	Bit	Content
0–1		Length (n+1), in binary, of Error Data GDS variable, including this Length field
2–3		GDS ID: X'12F4'
4–7		Error code: X'00010000' Invalid GDS ID: The mapped conversation verb component encountered a GDS ID that it did not recognize. X'00030001' Map Not Found: The specified map was not available at the target, or access to the referenced map could not be completed. X'00030002' Map Execution Failure: The map program was not able to process the data stream.
8		Length (n-8), in binary, of error parameter

Error Data (X'12F4') GDS Variable

Byte	Bit	Content
9–n		Error parameter: for a mapping failure, the map name carried in the GDS variable for which the error occurred; for an invalid GDS ID, the 2-byte GDS ID that was not recognized

Application Data (X'12FF') GDS Variable

The Application Data GDS variable, ID X'12FF', contains application data. The application transaction program's data as specified in the MC_SEND_DATA verb is (optionally) mapped and then sent as X'12FF' variables.

Chapter 17. SNA/File Services (FS)

This chapter contains the format descriptions for the SNA/FS data streams. The format descriptions are comprised of two parts, header description tables and structure descriptions. A header description table contains the header information for each structure. A structure description contains a prose description of the structure, bit-level representations, and any presence rules or length restrictions associated with a particular structure.

Encoding Rules and Representations

The definition of SNA/FS requires a byte-accurate description of the formats that must be understood by all SNA/FS-capable agents and servers. The SNA/FS formats are described in terms of encoded fields referred to as "structures" and the hierarchical relationship between these structures. In this appendix, the header description tables show each structure and its header. Elsewhere in this book, the header length is assumed not to be part of the overall structure length (e.g., *SNA_report_code*).

Structure Classifications

Fields and groupings of fields are known as structures. They are categorized in terms of their hierarchical position ("atomic," "child," or "parent"), the method by which their beginning and endings are determined, (length-bounded, delimited, or implied) and which kind of header is used to identify them (LT or LLID). Only certain combinations of characteristics are possible.

Length-Bounded Structures

Length-bounded structures consist of a header and usually some following information. A header may be either two bytes in length, referred to as an "LT" (length and type), or four bytes in length, referred to as an "LLID" (length and GDS code point). In either case, the length byte(s) include the length of the header itself and the following information, if any.

Atomic Structures

In many cases, a structure consists only of its own header followed by data. These structures cannot be decomposed, and therefore they are called "atomic." Atomic structures are always length-bounded and may have either LT or LLID headers.

Parent and Child Structures

Structures can contain other structures within them. The containing structure is known as a parent structure and the contained structures are known as children. These terms are relative, since a non-atomic child structure itself contains other structures and is a parent to them. Children of the same parent are siblings of each other. Parent structures may be length-bounded, delimited, or implied; and may be identified by LTs or LLIDs.

Length-Bounded Parent Structures

In this case, the parent structure has its own header, either an LT or an LLID. Its length includes the lengths of all its children plus the length of its own header. A length-bounded parent exists both as a logical grouping of its children and as an explicit encoded structure at its own encoding level.

Delimited Parent Structures

Sometimes it is convenient to define a group of related structures as existing within a parent structure without having that parent structure appear as a length-bounded structure in the message. The beginning and end of the parent are defined by its first and last children. These children are known as delimiters, the first child is the prefix delimiter and the last is the suffix delimiter. Delimiter children are length-bounded and must be present. They may be null, that is, with an LT of length = 2 or an LLID of length = 4. When the children's headers are LTs, the parent is classified as a delimited LT structure. When they are LLIDs, the parent is a delimited LLID structure.

Implied Parent Structures

It is possible to define a set of related structures as children of a parent structure where the existence and boundaries of the parent are implied by the existence and order of certain child structures. This set of children may occur within the parent structure, either ordered or unordered, until a structure occurs that is not an element of this set. This break in sequence implies the boundary between parent structures. Depending on its children's headers, an implied parent is classified as either implied LT or implied LLID.

Segmented Structures

Length-bounded LLID structures may be either segmentable or non-segmentable. For segmentable structures, the most significant bit of the LL bytes indicates whether any particular segment is the last (bit is equal to 0) or not last (bit is equal to 1) segment of the structure. The ID bytes of the segmentable structure are present on the first segment only.

Properties of Parent Structures

Order

A parent structure may have either ordered or unordered children. Ordered children occur in the parent structure in the same order as they are described in the format description table. Unordered children may occur in the parent structure in any order.

Unrecognized Children

Future enhancements to the formats might add structures that will not be recognized by implementations of the current format definitions. The current format must specify for each parent whether or not unrecognized child structures are allowed. If they are allowed, the definition must specify how long they might be. When unrecognized structures are found where they are allowed, they must be passed through without change at intermediate locations and gracefully ignored at final destinations. Unrecognized structures are identified by either LT or LLID headers, being of the same type as their siblings.

Number of Children

The number of children within a parent may range from a required minimum to an allowed maximum. For example, a parent might have several children, each defined with an occurrence of 0-1, and a number of children defined as 1. This means that any one, but only one, child is allowed.

Header Description Table

The header information and primary syntax associated with each structure are formally described in tabular form. These header description tables represent the formatting information required to either parse or build SNA/FS structures.

Structure Name

The first column of the header description table identifies SNA/FS structures, by name, and illustrates their hierarchical relationship by indentation of the column entries. The order of the structure entries in the table represents, unless specified otherwise, the order in which the structures appear in the SNA/FS datastream.

Structure Reference (Struct Ref)

As header information and primary syntax are described in the header description of a particular table, the semantics, bit representations, presence rules, and other characteristics are described formally in the structure description. This column contains a reference page number to where this structure information is found.

Structure Class (Struct Class)

Structures are classified as either length-bounded LLIDs (ID), length-bounded LTs (T), delimited LLIDs (Del-ID), delimited LTs (Del-T), implied LLIDs (Imp-ID), or implied LTs (Imp-T).

A structure classified as delimited must contain at least two required, length-bounded children that act as the prefix (pfx) and suffix (sfx) of the delimited structure. The "/pfx" notation indicates the length-bounded child structure that serves as the prefix for its parent delimited structure. The "/sfx" notation indicates the length-bounded structure that serves as the suffix for its parent delimited structure.

A structure classified as implied uses an identified child to identify the beginning of a sequence of children. The "/idc" notation indicates the length-bounded structure that serves as an identified child of its parent implied structure.

The "/seg" notation indicates that segmentation is allowed.

ID/T

This column contains the ID or T value within the header, in hexadecimal. To indicate that a delimited structure is identified by its prefix, the notation "pfx" is used. To indicate that an implied structure is identified by one of its children, the notation "idc," for identified child, is used.

Length

This column describes the length verification that would be appropriate at presentation services time. The range of length values specifies the minimum and maximum lengths of structures which an implementation is required to receive. For structures that allow unrecognized children, the maximum length value accommodates the possibility of these yet-to-be-defined structures. On the sending side, the maximum length value for a particular structure may be determined by subtracting the unrecognized reserve, if unrecognized children are allowed, from the maximum length.

Note: An asterisk denotes length restrictions for a particular structure. Length restrictions are detailed in the corresponding structure description.

Occurrences

Multiple occurrences of SNA/FS structures may or may not be permitted. A value of "1 - < some number >" in this column indicates the allowed range of occurrences of the corresponding structure. A value of "≥1" indicates that there is no architecturally defined maximum. A value of "1" in this column indicates that only a single instance of the corresponding structure is appropriate. A value of "0 - 1" indicates that an instance of the corresponding structure is optional.

Note: An asterisk denotes presence rules for a particular structure. Presence rules are detailed in the corresponding structure description.

Children

Unrecognized Children Allowed (Unrec): An entry of "Y" in the "Unrec" column indicates that the corresponding structure tolerates unrecognized child structures. An entry of "N" indicates that the particular structure tolerates only the architecturally-defined child structures. An entry of "—" indicates that unrecognized children are not applicable to the particular structure. By definition, atomic structures do not contain children, recognized or not.

Order: A value of "Y" in this column indicates that children are ordered, a value of "N" indicates that children are unordered, and a value of "—" indicates that no children are present.

Note: If a structure is atomic, this column is not applicable.

Number (Num): Each parent structure contains a certain number of different children. This column specifies the minimum and maximum number of different children for a particular parent structure. The maximum number also accounts for unrecognized children, if they are allowed within the parent structure. This column does not account for multiple occurrences of a particular child structure within the parent structure. The number of occurrences of each child is indicated in the "Occurrences" column.

Subtable: Sometimes the need to divide large tables into subtables becomes apparent, particularly when common children appear frequently within different header description tables. This column contains a reference page number to where these common children are described.

Structure Description

The structure description is referenced by a page number appearing in the "Structure Reference" column corresponding to each structure in the header description table. This description contains information pertaining to the data portion of a particular structure. Prose descriptions, presence rules, and semantics associated with the corresponding entry in the header description table may appear in the structure description.

SNA/FS Usage of SNA/DS Encodings

SNA/FS requires the services of SNA/DS implementations to transport SNA/FS encodings between SNA/FS-capable DSUs. The SNA/DS architecture is able to transport SNA/FS-defined encodings within three different SNA/DS-defined envelopes. The SNA/DS *agent_correl* envelope is used by SNA/FS to identify the SNA/FS unit-of-work. All SNA/DS distributions relating to one particular SNA/FS unit-of-work will carry the same *agent_correl* envelope. The SNA/DS *agent_object* envelope is used by SNA/FS to carry agent commands targeted for SNA/FS-capable agents. The SNA/DS *server_object* is used by SNA/FS to carry server instructions and data objects targeted for SNA/FS servers. An SNA/FS unit-of-work may require either or both of these two types of objects.

SNA/FS Requests and Reports

An SNA/FS unit-of-work may result in multiple SNA/DS distributions. These SNA/DS distributions can carry either an SNA/FS request or an SNA/FS report. An SNA/FS request solicits SNA/FS services from agents and/or servers at other DSUs. An SNA/FS report describes the relative success of the SNA/FS agent/server in performing a requested function. Since the distinction is significant from an encoding perspective, SNA/FS requests and SNA/FS reports are described in separate header description tables.

Header Description Tables for SNA/FS Encodings

Unit of Work Correlator

Figure 17-1. The SNA/FS Use of the SNA/DS Agent_Correl

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Agent_Unit_of_Work	C-19	ID	1549	27-128	1	Y	Y	2-8	—
U_of_W_Requester_DSU	C-19	T	01	8-22	1	N	Y	2	—
U_of_W_Requester_RGN	C-19	T	01	3-10	1	—	—	—	—
U_of_W_Requester_REN	C-19	T	02	3-10	1	—	—	—	—
U_of_W_Requester_User	C-19	T	03	8-22	0-1	N	Y	2	—
U_of_W_Requester_DGN	C-20	T	01	3-10	1	—	—	—	—
U_of_W_Requester_DEN	C-20	T	02	3-10	1	—	—	—	—
U_of_W_Requester_Agent	C-20	T	04	3-10	0-1*	—	—	—	—
U_of_W_Seqno_DTM	C-20	T	02	15-17	1	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-53	—	—	—	—	—

Note: * Refer to the structure description for presence rule(s).

SNA/FS Agent Request

Figure 17-2. The SNA/FS Use of the SNA/DS Agent_Object for Agent Requests

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
FS_Agent_Request	17-18	ID	1530	9-13321	1	N	Y	1-2	—
Command	17-18	ID	1546	5	1	—	—	—	—
Command_Parms	17-18	ID	1547	7-13312	0-1	Y	N	1-15	—
Source_Reporting_Action	17-19	T	02	3	0-1*	—	—	—	—
Target_Agent	17-19	T	03	3-10	0-1*	—	—	—	—
Target_Reporting_Action	17-19	T	04	3	0-1*	—	—	—	—
Report-To_Agent	17-20	T	05	3-10	0-1*	—	—	—	—
Report-To_DSU	17-20	T	07	8-22	0-1*	N	N	2	—
Report-To_RGN	17-21	T	08	3-10	1	—	—	—	—
Report-To_REN	17-21	T	09	3-10	1	—	—	—	—
Report-To_User	17-21	T	0A	8-22	0-1	N	N	2	—
Report-To_DGN	17-21	T	0B	3-10	1	—	—	—	—
Report-To_DEN	17-22	T	0C	3-10	1	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-13238	—	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

SNA/FS Server Request

Figure 17-3. The SNA/FS Use of the SNA/DS Server_Object for Server Requests

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
FS_Server_Request	17-22	Del-ID	pfx	≥28	1	Y	Y	3-11	—
FS_Server_Request_Prefix	17-22	ID/pfx	1531	8-19	1	N	Y	1-3	—
Decoder_Instruction	17-22	T	01	4-5	0-1*	—	—	—	—
Source_Instruction	17-22	T	02	4-5	0-1*	—	—	—	—
Target_Instruction	17-23	T	03	4-5	0-1*	—	—	—	—
Data_Object_Group	17-24	Del-ID	pfx	≥16	1	N	Y	3-5	—
Group_Prefix	17-24	ID/pfx	1533	4	1	—	—	—	—
Supplemental_FS_Info1	17-25	ID	1534	4-1024	0-1	Y	Y	1-9	—
Unrecognized_Reserve	C-15	T	—	2-1020	—	—	—	—	—
Supplemental_FS_Info2	17-25	ID	1535	8-32767	0-1	Y	Y	1-15	—
Supplemental_FS_Info3	17-25	ID	153C	9-283	0-1	—	—	—	—
Supplemental_FS_Info4	17-25	ID	1550	12-2048	0-1	—	—	—	—
Unrecognized_Reserve	C-15	ID	—	4-30432	—	—	—	—	—
Data_Object	17-25	Del-ID	pfx	≥18	1	Y	Y	3-19	—
D_O_Prefix	17-25	ID/pfx	1536	4	1	—	—	—	—
D_O_Attributes	17-25	ID	1537	10-1024	1	Y	Y	1-9	—
D_O_Class	17-25	T	81	6	1	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-1014	—	—	—	—	—
D_O_Global_Name	17-25	ID	1538	9-283	1	N	Y	1-7	17-10
Supplemental_FS_Info5	17-25	ID	1539	12-2048	0-1	—	—	—	—
D_O_Description	17-25	ID	153B	10-512	0-1	Y	Y	1-8	17-11
D_O_Transforms	17-25	ID	153E	14-1024	0-1	Y	Y	1-7	17-11
D_O_Allocation_Info	17-26	ID	153F	14-1024	0-1*	Y	Y	1-7	17-12
D_O_Contents	17-26	ID/seg	1541	≥5	0-1*	—	—	—	—
Unrecognized_Reserve	C-15	ID	—	4-32767	—	—	—	—	—
D_O_Suffix	17-26	ID/sfx	1542	4	1	—	—	—	—
Group_Suffix	17-26	ID/sfx	1543	4	1	—	—	—	—
Unrecognized_Reserve	C-15	ID	—	4-32767	—	—	—	—	—
FS_Suffix	17-26	ID/sfx	154C	4	1	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

SNA/FS Agent Report

Figure 17-4. The SNA/FS Use of the SNA/DS Agent_Object for Agent Reports

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
FS_Agent_Report	17-26	ID	154A	14-32763	1	N	Y	2-3	—
Command	17-18	ID	1546	5	1	—	—	—	—
SNA_Condition_Report	C-9	ID	1532	10-32749	0-1*	Y	Y	1-10	C-7
FS_Action_Summary	17-26	ID	1548	5	1	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

SNA/FS Server Report

Figure 17-5. The SNA/FS Use of the SNA/DS Server_Object for Server Reports

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
FS_Server_Report	17-27	Del-ID	pfx	≥22	1	N	Y	3-4	—
FS_Server_Report_Prefix	17-27	ID/pfx	154B	8-9	1	N	Y	1	—
Decoder_Instruction	17-22	T	01	4-5	1	—	—	—	—
SNA_Condition_Report	C-9	ID/seg	1532	10-32749	0-1*	Y	Y	1-10	C-7
Data_Object_Group	17-24	Del-ID	pfx	≥16	0-1*	N	Y	3-4	—
Group_Prefix	17-24	ID/pfx	1533	4	1	—	—	—	—
Supplemental_FS_Info2	17-25	ID	1535	8-32767	0-1	Y	Y	1-7	—
Supplemental_FS_Info3	17-25	ID	153C	9-360	0-1	—	—	—	—
Supplemental_FS_Info4	17-25	ID	1550	9-2045	0-1	—	—	—	—
Unrecognized_Reserve	C-15	ID	—	4-30358	—	—	—	—	—
Data_Object	17-25	Del-ID	pfx	≥8	1	Y	Y	2-13	—
D_O_Prefix	17-25	ID/pfx	1536	4	1	—	—	—	—
D_O_Global_Name	17-25	ID	1538	9-360	1	N	Y	1-8	17-10
Supplemental_FS_Info5	17-25	ID	1539	9-2045	0-1	—	—	—	—
Unrecognized_Reserve	C-15	ID	—	4-30354	—	—	—	—	—
D_O_Suffix	17-26	ID/sfx	1542	4	1	—	—	—	—
G_Suffix	17-26	ID/sfx	1543	4	1	—	—	—	—
FS_Suffix	17-26	ID/sfx	154C	4	1	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

Subtables

Global Names

Figure 17-6. Subtable Encoding of the SNA/FS Global Name

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Global_Names									
Token_Attributes	17-27	T	01	3-12	0-1*	—	—	—	—
To_Be_Fetched_Name	17-28	T	02	5-77*	0-1*	N	Y	1-10	C-8
Fetching_Match_Flags	17-29	T	03	3-12	0-1*	—	—	—	—
To_Be_Stored_Name	17-29	T	04	5-77*	0-1*	N	Y	1-10	C-8
To_Be_Deleted_Name	17-30	T	05	5-77*	0-1*	N	Y	1-10	C-8
Deleting_Match_Flags	17-30	T	06	3-12	0-1*	—	—	—	—
Supplemental_FS_Info6	17-25	T	07	3-12	0-1*	—	—	—	—
Fetched_Name	17-31	T	08	5-77*	0-1*	N	Y	1-10	C-8
Stored_Name	17-31	T	09	5-77*	0-1*	N	Y	1-10	C-8
Deleted_Name	17-31	T	0A	5-77*	0-1*	N	Y	1-10	C-8
Reported-On_Name	17-31	T	0B	5-77*	0-1*	N	Y	1-10	C-8

Notes:

1. The *to_be_fetched_name* and a *fetched_name* are mutually exclusive.
2. The *to_be_deleted_name* and a *deleted_name* are mutually exclusive.
3. The *to_be_stored_name* and a *stored_name* are mutually exclusive.
4. This subtable is referenced by the *FS_server_request* and the *FS_server_report*.
5. * Refer to the structure description for presence rule(s) and length restriction.

Object Description

Figure 17-7. Subtable Encoding of the Group/Object Description

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Object_Description									
Object_Text_Description	17-31	T	01	14-255	0-1	N	Y	3	—
CCSID	17-32	T	01	4	1	—	—	—	—
Language_ID	17-32	T	02	5	1	—	—	—	—
Description_Text	17-32	T	03	3-244	1	—	—	—	—
Uniform_Text_Content	17-32	T	02	6	0-1	N	Y	1	—
CCSID	17-32	T	01	4	1	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-247	—	—	—	—	—

Notes:

1. This subtable is referenced by the *FS_server_request*.

Object Transforms

Figure 17-8. Subtable Encoding of the Group/Object Transforms

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Object_Transforms									
Compression_Transform	17-32	T	01	10-128	0-1*	Y	Y	1-8	—
Compressed_State	17-32	T	01	3	1	—	—	—	—
FS_Compression	17-33	T	02	5-8	0-1*	N	Y	1-2	—
Compression_Technique	17-33	T	01	3	1	—	—	—	—
Prime_Character	17-33	T	02	3	0-1*	—	—	—	—
User_Compression	17-33	T	03	12-49	0-1*	N	Y	3	—
Algorithm_Name	17-33	T	01	3-14	1	—	—	—	—
Algorithm_Parms	17-33	T	02	3-29	1	—	—	—	—
CCSID	17-32	T	03	4	1	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-74	—	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-892	—	—	—	—	—

Notes:

1. This subtable is referenced by the *FS_server_request*.
2. * Refer to the structure description for presence rule(s).

Allocation Information

Figure 17-9. Subtable Encoding of the Allocation Information

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Allocation_Info									
Transfer_Size	17-34	T	08	10	1	—	—	—	—
Record_Info	17-34	T	09	15-64	0-1	Y	Y	2	—
Record_Format	17-34	T	0A	3	1	—	—	—	—
Maximum_Record_Size	17-34	T	0B	10	1	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-49	—	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-995	—	—	—	—	—

Note: This subtable is referenced by the *FS_server_request*.

SNA Condition Report

See Appendix C, "Common Structures" on page C-1 for the SNA Condition Report. Note that the six Structure Names for the children of the Reported-On_Dest parent have different names in this chapter than appears in the referenced Appendix.

In this Chapter the names are:

- Reported-On_Dest
  - Reported-On_Dest_DSU
    - Reported-On_Dest_RGN
    - Reported-On_Dest_REN
  - Reported-On_Dest_User
    - Reported-On_Dest_DGN
    - Reported-On_Dest_DEN

In the Appendix the names are:

- Reported-On_Dest
  - Reported-On_Location_Name
    - Reported-On_NETID
    - Reported-On_Node_ID
  - Reported-On_User
    - Reported-On_Naming_Auth
    - Reported-On_Individual_ID

Structure Descriptions

Agent_Unit_of_Work

Description: The *agent_unit_of_work*, assigned by the requesting agent, provides the basis to track the progress of a particular defined task. The unit-of-work request is uniquely identified by the combination of *u_of_w_requester_DSU*, *u_of_w_requester_user*, *u_of_w_requester_agent*, and *u_of_w_sequence number|date-time*.

In SNA/FS, the unit of work identifies one or more generated SNA/DS distributions as belonging to the same SNA/FS defined task.

U_of_W_Requester_DSU

Description: The *unit_of_work_requester_DSU* is the name of the DSU at which the unit-of-work was requested.

U_of_W_Requester_RGN

Description: The *unit_of_work_requester_RGN* is the first part of the name of the DSU at which the unit-of-work was requested. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

U_of_W_Requester_REN

Description: The *unit_of_work_requester_REN* is the second part of the name of the DSU at which the unit-of-work was requested. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

Note: In existing networks where LU names are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the REN may

contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new LU names.

U_of_W_Requester_User

Description: The *unit_of_work_requester_user* is the user name of the originator of the unit-of-work request.

U_of_W_Requester_DGN

Description: The *unit_of_work_requester_DGN* is the first part of the user name of the unit-of-work originator.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

U_of_W_Requester_DEN

Description: The *unit_of_work_requester_DEN* is the second part of the user name of the unit-of-work originator.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

U_of_W_Requester_Agent

Description: The *unit_of_work_requester_agent* identifies the transaction program that originated the unit-of-work request.

Presence Rule: When the *unit_of_work_requester_agent* is absent, the *origin_agent* specified in the SNA/DS distribution is the default.

Format: Character string, except for first byte

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

The first byte of an SNA-registered transaction program name ranges in value from X'00 to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA-registered. X'40' is not a valid first-byte value.

U_of_W_Seqno_DTM

Description: The sequence number is the number assigned to the unit-of-work request by the SNA/FS originating agent. The value ranges from 1 to (2^{31})-1. The date of the unit-of-work request is assigned by the *u_of_w_requester_agent*; the time of the unit-of-work request is assigned by the *u_of_w_requester_DSU*. The offset from GMT for local time is included.

Format: Byte string (See below)

Byte	Contents
0-1	LT header
2-5	Sequence number Signed binary integer limited to $(2^{31})-1$.
	DATE
6-7	Year, in binary (e.g., year 1989 is encoded as X'07C5')
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 23 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	GMT FLAG
14	Indicates that specified TIME is GMT and identifies whether offsets from GMT are required to calculate local time. (Equivalent EBCDIC characters are shown in parentheses.) X'E9' (Z) no offset required X'4E' (+) add required offset to GMT to get local time X'60' (−) subtract required offset from GMT to get local time
	OFFSET
15	Hour offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 23 are valid)
16	Minute offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 59 are valid)

Examples

A 9-byte date/time encoding is a date/time followed immediately by an EBCDIC "Z" and is considered to be GMT. Thus, 12:00 GMT on 2 January 1988 would be

```
X'07C401020C000000E9'
  yyyymmddhhmmsshhZ
```

An 11-byte date/time encoding is a date/time followed immediately by an EBCDIC "+" or "-" and two one-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00 a.m. on 2 January 1988 in New York would be 12:00 GMT - 5 hours, or

```
X'07C401020C000000600500'
  yyyymmddhhmmsshh- hhmm
```

Unrecognized_Reserve

Description: The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure.

SNA/FS servers pass *unrecognized_reserve* structures through unchanged in the outgoing *server_object*.

Format: Undefined byte string

FS_Agent_Request

Description: The *FS_agent_request* contains the control information that describes the SNA/FS agent action to be performed.

Command

Description: The *command* specifies the type of SNA/FS request or SNA/FS reporting action.

Format: Byte string

Byte	Content
0-3	LLID header
4	X'10' REPORT_FS_ACTION
	X'11' REPORTING_FS_ACTION
	X'12' TRANSFER_TO_REQUESTER

Note: REPORTING_FS_ACTION is valid only in reporting flows, while the other values are valid only in requesting flows.

Command_Parms

Description: The *command_parameters* contain and qualify the control information for the *command*.

Source_Reporting_Action

Description:	The *source_reporting_action* describes the type of reporting the source agent performs.
Presence Rule:	Occurs when the requesting agent requires reports from the source, and the *command* is TRANSFER_TO_REQUESTER; otherwise, precluded.
Format:	Byte string
Note:	The reporting action requested of the agent cannot be more demanding than that requested of the server.

Byte	Contents	
0-1	LT header	
2	X'01'	DETAILED
	X'10'	SUMMARY_OR_EXCEPTIONS
	X'11'	ONLY_IF_EXCEPTIONS

Target_Agent

Description:	The *target_agent* is the transaction program at the target location.
Presence Rule:	Occurs when the target_agent is different from the source agent, and the *command* is TRANSFER_TO_REQUESTER; otherwise, precluded. When the *target_agent* is absent, the *dest_agent* specified in the SNA/DS distribution is the default.
Format:	Character string, except for the first byte
CGCSGID:	01134-00500
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00 to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA-registered. X'40' is not a valid first-byte value.

Target_Reporting_Action

Description:	The *target_reporting_action* describes the type of reporting the target agent performs.
Presence Rule:	Occurs when the requester requires target reporting, and the *command* is REPORT_FS_ACTION or TRANSFER_TO_REQUESTER; otherwise, precluded.
Format:	Byte string

Byte	Contents	
0-1	LT header	
2	X'01'	DETAILED
	X'10'	SUMMARY_OR_EXCEPTIONS
	X'11'	ONLY_IF_EXCEPTIONS

Report-To_Agent

Description:	The *report-to_agent* is the name of the transaction program to which reports are to be delivered after the SNA/FS activity has concluded.
Presence Rule:	Occurs when the requester requires reporting to a third-party agent that is different from the requesting agent, and the *command* is REPORT_FS_ACTION or TRANSFER_TO_REQUESTER; otherwise, precluded. When the *report-to_agent* is absent and reporting is required, the *dest_agent* specified in the SNA/DS distribution is the default.
Format:	Character string, except for the first byte
CGCSGID:	01134-00500
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00 to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA-registered. X'40' is not a valid first-byte value.

Report-To_DSU

Description:	The *report-to_DSU* is the name of the DSU to which the SNA/FS reports are to be delivered.
Presence Rule:	Occurs when the requester requires reporting and requests the reports be delivered to a DSU other than the default DSU. When the *report-to_DSU* is absent, the *report-to_DSU* specified in the SNA/DS distribution is the default. If the *report-to_DSU* is also absent, the *origin_DSU* is the default. Typically the SNA/DS distributions between the source and target locations normally carry the requesting DSU as the SNA/DS *report-to_DSU*.

Report-To_RGN

Description: The *report-to_RGN* is the first part of the DSU name to which the SNA/FS reports are to be delivered. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

Report-To_REN

Description: The *report-to_REN* is the second part of the DSU name to which the SNA/FS reports are to be delivered. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

Note: In existing networks where LU names are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the REN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new LU names.

Report-To_User

Description: The *report-to_user* is the name of the user to which the SNA/FS reports are to be delivered.

Report-To_DGN

Description: The *report-to_DGN* is the first part of the user name to which the SNA/FS reports are to be delivered.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

Report-To_DEN

Description: The *report-to_DEN* is the second part of the user name to which the SNA/FS reports are to be delivered.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X'40') characters are not allowed.

FS_Server_Request

Description: The *FS_server_request* describes the action to be performed by the server, and may also contain object identifiers and object contents.

FS_Server_Request_Prefix

Description: The *FS_server_request_prefix* identifies the beginning of the *FS_server_request.*

Decoder_Instruction

Description: The *decoder_instruction* describes the server action to be performed by the decoder-role server at either the source location or report-to location.

Presence Rules: Occurs when:

- The TRANSFER_TO_REQUESTER agent command and its accompanying server request flow from the requesting location to the source location.
- The REPORTING_FS_ACTION agent command and its accompanying server report flow from the target location to the report-to location.

Format: Bit string

Note: The values for the *decoder_instruction* are described on page 17-24.

Source_Instruction

Description: The *source_instruction* describes the action to be performed by the source-role server at the source location.

Presence Rule: Occurs when the TRANSFER_TO_REQUESTER agent command and its accompanying server request flow from the requesting location to the source location.

Format: Bit string

Note: The values for the *source_instruction* are described on page 17-24.

Target_Instruction

Description:	The *target_instruction* describes the server action to be performed by the target-role server at the target location.
Presence Rules:	Occurs when: • The TRANSFER_TO_REQUESTER agent command and its accompanying server request flow from the requesting location to the source location. • A server request containing a data object flows from the source location to the target location. • A server request for a deletion flows from the requesting location to the target location.
Format:	Bit string
Note:	The values for the *target_instruction* are described on page 17-24.

Server Instructions, Decoder, Source, and Target

Byte	Bit	Contents	Server Role
0-1		LT header	
2	0-3	Server instruction:	
		0001 FETCH	source
		0010 DECODE	decoder
		0011 CREATE_LOAD_OR_REPLACE	target
		0100 DELETE	target
		0101 REPLACE	target
		0110 CREATE_LOAD	target
	4-7	Exception action:	
		0001 ABEND	decoder, source, or target
		0010 BACKOUT	target
3	0-3	Reporting action:	
		0001 DETAILED	decoder, source, or target
		0010 SUMMARY_OR_EXCEPTIONS	decoder, source, or target
		0011 ONLY_IF_EXCEPTIONS	decoder, source, or target
	4-7	Reserved	
4	0-3	Intention (see Note 2):	
		0001 EXECUTING	target
		0011 STORING	target
		0100 NOT APPLICABLE	decoder, source, or target
	4-7	Reserved	

Note: Byte 4 is optional and may be omitted.

Data_Object_Group

Description: The *data_object_group* defines the overall characteristics about the data object.

Presence Rules: Required in:

- The *FS_server_request.*
- The *FS_server_report* whenever the *SNA_condition_report* is absent; otherwise, optional.

Group_Prefix

Description: The *data_object_group_prefix* identifies the beginning of the *data_object_group.*

Supplemental_FS_Info1-Supplemental_FS_Info6

Description: The *supplemental_FS_info1 - supplemental_FS_info6* structures are reserved for future use.

Data_Object

Description: The *data_object* is the basic entity managed by SNA/FS.

D_O_Prefix

Description: The *data_object_prefix* identifies the beginning of the *data_object.*

D_O_Attributes

Description: The *data_object_attributes* contain information about the contents of the data object that the SNA/FS server uses to determine whether the server can honor the request.

D_O_Class

Description: The *data_object_class* identifies the class of the data object by means of a hierarchical structure of codes. The classification and intention information are used by the target server to determine whether or not the request can be honored.

Format: Byte string

Notes: 1. Refer to "SNA/FS Data Object Classification Codes" on page 17-35 for the value descriptions.

2. When unknown by requester, all 0 bits are used. Source will supply.

D_O_Global_Name

Description: The *data_object_global_name* is the unique, system-independent identifier for the data object. The name is assigned according to naming conventions established by the using architecture. The canonical identifier consists of a string of tokens, where the leftmost tokens are more significant. A higher-order token identifies the naming authority that issues or manages the values of the lower-order tokens.

D_O_Description

Description: The *data_object_description* provides brief descriptive information about the object.

D_O_Transforms

Description: The *data_object_transform* defines the object transform that has been applied to the contents of the data object.

D_O_Allocation_Info

Description:	The *data_object_allocation_info* provides the target location with space requirements needed to store the data object.
Presence Rule:	Occurs when *data_object_contents* is present.

D_O_Contents

Description:	The *data_object_contents* is the byte contents of the *data_object*.
Presence Rule:	Precluded when the *decoder_instruction* is present or the *target_instruction* is DELETE.
Format:	Undefined byte string

D_O_Suffix

Description:	The *data_object_suffix* contains no information and marks the end of the *data_object*.

G_Suffix

Description:	The *data_object_group_suffix* contains no information and marks the end of the *data_object_group*.

FS_Suffix

Description:	The *FS_suffix* contains no information and marks the end of the *FS_request* or the *FS_report*.

FS_Agent_Report

Description:	The *FS_agent_report* provides a summary on the relative success of a previous SNA/FS request.

FS_Action_Summary

Description:	The *FS_action_summary* indicates whether the actions requested of the server were successfully performed.
Format	Bit string
Note:	The values for the *FS_action_summary* bit string are described on page 17-26.

Byte	Bit	Contents
0-3		LLID header
4	0-1	01 ALL_SUCCESSFUL (see Note) 11 NONE_SUCCESSFUL
	2-3	00 NO_BACKOUT_ATTEMPTED 01 ALL_BACKED_OUT
	4-5	00 ABEND_NOT_APPLICABLE 01 SERVER_ABEND
	6-7	Reserved

Note: If this value (ALL SUCCESSFUL) is present, all subsequent bits are 0.

FS_Server_Report

Description:	The *FS_server_report* provides information on the relative success of one or more server operations.

FS_Server_Report_Prefix

Description:	The *FS_server_report_prefix* identifies the beginning of the *FS_server_report.*

Token_Attributes

Description:	The *token_attributes* define for each token in the global name how that token can be used in partial matching or token value generation. These attributes are stored in the SNA/FS catalog.
Presence Rule:	Occurs when the server instruction is a create operation (e.g., CREATE_LOAD; CREATE_LOAD_OR_REPLACE).
Format:	Bit string (See below)

Byte	Contents
0-1	LT header
2-11	Up to 10 different token attributes can be specified.

Token Attribute Values

For each token in the token string, there will be a single byte of attribute information, as follows:

Bit	Contents	
0	0	MUST_MATCH
	1	NEED_NOT_MATCH
1	0	NOT_GENERABLE
	1	GENERABLE
2	Reserved	
3-7	00000	UNSPECIFIED TYPE, ≤16 CHARACTERS
	00001	NETID
	00010	LU-NAME
	00011	SYSTEM_TYPE
	00100	SUBTREE_INDICATOR (STI)
	10000	ORDERED, ≤16 CHARACTERS
	10001	ORDERED, ≤16 DECIMAL NUMERICS
	10010	ORDERED, DATE - Y1991M12D31
	10011	ORDERED, TIME - H23M59S59
	10100	ORDERED, G00V00

Notes:

1. The target SNA/FS server is obligated to preserve the attribute characteristic in the catalog at the target node and to honor subsequent deletion requests based on this characteristic. If all bits in the catalog entry attribute byte are 0, i.e., MUST_MATCH, the corresponding identifier must be exactly matched for deleting and replacing operations.
2. If all of the flag bits in all of the token attribute bytes are set at their default values, e.g., 0, the token attributes may be omitted and the target server assumes the default.

To_Be_Fetched_Name

Description:	The *to_be_fetched_name* is the name of the object, at the source location, that is to be fetched by the SNA/FS server.
Presence Rule:	Occurs in: • The *FS_server_request* when an object is to be fetched from the source location. The source server instruction must be FETCH. • The *FS_server_report* when the FETCH server operation was unsuccessful or not attempted, and reporting was requested.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Fetching_Match_Flags

Description:	The *fetching_match_flags* govern the partial matching operation at fetch time.
Presence Rule:	Occur when partial matching is required at fetch time.
Format:	Byte string

Byte	Contents
0-1	LT header
2-11	For each token in the token string, up to a maximum of 10 tokens, a single byte describes that token's use in a fetch operation.

Fetching Match Flag Values

Values	
X'00'	FIND_A_MATCH
X'01'	IGNORE
X'02'	SELECT_HIGHEST
X'03'	SELECT_LOWEST

To_Be_Stored_Name

Description:	The *to_be_stored_name* is the name of the object that is to be stored at the target location. Typically, the source-role server will obtain the name at fetch time.
Presence Rule:	Occurs in: • The *FS_server_request* flow between the source and target locations when an object is to be stored at the target location. The requester can also specify parts of a *to_be_stored_name*; therefore, in this case, the structure is present between the requesting and source locations. • The *FS_server_report* when the storing operation was unsuccessful or not attempted, and reporting was requested.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

To_Be_Deleted_Name

Description:	The *to_be_deleted_name* is the name of the object, at the target location, that is to be deleted by the SNA/FS server.
Presence Rule:	Occurs in: • The *FS_server_request* when an object is to be deleted from the target location. • The *FS_server_report* when the delete operation was unsuccessful or not attempted, and reporting was requested.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.
Note	For a replace operation, the *to_be_deleted* name needs to contain only the NEED_NOT_MATCH tokens that differ from the values in the identifier of the *to_be_stored* data objects.

Deleting_Match_Flags

Description:	The *deleting_match_flags* govern the matching operation, at the target location, of the object to be deleted.
Presence Rule:	Occurs when partial matching is required to identify the *to_be_deleted* object.
Format:	Byte string

Byte	Contents
0-1	LT header
2-11	For each token in the token string, up to a maximum of 10 tokens, a single byte describes that token's use in a delete operation.

Deleting Match Flag Values

Values	
X'00'	FIND_A_MATCH
X'01'	IGNORE
X'02'	SELECT_HIGHEST
X'03'	SELECT_LOWEST

Fetched_Name

Description:	The *fetched_name* is the name of the object fetched by the SNA/FS server.
Presence Rule:	Occurs only in the *FS_server_report* when the source agent reports that an object has been fetched.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Stored_Name

Description:	The *stored_name* is the name of the object stored by the SNA/FS server.
Presence Rules:	Occurs: • In the *FS_server_report* when the target agent reports that an object has been stored. • When the request is being used to convey a data object name.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Deleted_Name

Description:	The *deleted_name* is the name of the object deleted by the SNA/FS server.
Presence Rule:	Occurs only in the *FS_server_report* when the target agent reports that an object has been deleted.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Reported-On_Name

Description:	The *reported-on_name* is the name of the object being reported by the SNA/FS server. The *reported-on_name* is used in cases when the state of the object being reported on cannot be determined.
Presence Rule:	Occurs only in the *FS_server_report*.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Object_Text_Description

Description:	The *object_text_description* identifies the descriptive text and how the text is to be interpreted.

CCSID

Description: The *coded_character_set_id* identifies the codepage and character set in which the text message is encoded. The structure of the CCSID is documented in the *&CCSID.*.

Format: Bit string

Language_ID

Description: The *language_id* identifies the coded national language in which the text message is written. The language IDs are defined in Volume 2 of the *&LANGID.*.

Format: Character string

Descriptive_Text

Description: The *descriptive_text* contains a brief description about the data object.

Format: Character string

Uniform_Text_Content

Description: The *uniform_text_content* identifies for any text data object its associated codepage and character set.

Compression_Transform

Description: The *compression_transform* indicates that the data object has been compressed.

Presence Rule: Required when the *data_object_contents* was stored compressed at the source server.

Compressed_State

Description: The *compressed_state* indicates if the *data_object_contents* is compressed.

Format: Byte string

Byte	Contents
0-1	LT header
2	X'01' OBJECT_IS_COMPRESSED X'02' OBJECT_IS_NOT_COMPRESSED

FS_Compression

Description:	The *FS_Compression* identifies that the *data_object_contents* has been compressed using the compression algorithm defined by the SNA/FS server.
Presence Rule:	Precluded when *user_compression* is present.

Compression_Technique

Description:	The *compression_technique* identifies the SNA/FS-defined compression algorithm.
Format:	Byte string

Byte	Contents
0-1	LT header
2	X'01' SCB_COMPRESSION (STRING CONTROL BYTE)

Prime_Character

Description:	The *prime_character* identifies the character to be used to replace repetitive sequences of that character.
Presence Rule:	Required when the *prime_character* is not the default value.
Format:	Byte string, single byte; the default is the space character (X'40').

User_Compression

Description:	The *user_compression* identifies that the *data_object_contents* has been compressed using a user-defined compression algorithm.
Presence Rule:	Precluded when *FS_compression* is present.

Algorithm_Name

Description:	The *algorithm_name* identifies the user-defined compression algorithm.
Format:	Character string

Algorithm_Parms

Description:	The *algorithm_parms* identifies the parameters needed for the user-defined algorithm.
Format:	Character string

Transfer_Size

Description: The *transfer_size* is an estimate of the number of bytes in the *data_contents*. It can be larger or smaller than the actual size; however, it should be accurate enough for the target location to use for space decisions.

Format: Unsigned binary integer (1-origin)

Record_Information

Description: The *record_information* describes the record layout of the data object.

Record_Format

Description: This specifies the *record_format* of the data object.

Format Byte string

Byte	Contents
0-1	LT header
2	X'01' FIXED X'02' VARIABLE

Max_Record_Size

Description: This is the *maximum_record_size* of any record that can occur in the data object.

Format Unsigned binary integer (1-origin)

SNA/FS Data Object Classification Codes

SNA/FS Data Object Classes				Hex Codes			
Level 1	Level 2	Level 3	Level 4	1	2	3	4
Executable	System Microcode [1]	Unspecified	Unspecified	10	10	00	00
		Patch	Unspecified	10	10	10	00
			Product Specific	10	10	10	Ex
		Fix	Unspecified	10	10	20	00
			Product Specific	10	10	20	Ex
		Suffix_EC	Unspecified	10	10	30	00
			Product Specific	10	10	30	Ex
		Maint_EC	Unspecified	10	10	40	00
			Product Specific	10	10	40	Ex
		Funct_EC	Unspecified	10	10	50	00
			Product Specific	10	10	50	Ex
		Feature	Unspecified	10	10	60	00
			NLS_EC	10	10	60	51
			Product Specific	10	10	60	Ex
			I/O_EC	10	10	60	E0
			Customer Specific	10	10	60	Fx
	Microcode Customization	Unspecified	Unspecified	10	20	00	00
		Product Specific	Unspecified	10	20	Ex	00
		History_Log	Unspecified	10	20	E0	00
		Activate_Log	Unspecified	10	20	E1	00
		Canonical_Directory	Unspecified	10	20	E2	00
		MCF_Directory	Unspecified	10	20	E3	00
	Software	Unspecified	Unspecified	10	30	00	00
		Product Specific	Unspecified	10	30	Ex	00
	Application Procedure	Unspecified	Unspecified	10	50	00	00
		CLIST	Unspecified	10	50	20	00
		EXEC	Unspecified	10	50	30	00
		SAA REXX	Unspecified	10	50	50	00
		Product Specific	Unspecified	10	50	Ex	00

SNA/FS Data Object Classes				Hex Codes			
Level 1	Level 2	Level 3	Level 4	1	2	3	4
Processable	Flat	Unspecified	Unspecified	20	10	00	00
	Relational	Unspecified	Unspecified	20	20	00	00
	Keyed	Unspecified	Unspecified	20	25	00	00
Maintenance	Dump	Unspecified	Unspecified	40	10	00	00
	Configuration File	Unspecified	Unspecified	40	20	00	00
	Trace Information	Unspecified	Unspecified	40	30	00	00
	Error Log	Unspecified	Unspecified	40	40	00	00

Code Points Used by SNA/FS

The values of the ID component of the LLID structures as used for SNA/FS GDS variables are shown below:

ID	Structure Name
1530	FS Agent Request
1531	FS Server Request Prefix
1532	SNA Condition Report
1533	Data Object Group Prefix
1534	Supplemental FS Information1
1535	Supplemental FS Information2
1536	Data Object Prefix
1537	Data Object Attributes
1538	Data Object Global Name
1539	Supplemental FS Information5
153B	Data Object Description
153C	Supplemental FS Information3
153E	Data Object Transforms
153F	Data Object Allocation Information

[1] Microcode may be classified as IBM Licensed Internal Code. See "Notices" near the beginning of this document for more information.

1541	Data Object Contents
1542	Data Object Suffix
1543	Data Object Group Suffix
1546	Command
1547	Command Parms
1548	FS Action Summary
1549	Agent Unit of Work Correlator
154A	FS Agent Report
154B	FS Server Report Prefix
154C	FS Suffix
1550	Supplemental FS Information4

Transaction Program and Server Names

The following is a list of the SNA/FS-defined server name, the SNA/FS-defined transaction program name, and the names of other SNA/FS-capable transaction programs.

Code	Meaning
X'24F0F0F0'	SNA/FS server name
X'23F0F0F0'	SNA/MS change management agent TP name

Global Name Registration

The following is a list of the identifier tokens that have been registered in SNA/FS on behalf of SNA/FS-capable agents.

First Identifier	Agent
C'MCODE'	SNA/MS change management
C'MCUST'	SNA/MS change management
Registered Enterprise ID	SNA/MS change management

The following is a list of the subtree indicator tokens that have been registered in SNA/FS on behalf of SNA/FS-capable agents.

Subtree Indicator	Agent
C'GRP' (Group)	SNA/MS change management
C'REF' (Refresh)	SNA/MS change management
C'UPD' (Update)	SNA/MS change management
C'FIX' (Fix)	SNA/MS change management

Appendix A. SNA Character Sets and Symbol-String Types

This appendix describes the character sets and symbol-string types used, for example, for the following fields:

- LU name
- Network-qualified LU name
- Mode name
- COS name
- Transaction program name
- Access security information subfields
- Program initialization parameters (PIP) subfields
- Map name
- SNADS server, user (DGN, DEN), and service unit (RGN, REN) names

The detailed syntax of these strings is described in other chapters where their usage within individual message units is defined.

Symbol-String Type

The symbol-string type specifies the set of code points and corresponding characters from which the strings listed above are composed, as follows:

- Type A (Assembler oriented): a character string consisting of one or more characters from character set A. The first character of a type-A symbol string is not a numeric; i.e., it is different from X'F0', X'F1', ..., or X'F9'.
- Type 1134 (Type A subset): a character string consisting of one or more EBCDIC uppercase letters A through Z and numerics 0 through 9. For certain names, IBM implementation usage constrains the leading character to be alphabetic; these names include the following:
  - network ID
  - network name (e.g., LU name, link name)
  - mode name
  - class-of-service name (COS name)
- Type AE (A extended): a character string consisting of one or more characters from character set AE, with no restriction on the first character.
- Type 930 (distribution services oriented): a character string consisting of one or more characters from character set 930, with the following rules:
  - No leading space (X'40') characters are used, but no other restrictions exist on the first character.
  - Imbedded space (X'40') characters are significant.
  - Trailing space (X'40') characters are not significant.

- Type USS (unformatted system services oriented, used for character-coded requests): a character string consisting of one or more characters from character set USS, with no restriction on the first character.
- Type GR (EBCDIC graphics): a byte string consisting of one or more bytes within the range X'41' through X'FE', with no restriction on the first byte.
- Type G (general): a byte string consisting of one or more bytes within the range X'00' through X'FF', with no restriction on the first byte.
- Type DB (double byte): a byte-string consisting of an even number of four or more bytes beginning with a byte set to X'0E', followed by bytes having values in the range X'41' through X'FE', and ending with a byte set to X'0F'.

SNA Character Sets and Encodings

A character set is a set of graphic characters, such as letters, numbers, and special symbols. SNA formats make use of a variety of character sets. Character sets A, AE, 930, USS, 1134, and 640 define the characters that are allowed in the corresponding symbol-strings.

Each character set is encoded using a code page. A code page is the specification of code points, or hexadecimal values, for one or more character sets. All character sets used by SNA are encoded using IBM code page 00500, the relative encodings of which are shown in Figure A-1.

Character sets encoded using a specific code page are officially denoted by the concatenation of their character set and code page numbers, such as 00640-00500 and 01134-00500. The concatenation of these two numbers specifies a *coded graphic character set*. The older character sets—A, AE, 930, and USS—and their encodings continue to be supported but not for new formats, which now use 00640-00500 and 01134-00500.

Figure A-1 on page A-3 defines the character sets and encodings for A, AE, 930, USS, 01134-00500, and 00640-00500. The code points that do not belong to any of these sets are not shown.

Figure A-1 (Page 1 of 3). Character Sets A, AE, 930, USS, 1134, and 640

Hex Code	Graphic	Description	Set					
			A	AE	930	USS	1134	640
15		Line Feed				X		
40		Space			X	X		X
4B	.	Period		X	X	X		X
4C	<	Less Than Sign						X
4D	(	Left Parenthesis				X		X
4E	+	Plus Sign				X		X
50	&	Ampersand			X	X		X
59	ß	Sharp s			X			
5B	$	Dollar Sign	X	X	X	X		
5C	*	Asterisk				X		X
5D	)	Right Parenthesis				X		X
5E	;	Semicolon						X
60	-	Minus Sign			X	X		X
61	/	Slash			X	X		X
62	Â	A Circumflex, Capital			X			
63	Ä	A Diaeresis, Capital			X			
64	À	A Grave, Capital			X			
65	Á	A Acute, Capital			X			
66	Ã	A Tilde, Capital			X			
67	Å	A Overcircle, Capital			X			
68	Ç	C Cedilla, Capital			X			
69	Ñ	N Tilde, Capital			X			
6B	,	Comma			X	X		X
6C	%	Percent Sign						X
6D	_	Underline						X
6E	>	Greater Than Sign						X
6F	?	Question Mark						X
71	É	E Acute, Capital			X			
72	Ê	E Circumflex, Capital			X			
73	Ë	E Diaeresis, Capital			X			
74	È	E Grave, Capital			X			
75	Í	I Acute, Capital			X			
76	Î	I Circumflex, Capital			X			
77	Ï	I Diaeresis, Capital			X			
78	Ì	I Grave, Capital			X			
7A	:	Colon						X
7B	#	Number Sign	X	X	X	X		
7C	@	At Sign	X	X	X	X		
7D	'	Apostrophe			X	X		X
7E	=	Equal Sign				X		X
7F	"	Quotation Marks						X

Figure A-1 (Page 2 of 3). Character Sets A, AE, 930, USS, 1134, and 640

Hex Code	Graphic	Description	Set					
			A	AE	930	USS	1134	640
80	Ø	O Slash, Capital			X			
81	a	a, Small		X				X
82	b	b, Small		X				X
83	c	c, Small		X				X
84	d	d, Small		X				X
85	e	e, Small		X				X
86	f	f, Small		X				X
87	g	g, Small		X				X
88	h	h, Small		X				X
89	i	i, Small		X				X
91	j	j, Small		X				X
92	k	k, Small		X				X
93	l	l, Small		X				X
94	m	m, Small		X				X
95	n	n, Small		X				X
96	o	o, Small		X				X
97	p	p, Small		X				X
98	q	q, Small		X				X
99	r	r, Small		X				X
9A	a̲	a Underscore, Small			X			
9B	o̲	o Underscore, Small			X			
9E	Æ	AE Dipthong, Capital			X			
A0	μ	Micro, Mu			X			
A2	s	s, Small		X				X
A3	t	t, Small		X				X
A4	u	u, Small		X				X
A5	v	v, Small		X				X
A6	w	w, Small		X				X
A7	x	x, Small		X				X
A8	y	y, Small		X				X
A9	z	z, Small		X				X
AC	Ð	D Stroke, Capital			X			
AD	Ý	Y Acute, Capital			X			
AE	Þ	Thorn, Capital			X			
C1	A	A, Capital	X	X	X	X	X	X
C2	B	B, Capital	X	X	X	X	X	X
C3	C	C, Capital	X	X	X	X	X	X
C4	D	D, Capital	X	X	X	X	X	X
C5	E	E, Capital	X	X	X	X	X	X
C6	F	F, Capital	X	X	X	X	X	X
C7	G	G, Capital	X	X	X	X	X	X

Figure A-1 (Page 3 of 3). Character Sets A, AE, 930, USS, 1134, and 640

Hex Code	Graphic	Description	Set					
			A	AE	930	USS	1134	640
C8	H	H, Capital	X	X	X	X	X	X
C9	I	I, Capital	X	X	X	X	X	X
D1	J	J, Capital	X	X	X	X	X	X
D2	K	K, Capital	X	X	X	X	X	X
D3	L	L, Capital	X	X	X	X	X	X
D4	M	M, Capital	X	X	X	X	X	X
D5	N	N, Capital	X	X	X	X	X	X
D6	O	O, Capital	X	X	X	X	X	X
D7	P	P, Capital	X	X	X	X	X	X
D8	Q	Q, Capital	X	X	X	X	X	X
D9	R	R, Capital	X	X	X	X	X	X
DF	ÿ	y Diaeresis, Small			X			
E2	S	S, Capital	X	X	X	X	X	X
E3	T	T, Capital	X	X	X	X	X	X
E4	U	U, Capital	X	X	X	X	X	X
E5	V	V, Capital	X	X	X	X	X	X
E6	W	W, Capital	X	X	X	X	X	X
E7	X	X, Capital	X	X	X	X	X	X
E8	Y	Y, Capital	X	X	X	X	X	X
E9	Z	Z, Capital	X	X	X	X	X	X
EB	Ô	O Circumflex, Capital			X			
EC	Õ	O Diaeresis, Capital			X			
ED	Ò	O Grave, Capital			X			
EE	Ó	O Acute, Capital			X			
EF	Õ	O Tilde, Capital			X			
F0	0	Zero	X	X	X	X	X	X
F1	1	One	X	X	X	X	X	X
F2	2	Two	X	X	X	X	X	X
F3	3	Three	X	X	X	X	X	X
F4	4	Four	X	X	X	X	X	X
F5	5	Five	X	X	X	X	X	X
F6	6	Six	X	X	X	X	X	X
F7	7	Seven	X	X	X	X	X	X
F8	8	Eight	X	X	X	X	X	X
F9	9	Nine	X	X	X	X	X	X
FB	Û	U Circumflex, Capital			X			
FC	Ü	U Diaeresis, Capital			X			
FD	Ù	U Grave, Capital			X			
FE	Ú	U Acute, Capital			X			

Appendix B. GDS ID Description and Assignments

This appendix defines the *general data stream* (GDS), which is used in a variety of ways in SNA. For instance, it is used to encode the Document Interchange Architecture (DIA) message units. The basic structural unit in GDS is the structured field, a string of bytes preceded by a length and beginning with a GDS identifier (ID) that defines the structure of the remainder of the field. Some structured fields are used by components of SNA; these fields are defined in Chapter 13, "GDS Variables for SNA Service Transaction Programs (STPs)," Chapter 16, "GDS Variables for Application Programs," Chapter 5, "Request/Response Units (RUs)," and Chapter 11, "Function Management (FM) Headers." GDS IDs are assigned, generally in blocks of consecutive values, to different layers and components of SNA and to other interconnection architectures. For a complete listing of these block assignments, see below.

The general data stream applies to data exchanged between nodes over links and to data exchanged via removable storage media or shared storage facilities.

Structured Fields

Each structured field has the format shown in Figure B-1.

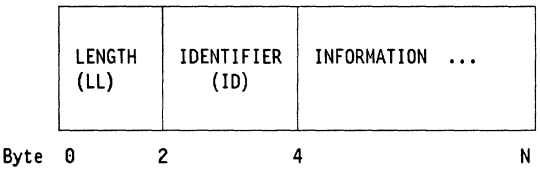


Figure B-1. GDS Structured Field

Length (LL) Description

The LLID is a 4-byte field in which the two LL bytes are used to indicate the length of the LLID field itself (4 bytes) plus the data following the LLID; up to 32,763 bytes of data may follow the LLID. Values 0 and 1 of the LL are reserved for use as escape sequences; values 2 and 3 are not used. For example, a value of X'0001' indicates a presentation services header, which is used for sync point management.

Bit 0 (high-order bit) of byte 0 is used as a length continuation (or not-last segment) indicator. If that bit is set to 1, the logical record is continued by another 2-byte LL; the ID occurs only following the first LL. The continuing LL is located immediately following the information bytes encompassed by the first LL. The continuing LL might itself be continued. In other words, the length specified by the continuing LL might not be the entire remainder of the logical record; it might be followed by yet another LL. The amount of data spanned by each continuing LL can be any size convenient to the sender (including 0). Eventually, the chain of continuing LLs is ended by a final LL, i.e., one with the high-order (not-last) bit set to 0. The final LL may indicate a null information field follows (length = 2).

When an LLID encompasses a string of logical records identified by full LLIDs, the length of the string, determined by summing the (nested) encompassed LLs, equals the length definer of the (outer) encom-

passing LLID less 4 (this applies at each level of nesting). If the encompassing LLID is continued by segmenting, the length of the string of segments equals the sum of the initial LL and all continuing LLs of the encompassing ID less 4 for the initial LLID and 2 for each continuing LL.

The 2-byte ID values, irrespective of the level of nesting at which they occur, are defined uniquely across all levels of nesting, with the following exception. The ID values X'FF00' through X'FFFF' are used only within an encompassing LLID (which is not necessarily the immediate parent structure); their meaning is defined by the architecture that owns the higher-level ID and it applies only within the context of that ID. In other words, ID values in the X'FF**' range are context dependent. All other ID values are context independent.

Identifier (ID) Description

The 2-byte identifier that follows the length field indicates the format and meaning of the data that follows. Sometimes additional values appearing in the information field are needed to completely specify the information field's content. The uniqueness of the identifier (with the exceptions noted above) makes it easy to decode structured fields in line traces, and also to make it easier to create composite data streams by including elements of several architectures. DIA carried by SNADS is an example of such a use.

Identifier Registry

The identifiers that have been assigned for specific use are listed below. Identifiers are assigned in blocks; not all identifiers in a block are necessarily currently used by the owner. As usual, the asterisk (*) indicates "any value."

Figure B-2 (Page 1 of 4). Identifier Registry

GDS ID	Structured Field Owner
0000 – 01FF	3270
03**	3270
06**	3270
09**	3270
0B00 – 0EFF	3270
0F**	3270
101*	3270
1030 – 1034	Print Job Restart
1100 – 1104	SNA Character String
12**	LU 6.2 and APPN

Figure B-2 (Page 2 of 4). Identifier Registry

GDS ID	Structured Field Owner
13**	SNA/Management Services
140*	3820 Page Printer
1530 – 1531	SNA File Services
1532	SNA Condition Report
1533 – 154F	SNA File Services
1550 – 155F	SNA File Services
1570 – 158F	SNA/Distribution Services
4000 – 41FF	3270
4A00 – 4CFF	3270
71**	3250
8000 – 81FF	3270
C00*	Document Interchange Architecture
C100 – C104	Document Interchange Architecture
C105	SNA/Distribution Services
C10A – C122	Document Interchange Architecture
C123 – C124	SNA/Distribution Services
C219	Document Interchange Architecture
C300 – C345	Document Interchange Architecture
C350 – C361	SNA/Distribution Services
C366 – C46F	Document Interchange Architecture
C500 – C56F	Document Interchange Architecture
C600 – C66F	Document Interchange Architecture
C7**	Graphical Display Data Manager
C800 – C87F	Document Interchange Architecture

Figure B-2 (Page 3 of 4). Identifier Registry

GDS ID	Structured Field Owner
C900 – CB0F	Document Interchange Architecture
CC00 – CC3F	Document Interchange Architecture
CD00 – CD3F	Document Interchange Architecture
CF0*	Document Interchange Architecture
D0**	Distributed Data Management
D3**	Document Content Architecture
D6**	Intelligent Printer Data Stream
D780 – D7BF	Facsimile Architecture
D820 – D821	AS/400 (5250)
D822 – D826	AS/400 (5394)
D930 – D95F	AS/400 (5250)
E100 – E10F	Level-3 Document Content Architecture
E20*	Level-3 Document Content Architecture
E30*	Level-3 Document Content Architecture
E40*	Level-3 Document Content Architecture
E50*	Level-3 Document Content Architecture
E60*	Level-3 Document Content Architecture
E70*	Level-3 Document Content Architecture
E80*	Level-3 Document Content Architecture
E90*	Level-3 Document Content Architecture
EA0*	Level-3 Document Content Architecture
EFFF	IBM Token-Ring Network PC Adapter
F000 – FEFF	Non-IBM Reserved Block

Figure B-2 (Page 4 of 4). Identifier Registry

GDS ID	Structured Field Owner
FF**	Context-Dependent Block

Appendix C. Common Structures

This appendix contains the information about the SNA/DS, SNA/FS, and SNA/MS-CM SNA Condition Report (SNACR). For more information about on the SNACR refer to the following books:

- *SNA/Distribution Services Reference*
- *SNA/File Services Reference*
- *SNA/Management Services Reference*

The format descriptions are comprised of two parts, header description tables and structure descriptions. A header description table contains the header information for each structure. A structure description contains a prose description of the structure, bit-level representations, and any presence rules or length restrictions associated with a particular structure.

Encoding Rules and Representations

The SNA Condition Report (SNACR) format is described in terms of encoded fields referred to as "structures" and the hierarchical relationship between these structures. In this document, the header description tables show each structure and its header. Elsewhere in this book, the header length is assumed not to be part of the overall structure length (e.g., *SNA_report_code*).

Structure Classifications

Fields and groupings of fields are known as structures. They are categorized in terms of their hierarchical position ("atomic," "child," or "parent"), the method by which their beginning and endings are determined, (length-bounded, delimited, or implied) and which kind of header is used to identify them (LT or LLID). Only certain combinations of characteristics are possible.

Length-bounded Structures

Length-bounded structures consist of a header and usually some following information. A header may be either two bytes in length, referred to as an "LT" (length and type), or four bytes in length, referred to as an "LLID" (length and GDS code point). In either case, the length byte(s) include the length of the header itself and the following information, if any.

Atomic Structures

In many cases, a structure consists only of its own header followed by data. These structures cannot be decomposed, and therefore they are called "atomic." Atomic structures are always length-bounded and may have either LT or LLID headers.

Parent and Child Structures

Structures can contain other structures within them. The containing structure is known as a parent structure and the contained structures are known as children. These terms are relative, since a non-atomic child structure itself contains other structures and is a parent to them. Children of the same parent are siblings of each other. Parent structures may be length-bounded, delimited, or implied; and may be identified by LTs or LLIDs.

Length-Bounded Parent Structures

In this case, the parent structure has its own header, either an LT or an LLID. Its length includes the lengths of all its children plus the length of its own header. A length-bounded parent exists both as a logical grouping of its children and as an explicit encoded structure at its own encoding level.

Delimited Parent Structures

Sometimes it is convenient to define a group of related structures as existing within a parent structure without having that parent structure appear as a length-bounded structure in the message. The beginning and end of the parent are defined by its first and last children. These children are known as delimiters, the first child is the prefix delimiter and the last is the suffix delimiter. Delimiter children are length-bounded and must be present. They may be null, that is, with an LT of length = 2 or an LLID of length = 4. When the children's headers are LTs, the parent is classified as a delimited LT structure. When they are LLIDs, the parent is a delimited LLID structure.

Implied Parent Structures

It is possible to define a set of related structures as children of a parent structure where the existence and boundaries of the parent are implied by the existence and order of certain child structures. This set of children may occur within the parent structure, either ordered or unordered, until a structure occurs that is not an element of this set. This break in sequence implies the boundary between parent structures. Depending on its children's headers, an implied parent is classified as either implied LT or implied LLID.

Segmented Structures

Length-bounded LLID structures may be either segmentable or non-segmentable. For segmentable structures, the most significant bit of the LL bytes indicates whether any particular segment is the last (bit is equal to 0) or not last (bit is equal to 1) segment of the structure. The ID bytes of the segmentable structure are present on the first segment only.

Properties of Parent Structures

Order

A parent structure may have either ordered or unordered children. Ordered children occur in the parent structure in the same order as they are described in the format description table. Unordered children may occur in the parent structure in any order.

Unrecognized Children

Future enhancements to the formats might add structures that will not be recognized by implementations of the current format definitions. The current format must specify for each parent whether or not unrecognized child structures are allowed. If they are allowed, the definition must specify how long they might be. When unrecognized structures are found where they are allowed, they must be passed through without change at intermediate locations and gracefully ignored at final destinations. Unrecognized structures are identified by either LT or LLID headers, being of the same type as their siblings.

Number of Children

The number of children within a parent may range from a required minimum to an allowed maximum. For example, a parent might have several children, each defined with an occurrence of 0-1, and a number of children defined as 1. This means that any one, but only one, child is allowed.

Header Description Table

The header information and primary syntax associated with each structure are formally described in tabular form. These header description tables represent the formatting information required to either parse or build the SNACR.

Structure Name

The first column of the header description table identifies the SNACR structures, by name, and illustrates their hierarchical relationship by indentation of the column entries. The order of the structure entries in the table represents, unless specified otherwise, the order in which the structures appear in the SNACR datastream.

Structure Reference (Struct Ref)

As header information and primary syntax are described in the header description of a particular table, the semantics, bit representations, presence rules, and other characteristics are described formally in the structure description. This column contains a reference page number to where this structure information is found.

Structure Class (Struct Class)

Structures are classified as either length-bounded LLIDs (ID), length-bounded LTs (T), delimited LLIDs (Del-ID), delimited LTs (Del-T), implied LLIDs (Imp-ID), or implied LTs (Imp-T).

A structure classified as delimited must contain at least two required, length-bounded children that act as the prefix (pfx) and suffix (sfx) of the delimited structure. The "/pfx" notation indicates the length-bounded child structure that serves as the prefix for its parent delimited structure. The "/sfx" notation indicates the length-bounded structure that serves as the suffix for its parent delimited structure.

A structure classified as implied uses an identified child to identify the beginning of a sequence of children. The "/idc" notation indicates the length-bounded structure that serves as an identified child of its parent implied structure.

The "/seg" notation indicates that segmentation is allowed.

ID/T

This column contains the ID or T value within the header, in hexadecimal. To indicate that a delimited structure is identified by its prefix, the notation "pfx" is used. To indicate that an implied structure is identified by one of its children, the notation "idc," for identified child, is used.

Length

This column describes the length verification that would be appropriate at presentation services time. The range of length values specifies the minimum and maximum lengths of structures which an implementation is required to receive. For structures that allow unrecognized children, the maximum length value accommodates the possibility of these yet-to-be-defined structures. On the sending side, the maximum length value for a particular structure may be determined by subtracting the unrecognized reserve, if unrecognized children are allowed, from the maximum length.

Note: An asterisk denotes length restrictions for a particular structure. Length restrictions are detailed in the corresponding structure description.

Occurrences

Multiple occurrences of the SNACR structures may or may not be permitted. A value of "1 - < some number >" in this column indicates the allowed range of occurrences of the corresponding structure. A value of "≥1" indicates that there is no architecturally defined maximum. A value of "1" in this column indicates that only a single instance of the corresponding structure is appropriate. A value of "0 - 1" indicates that an instance of the corresponding structure is optional.

Note: An asterisk denotes presence rules for a particular structure. Presence rules are detailed in the corresponding structure description.

Children

Unrecognized Children Allowed (Unrec): An entry of "Y" in the "Unrec" column indicates that the corresponding structure tolerates unrecognized child structures. An entry of "N" indicates that the particular structure tolerates only the architecturally-defined child structures. An entry of "—" indicates that unrecognized children are not applicable to the particular structure. By definition, atomic structures do not contain children, recognized or not.

Order: A value of "Y" in this column indicates that children are ordered, a value of "N" indicates that children are unordered, and a value of "—" indicates that no children are present.

Note: If a structure is atomic, this column is not applicable.

Number (Num): Each parent structure contains a certain number of different children. This column specifies the minimum and maximum number of different children for a particular parent structure. The maximum number also accounts for unrecognized children, if they are allowed within the parent structure. This column does not account for multiple occurrences of a particular child structure within the parent structure. The number of occurrences of each child is indicated in the "Occurrences" column.

Subtable: Sometimes the need to divide large tables into subtables becomes apparent, particularly when common children appear frequently within different header description tables. This column contains a reference page number to where these common children are described.

Structure Description

The structure description is referenced by a page number appearing in the "Structure Reference" column corresponding to each structure in the header description table. This description contains information pertaining to the data portion of a particular structure. Prose descriptions, presence rules, and semantics associated with the corresponding entry in the header description table may appear in the structure description.

Overview

The SNA Condition Report is a means of encoding exception information for any type of SNA exception. Information about the exception and the location of the exception can be encoded in the SNA Condition Report for the purpose of reporting. The SNA Report Code contains the code and subcode that describe the exception. The location information is encoded in standard structure identification of the Structure Report. When needed, other structures can also be included in the SNA Condition Report to add any other necessary information about the exception.

The SNA Condition Report consists of the following major parts:

SNA_Report_Code
: The SNA_Report_Code is a required SNA registered code identifying the condition that is being reported. The primary report code is placed in bytes 2-3 and the subcode is placed in bytes 4-5.

Structure_Report
: The structure_report contains information about the structure involved in a format related exception.

Reported-On_Dest_List
: Contains the portion of the distribution destinations that are being reported on.

Reported-On_Agent
: Contains the name of the transaction program that is being reported on.

Reported-On_Token_String
: Contains the canonical identifier of a data object related to the detected condition.

Supplemental_Report
: Contains other information about the exception that could not be encoded in any of the structures described above.

The combination of the SNA registered report code and a standard structure identifier creates a powerful tool for reporting format exceptions. Both the reason for the exception and its location in the data stream can be described in a general, independent fashion. This has a secondary benefit of allowing similar exceptions to be reported on with the same report code, even if the exceptions occurred in entirely different data streams.

Header Description Tables

SNA Condition Report

Figure C-1. SNA_Condition_Report as Defined by SNA/FS and CM

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
SNA_Condition_Report	C-9	ID	1532	10-32749	0-1*	Y	Y	1-10	—
SNA_Report_Code	C-9	T	7D	6	1	—	—	—	—
Structure_Report	C-9	T	01	14-255	0-10*	Y	Y	2-10	—
Structure_State	C-10	T	01	3	1	—	—	—	—
Structure_Contents	C-10	T	02	3-100	0-1*	—	—	—	—
Parent_Spec	C-10	T	03	5-17	0-7	N	Y	1-4	—
Parent_ID_Or_T	C-10	T	01	3-4	1	—	—	—	—
Parent_Class	C-11	T	02	3	0-1*	—	—	—	—
Parent_Position	C-11	T	03	4	0-1	—	—	—	—
Parent_Instance	C-11	T	04	4	0-1	—	—	—	—
Structure_Spec	C-11	T	04	5-17	0-1*	N	Y	1-4	—
Structure_ID_Or_T	C-12	T	01	3-4	0-1*	—	—	—	—
Structure_Class	C-12	T	02	3	0-1*	—	—	—	—
Structure_Position	C-12	T	03	4	0-1	—	—	—	—
Structure_Instance	C-12	T	04	4	0-1	—	—	—	—
Structure_Segment_Number	C-13	T	05	4	0-1*	—	—	—	—
Structure_Byte_Offset	C-13	T	06	4	0-1	—	—	—	—
Sibling_List	C-13	T	07	3-100	0-1*	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-241	—	—	—	—	—
Reported-On_Dest_List	C-13	Del-T	pfx	12-11268	0-1*	N	Y	3	—
Reported-On_Dest_Prefix	C-13	T/pfx	08	2	1	—	—	—	—
Reported-On_Dest	C-13	Imp/T	idc	8-5654	≥1	N	Y	1-2	—
Reported-On_Location_Name	C-14	T/idc	09	2-22	1	N	Y	0-2	—
Reported-On_NETID	C-14	T	01	3-10	0-1*	—	—	—	—
Reported-On_Node_ID	C-14	T	02	3-10	0-1*	—	—	—	—
Reported-On_User	C-14	T	0A	8-22	≥0	N	Y	2	—
Reported-On_Naming_Auth	C-14	T	01	3-10	1	—	—	—	—
Reported-On_Individual_ID	C-14	T	02	3-10	1	—	—	—	—
Reported-On_Dest_Suffix	C-15	T/gfx	0B	2	1	—	—	—	—
Reported-On_Agent	C-15	T	04	3-10	0-1*	—	—	—	—
Reported-On_Token_String	C-15	T	02	5-182	0-10*	N	Y	1-10	C-8
Supplemental_Report	C-15	T	03	3-255	0-5*	—	—	—	—
Unrecognized_Reserve	C-15	T	—	2-15826	—	—	—	—	—

Note: * Refer to the structure description for presence rule(s) and length restriction.

Subtable

Tokens

Figure C-2. Subtable Encoding of the Global Name Tokens.

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Tokens									
First_Token	C-16	T	01	3-18	1	—	—	—	—
Second_Token	C-16	T	02	3-18	0-1	—	—	—	—
Third_Token	C-16	T	03	3-18	0-1	—	—	—	—
Fourth_Token	C-16	T	04	3-18	0-1	—	—	—	—
Fifth_Token	C-16	T	05	3-18	0-1	—	—	—	—
Sixth_Token	C-16	T	06	3-18	0-1	—	—	—	—
Seventh_Token	C-16	T	07	3-18	0-1	—	—	—	—
Eighth_Token	C-16	T	08	3-18	0-1	—	—	—	—
Ninth_Token	C-16	T	09	3-18	0-1	—	—	—	—
Tenth_Token	C-16	T	0A	3-18	0-1	—	—	—	—

SNA_Condition_Report

Description: The *SNA_condition_report* describes the condition being reported. The condition is always identified by an *SNA_report_code*.

Certain conditions can be more fully described by supplementary information. Conditions pertaining to one or more structures in a format can have the location and contents of each of those structures specified by a *structure_report*. Certain conditions arise from inconsistencies among multiple portions of the MU. Each portion is described by a separate *structure_report*.

Data objects related to the reported-on condition can be specified in a *reported-on_token_string*. Other information related to the condition can be specified in a *supplemental_report*.

Presence Rule: Occurs when a reportable condition was detected by the agent/server and the agent has determined that reporting is appropriate.

SNA_Report_Code

Description: The *SNA_report_code* is an SNA registered code identifying the condition that is being reported. Refer to the "Sense Data" chapter in the *SNA/Format* book for allowable values and descriptions.

Format: Byte string

Byte	Content
0-1	LT header
2-3	Primary report code
4-5	Subcode

Structure_Report

Description: The *structure_report* reports on a structure involved in a format-related condition. Depending on the condition, the *structure_report* may describe a structure that was present in, or absent from, the reported-on MU.

A format condition has its location in the MU pinpointed by a *structure_spec* and a list of *parent_specs* that define a line-of-descent. The line-of-descent begins with the MU and continues down the parent-child hierarchy to a level as low as the particular condition warrants. A registered ID always appears in a *structure_report*; if the reported-on structure is not itself a registered ID, its line-of-descent is traced up to include a registered ancestor.

Presence Rule: The presence or absence of this structure is governed by the using architecture.

Structure_State

Description:	The *structure_state* indicates whether the reported-on structure was present or absent.
Format:	Hexadecimal code

Byte	Content
0-1	LT header
2	X'01' STRUCTURE_PRESENT
	X'02' STRUCTURE_ABSENT

Note: All other values are reserved.

Structure_Contents

Description:	The *structure_contents* is the portion of the MU that is relevant to the detected condition. Typically, the *structure_contents* contains the header of the structure and at least the beginning of its contents. When the condition can be isolated to a portion of the structure, the *structure_contents* contains only that portion of the structure relevant to the condition. In this case, the *structure_segment_number* and *structure_byte_offset* locate the portion of the structure relevant to the condition.
Presence Rule:	Allowed only when *structure_state* = STRUCTURE_PRESENT.
Format:	Undefined byte string

Parent_Spec

Description:	The *parent_specification* contains the identifier (ID or T) and the class of a parent structure. For a parent structure that occurs multiple times, the instance may also be included. The value of the *parent_instance* identifies the particular instance. The position of this parent structure within its parent (if one exists) may also be included. This would typically be done when this parent structure is an unordered child of its parent.

Parent_ID_Or_T

Description:	The *parent_ID_or_T* is the ID or T value of a parent structure. ID values are the registered GDS code points. T values are architecture-specific values relative to the encompassing ID.
Format:	Undefined byte string

Parent_Class

Description:	The *parent_class* is the class of a parent structure.
Presence Rule:	If absent, defaults to LENGTH-BOUNDED_LT_STRUCTURE.
Format:	Hexadecimal code

Byte	Content	
0-1	LT header	
2	X'01'	LENGTH-BOUNDED_LLID_STRUCTURE (ID)
	X'02'	LENGTH-BOUNDED_LT_STRUCTURE (T) (default)
	X'03'	DELIMITED_LLID_STRUCTURE (DEL-ID)
	X'04'	DELIMITED_LT_STRUCTURE (DEL-T)
	X'05'	IMPLIED_LLID_STRUCTURE (IMP-ID)
	X'06'	IMPLIED_LT_STRUCTURE (IMP-T)

Note: All other values are reserved.

Parent_Position

Description:	The *parent_position* is the position of this parent structure within its parent (if one exists) in this particular MU. Multiple consecutive instances of a repeatable parent structure share a single position, and can be distinguished by *parent_instance*.
Format:	Signed binary integer

Parent_Instance

Description:	The *parent_instance* is used when a parent structure occurs multiple times. The value of *parent_instance* identifies the particular instance within a position.
Format:	Signed binary integer

Structure_Spec

Description:	The *structure_specification* contains the identifier (ID or T) and the class of a structure. For a structure that occurs multiple times, the instance may also be included. The value of the *structure_instance* identifies the particular instance. The position of this structure within its parent structure may also be included. This would typically be done when the parent structure contains unordered children.
Presence Rule:	Absent only when the *structure_class* is the default and the *sibling_list* contains all pertinent ID or T values.

Structure_ID_Or_T

Description:	The *structure_ID_or_T* is the ID or T value of the structure. ID values are the registered GDS code points. T values are architecture-specific values relative to the encompassing ID.
Presence Rule:	Required except when *sibling_list* contains all pertinent ID or T values. In this case, the structures specified by *sibling_list* are the structures being reported on.
Format:	Undefined byte string

Structure_Class

Description:	The *structure_class* is the class of the reported-on structure and any siblings identified in *sibling_list*.
Presence Rule:	If absent, defaults to LENGTH-BOUNDED_LT_STRUCTURE.
Format:	Hexadecimal code

Byte	Content	
0-1	LT header	
2	X'01'	LENGTH-BOUNDED_LLID_STRUCTURE (ID)
	X'02'	LENGTH-BOUNDED_LT_STRUCTURE (T) (default)
	X'03'	DELIMITED_LLID_STRUCTURE (DEL-ID)
	X'04'	DELIMITED_LT_STRUCTURE (DEL-T)
	X'05'	IMPLIED_LLID_STRUCTURE (IMP-ID)
	X'06'	IMPLIED_LT_STRUCTURE (IMP-T)

Note: All other values are reserved.

Structure_Position

Description:	The *structure_position* is either the actual or expected position of this structure within its parent in this particular MU. Multiple consecutive instances of a repeatable structure share a single position, and can be distinguished by *structure_instance*.
Format:	Signed binary integer (1-origin)

Structure_Instance

Description:	The *structure_instance* is used when the structure is one of multiple occurrences of a repeatable structure. The value of *structure_instance* identifies the particular instance within a position.
Format:	Signed binary integer (1-origin)

Structure_Segment_Number

Description:	The *structure_segment_number* is the segment of the structure in which the condition was detected.
Presence Rule:	Occurs when the beginning of *structure_contents* was not contained in the first segment of the reported-on structure.
Format:	Signed binary integer (1-origin)

Structure_Byte_Offset

Description:	The *structure_byte_offset* marks the start of *structure_contents* within the reported-on structure. If *structure_segment_number* is present, this value is the offset from the start of the indicated segment; otherwise, it is the offset from the beginning of the structure.
Format:	Signed binary integer (0-origin)

Sibling_List

Description:	The *sibling_list* contains a string of ID or T values necessary to describe the detected condition. The structures identified in *sibling_list* are children of the parent identified in *parent_spec* and/or siblings of the structure identified in *structure_spec*. The class of the sibling structures is the same as *structure_class*. The expected position, when applicable, is given by *structure_position*.
Presence Rule:	The presence or absence of this structure is governed by the using architecture.
Format:	Byte string

Reported-On_Dest_List

Description:	The *reported-on_destination_list* contains the portion of the distribution destinations that are being reported on.
Presence Rule:	The presence or absence of this structure is governed by the using architecture.

Reported-On_Dest_Prefix

Description:	The *reported-on_destination_prefix* is the prefix of the *reported-on_destination_list*.

Reported-On_Dest

Description:	The *reported-on_destination* associates *reported-on_user*s with a *reported-on_location_name* for those destinations specified in the original distribution request being reported on. For flat destination lists (i.e., lists containing only location names and/or location-user pairs), there are zero or one user names per location list. For factored destination lists, there can be multiple user names per location list.

Reported-On_Location_Name

Description: The *reported-on_location_name* is one of the original destination locations being reported on.

Reported-On_NETID

Description: The *reported-on_NETID* is the first part of the name of one of the original destination locations being reported on.

Format: Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Reported-On_Node_ID

Description: The *reported-on_Node_ID* is the second part of the name of one of the original destination locations being reported on.

Format: Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Reported-On_User

Description: The *reported-on_user* is the name of one of the original destination users being reported on.

Reported-On_Naming_Auth

Description: The *reported-on_naming_authority* is the first part of the name of one of the original destination users being reported on.

Format: Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Reported-On_Individual_ID

Description: The *reported-on_individual_ID* is the second part of the name of one of the original destination users being reported on.

Format: Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Reported-On_Dest_Suffix

Description: The *reported-on_destination_suffix* is the suffix of the *reported-on_destination_list*.

Reported-On_Agent

Description. The *reported-on_agent* is the name of the transaction program that is being reported on.

Presence Rules: Present if needed to identify the reported-on event and when not implied by the context.

Format Character string, except for the first byte.

CGCSGID: 01134-00500 (Character Set AR)

String Convention: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

The first byte of an SNA-registered transaction program name ranges in value from X'00 to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA-registered. X'40' is not a valid first-byte value.

Reported-On_Token_String

Description: The *reported-on_token_string* contains the SNA/FS canonical identifier of a data object related to the detected condition.

Presence Rule: The presence or absence of this structure is governed by the using architecture.

Supplemental_Report

Description: The *supplemental_report* contains other information pertaining to a condition. The contents of the *supplemental_report* are governed by the using architecture.

Presence Rule: The presence or absence of this structure is governed by the using architecture.

Unrecognized_Reserve

Description: The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure.

Format: Undefined byte string

First_Token

Description:	The *first_token* is the highest level part of the data object name. Its values are assigned and registered by SNA.
Format:	Character string

CGCSGID:	01134-00500 (Character Set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Second_Token

Description:	The *second_token* is the second-highest level part of the data object name. The values of this token are assigned by the authority identified by the name in *first_token.*
Format:	Character string

CGCSGID:	01134-00500 (Character Set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Third_Token-Tenth_Token

Description:	The *third_to_tenth_tokens* are the nth highest-level part of the data object name. The value of the nth token is assigned by the authority identified by the name in the (n-1)th token.
Format	Character string

CGCSGID:	01134-00500 (Character Set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Unit of Work Correlator—Overview

The Unit of Work Correlator carries enough information to be a network wide correlator. It is used in situations where multiple requests may be sent over the network before replies are received. In such cases the responding agent uses the Unit of Work Correlator to identify the request that each reply corresponds to, thus allowing the requesting agent to match the responses to the requests.

The Unit of Work Correlator consists of the following parts:

Requester_Location_Name
The name of the location where the request originated.

Requester_User
The user name of the requester.

Requester_Agent
The transaction program that originated the request.

Seqno_DTM
The sequence number assigned to the request by the requesting agent.

Header Description Tables

Unit of Work Correlator

Figure C-3. Unit of Work Correlator as defined by SNA/FS and CM

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Agent_Unit_of_Work	C-19	ID	1549	27-128	1	Y	Y	2-8	—
Requester_Location_Name	C-19	T	01	8-22	1	N	Y	2	—
Requester_Netid	C-19	T	01	3-10	1	—	—	—	—
Requester_Node_ID	C-19	T	02	3-10	1	—	—	—	—
Requester_User	C-19	T	03	8-22	0-1	N	Y	2	—
Requester_Naming_Auth_ID	C-20	T	01	3-10	1	—	—	—	—
Requester_Individual_ID	C-20	T	02	3-10	1	—	—	—	—
Requester_Agent	C-20	T	04	3-10	0-1*	—	—	—	—
Seqno_DTM	C-20	T	02	15-17	1	—	—	—	—
Unrecognized_Reserve	C-21	T	—	2-53	—	—	—	—	—

Note: * Refer to the structure description for presence rule(s).

Structure Descriptions

Agent_Unit_of_Work

Description: The *agent_unit_of_work*, assigned by the requesting agent, provides the basis to track the progress of a particular defined task. The unit-of-work request is uniquely identified by the combination of *requester_location_name*, *requester_user*, *requester_agent*, and *sequence_number|date-time*.

Requester_Location_Name

Description: The *requester_location_name* is the name of the location at which the unit-of-work was requested.

Requester_Netid

Description: The *requester_netid* is the first part of the name of the location at which the unit-of-work was requested.

Format: Character string

CGCSGID: 01134-00500 (Character Set AR)

String Conventions: Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_Node_ID

Description: The *requester_node_ID* is the second part of the name of the location at which the unit-of-work was requested.

Format: Character string

CGCSGID: 01134-00500 (Character Set AR)

String Conventions: Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_User

Description: The *requester_user* is the user name of the originator of the unit-of-work request.

Requester_Naming_Auth_ID

Description:	The *requester_naming_authority_ID* is the first part of the user name of the unit-of-work originator.
Format:	Character string

CGCSGID:	01134-00500 (Character Set AR)
String Conventions:	Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_Individual_ID

Description:	The *requester_individual_ID* is the second part of the user name of the unit-of-work originator.
Format:	Character string

CGCSGID:	01134-00500 (Character Set AR)
String Conventions:	Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_Agent

Description:	The *requester_agent* identifies the transaction program that originated the unit-of-work request.
Presence Rule:	When the *requester_agent* is absent, the originating agent specified in the distribution is the default.
Format:	Character string, except for first byte

CGCSGID:	01134-00500 (Character Set AR)
String Convention:	Leading and imbedded blanks are not allowed; however, trailing blanks are allowed. The first byte of an SNA-registered transaction program name ranges in value from X'00 to X'3F'. When the first byte ranges in value from X'41' to X'FF', the transaction program is not SNA-registered. X'40' is not a valid first-byte value.

Seqno_DTM

Description:	The sequence number is the number assigned to the unit-of-work request by the originating agent. The value ranges from 1 to (2^{31})-1. The date of the unit-of-work request is assigned by the *requester_agent*; the time of the unit-of-work request is assigned by the *requester_location_name*. The offset from GMT for local time is included.
Format:	Byte string

Byte	Contents
0-1	LT header
2-5	Sequence number Signed binary integer limited to (2^{31})-1.
	DATE
6-7	Year, in binary (e.g., year 1989 is encoded as X'07C5')
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 23 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	GMT FLAG
14	Indicates that specified TIME is GMT and identifies whether offsets from GMT are required to calculate local time. (Equivalent EBCDIC characters are shown in parentheses.) X'E9' (Z) no offset required X'4E' (+) add required offset to GMT to get local time X'60' (−) subtract required offset from GMT to get local time
	OFFSET
15	Hour offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 23 are valid)
16	Minute offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 59 are valid)

Examples

A 9-byte date/time encoding is a date/time followed immediately by an EBCDIC 'Z', and is considered to be GMT. Thus, 12:00 GMT on 2 January 1988 would be

```
X'07C401020C000000E9'
  yyyymmddhhmmsshhZ
```

An 11-byte date/time encoding is a date/time followed immediately by an EBCDIC '+' or '-' and two one-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00 a.m. on 2 January 1988 in New York would be 12:00 GMT - 5 hours, or

```
X'07C401020C000000600500'
  yyyymmddhhmmsshh- hhmm
```

Unrecognized_Reserve

Description:	The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure.
Format:	Undefined byte string

Appendix D. List of Abbreviations and Symbols

A

A	address (SDLC)
ACT	active, activate
APPN	advanced peer-to-peer networking

B

B'nnnn'	binary digits (usually shown simply as nnnn)
BB	begin bracket
BBI	begin bracket indicator
BC	begin chain
BCI	begin chain indicator
BETB	between brackets
BF	boundary function
BIU	basic information unit
BLU	basic link unit
BSC	Binary Synchronous Communication
BTU	basic transmission unit

C

(c)	configuration services
C	control (SDLC)
CCA	communication controller adapter
CCITT	International Telegraph and Telephone Consultative Committee
CD	cross-domain, change direction
CDI	change direction indicator
CDRM	cross-domain resource manager
CEB	conditional end bracket
CEBI	conditional end bracket indicator
CICS/VS	Customer Information Control System/Virtual Storage
CMI	compression indicator
CNOS	change number of sessions
CONT	contention
COS	class of service; common operations services
CP	control point
CPI	compaction indicator
CRC	cyclic redundancy check
CRV	cryptography verification
CS	configuration services
CSI	code selection indicator
CSP	control sequence prefix
CV	control vector

D

DAF	destination address field
DCE	data circuit-terminating equipment
DCF	data count field

DD	day of month
ddd	day of year
DEF	destination element address field
DEN	distribution user element name (SNADS)
DES	Date Encryption Standard
DFC	data flow control
DGN	distribution user group name (SNADS)
DISC	Disconnect (SDLC)
DISOSS	Distributed Office Support System
DISTIU	distribution interchange unit (SNADS)
DLC	data link control
DLCI	Data Link Connection Identifier
DLU	destination logic unit
DM	Disconnected Mode (SDLC)
DPN	destination program name
DQ	dequeue
DR1I	definite response 1 indicator
DR2I	definite response 2 indicator
DS	distribution services
DSAF	Destination Subarea Address Field
DSP	data stream profile
DST	data services task or device service task
DSU	distribution service unit (SNADS)
DTE	data terminal equipment

E

EB	end bracket
EBCDIC	extended binary coded decimal interchange code
EBI	end bracket indicator
EC	end chain
ECI	end chain indicator
ED	enciphered data
EDI	enciphered data indicator
EFI	expedited flow indicator
EN	end node
ENA	extended network addressing
ENCP	end node control point
ENP	Enable Presentation
ER	explicit route
ERP	error recovery procedures
ERCL	exchange record length
ERI	exception response indicator
ERN	explicit route number
ERP	error recovery procedures
Exp	expedited flow
EXR	EXCEPTION REQUEST

F

F	flag (SDLC)
FCB	forms control block

FCS	frame check sequence (SDLC)
FDX	full-duplex data flow
FF	flip-flop direction control
FFR	field-formatted record
FI	format indicator
FID	format identification
FIFO	first-in, first-out
FM	function management
FMD	function management data
FMDS	function management data services
FMH	function management header
FMHC	function management header concatenation
FNI	fixed fields without field separators
FRMR	Frame Reject (SDLC)
FRSE	Frame-relay switching equipment
FRTE	Frame-relay terminating equipment
FS	fixed fields with field separators
FS2	fixed fields with or without field separators

G

GDS	general data stream
GE	greater than or equal to

H

HDX	half-duplex data flow
hex	hexadecimal
HH	hours
HPCA	High-Performance Communication Adapter

I

I	information (SDLC), initiate only
ID	identification
IERN	initial explicit route number
ILU	initiating logical unit
IMS/VS	Information Management Systems/Virtual Storage
INB	in bracket
INP	Inhibit Presentation
IPL	initial program load
IPM	ISOLATED PACING MESSAGE
IPR	ISOLATED PACING RESPONSE
I/Q	initiate or queue
IRS	interchange record separator
ISO	International Organization for Standardization
IU	interchange unit (SNADS)

K

KEYIND	key indicator

L

LAN local-area network
LCID local coded graphic character set identifier
LH link header
LIFO last-in, first-out
LL logical record length (prefix)
LMS logical messages services
LRH logical record header
LT link trailer; less than
LSID local session identification
LU logical unit
LVx variable length parameter
LV1 variable length parameter, first position

M

(ma) maintenance services
MGR manager
MM month, minutes
MPC maximum presentation column
MPF mapping field (BIU segments)
MPL maximum presentation line

N

NA network address
NAU network addressable unit
NC network control
NN network node
NNCP network node control point
Norm normal flow
NS network services
NUMRECS number of records

O

OAF origin address field
ODAI OAF'-DAF' assignor indicator
OEF origin element field
OII office information interchange
OLU originating logical unit
OSAF origin subarea field

P

P primary
PC path control
PCID procedure correlation identifier
PD padded data
PDI padded data indicator
PDIR peripheral data information record
PI pacing indicator
PIP program initialization parameter
PIU path information unit
PLU primary logical unit

POC	Program Operator Communication
PPU	primary physical unit
PRI	primary
PRID	procedure related identifier
PRN	primary resource name
PRTY	priority
PS	presentation services
PSH	presentation services header
PU	physical unit
PUCP	physical unit control point
P/F	poll/final (SDLC)

Q

Q	queue
QC	quiesce complete
QEC	quiesce at end of chain
QR	queued response
QRI	queued response indicator

R

RCV	receive
RD	Request Disconnect (SDLC)
REC	receive
RECLEN	record length
RECID	record identification
RECTYPE	record type
REJ	Reject (SDLC)
RELQ	release quiesce
REN	routing element name (SNADS)
REQECHO	Request Echo Test
RH	request/response header
RIM	Request Initialization Mode (SDLC)
RJE	remote job entry
RLSD	released
RNR	Receive Not-Ready (SDLC)
RQ	request
RQD	definite-response request
RQE	exception request
RQR	request recovery
RR	Receive Ready (SDLC)
RRI	request/response indicator
RSP	response
RTI	response type indicator (+/-)
RTR	Ready To Receive (SDLC)
RU	request/response unit

S

S	secondary
(s)	session services
SC	session control

SCB	string control byte
SCS	SNA character string
SDI	sense data included indicator
SDLC	Synchronous Data Link Control
SEC	secondary
SESS	session
SIM	Set Initialization Mode (SDLC)
SLU	secondary logical unit
SNA	Systems Network Architecture
SNC	sense code
SNF	sequence number field
SNI	SNA network interconnection
SNADS	SNA distribution services
SNRM	Set Normal Response Mode (SDLC)
SPC	sync point command
SPU	secondary physical unit
SQN	sequence number
SRI	stack reference indicator
SS	seconds
SSCP	system services control point
STP	service transaction program
SU	shared; unnamed
SVC	services

T

T1	type-1 (node)
T2	T2.0 or T2.1 (node)
T2.0	type-2.0 (node)
T2.1	type-2.1 (node)
T4	type-4 (node)
T5	type-5 (node)
TC	transmission control
TERM	terminate
TEST	Test (SDLC)
TG	transmission group
TGN	transmission group number
TH	transmission header
TLU	terminating logical unit
TPF	transmission priority field
TPN	transaction program name
TRN	transparent
TS	transmission services
TWX	teletypewriter exchange service

U

UA	Unnumbered Acknowledgment (SDLC)
UI	Unnumbered Information (SDLC)
UNAVL	unavailable
UP	Unnumbered Poll (SDLC)
URC	user request correlation

V

VD	variable-length positional parameter
VOLID	volume identification
VR	virtual route
VRID	virtual route identifier
VRN	virtual route number
VRPRQ	virtual route pacing request
VRPRS	virtual route pacing response
VT	vertical tab

W

WP	word processing

X

XID	Exchange Identification (SDLC)
X'n...n'	hexadecimal digits
XMIT	transmit
XRF	Extended Recovery Facility

Y

YY	year

Special Characters

|	(vertical stroke) exclusive or
*	(asterisk) any value
¬	(not sign) logical not
_	(underscore) separates multiple terms, or qualifiers, in a phrase

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

X

Readers' Comments

Systems Network Architecture Formats

Publication No. GA27-3136-12

Use this form to tell us what you think about this manual. If you have found errors in it, or if you want to express your opinion about it (such as organization, subject matter, appearance) or make suggestions for improvement, this is the form to use.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer. This form is provided for comments about the information in this manual and the way it is presented.

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

Be sure to print your name and address below if you would like a reply.

Name

Address

Company or Organization

Phone No.

IBM®

Cut o
Along

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

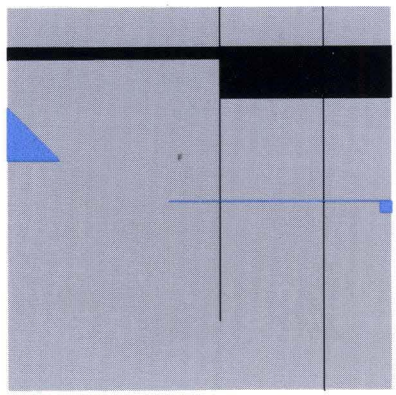
International Business Machines Corporation
Department C40/Building 673
PO Box 12195
Research Triangle Park, N.C. 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

Cut or
Along l

File Number: GENL-30 (SNA)
Program Numbers: 5743-SNA

Printed in USA



GA27-3136-12