IBM

Systems Network Architecture

GA27-3136-10

Formats

IBM

Systems Network Architecture

GA27-3136-10

Formats

Eleventh Edition (June 1989)

This edition, GA27-3136-10, is a major revision of the previous edition, GA27-3136-9, and obsoletes that edition; it applies until otherwise indicated in a new edition. Consult Part 3 of the latest edition of *IBM System/370, 30xx, and 4300 Processors — Bibliography*, GC20-0001, for current information on this communication architecture. For a summary of the changes in this book, see "Summary of Changes."

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

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

IBM Corporation
Department E89
P.O. Box 12195
Research Triangle Park, North Carolina 27709, U.S.A.

Special 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 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 IBM agreements, such as the Agreement for Purchase of IBM Machines and the Agreement for Lease or Rental of IBM Machines.

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 using peer-to-peer protocols.

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. Figure 1-1 on page vii illustrates the organization of this book.

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 an SDLC and System/370 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, session keys, and management services vectors that appear in a request or response unit.

Chapter 9 explains the meanings of the sense data defined by System Network Architecture (SNA) that appear, for example, in negative response units.

Chapter 10 presents the descriptions and formats of the different function management headers.

Chapter 11 identifies the formats and meanings of the bytes in a presentation services header.

Chapter 12 identifies the formats and meanings of the general data stream (GDS) variables that are specific to SNA service transaction programs.

Chapter 13 identifies the formats and meanings of the message units that SNA/Distribution Services transaction programs use.

Chapter 14 identifies the general data stream (GDS) variables that are for general use.

Chapter 15 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.

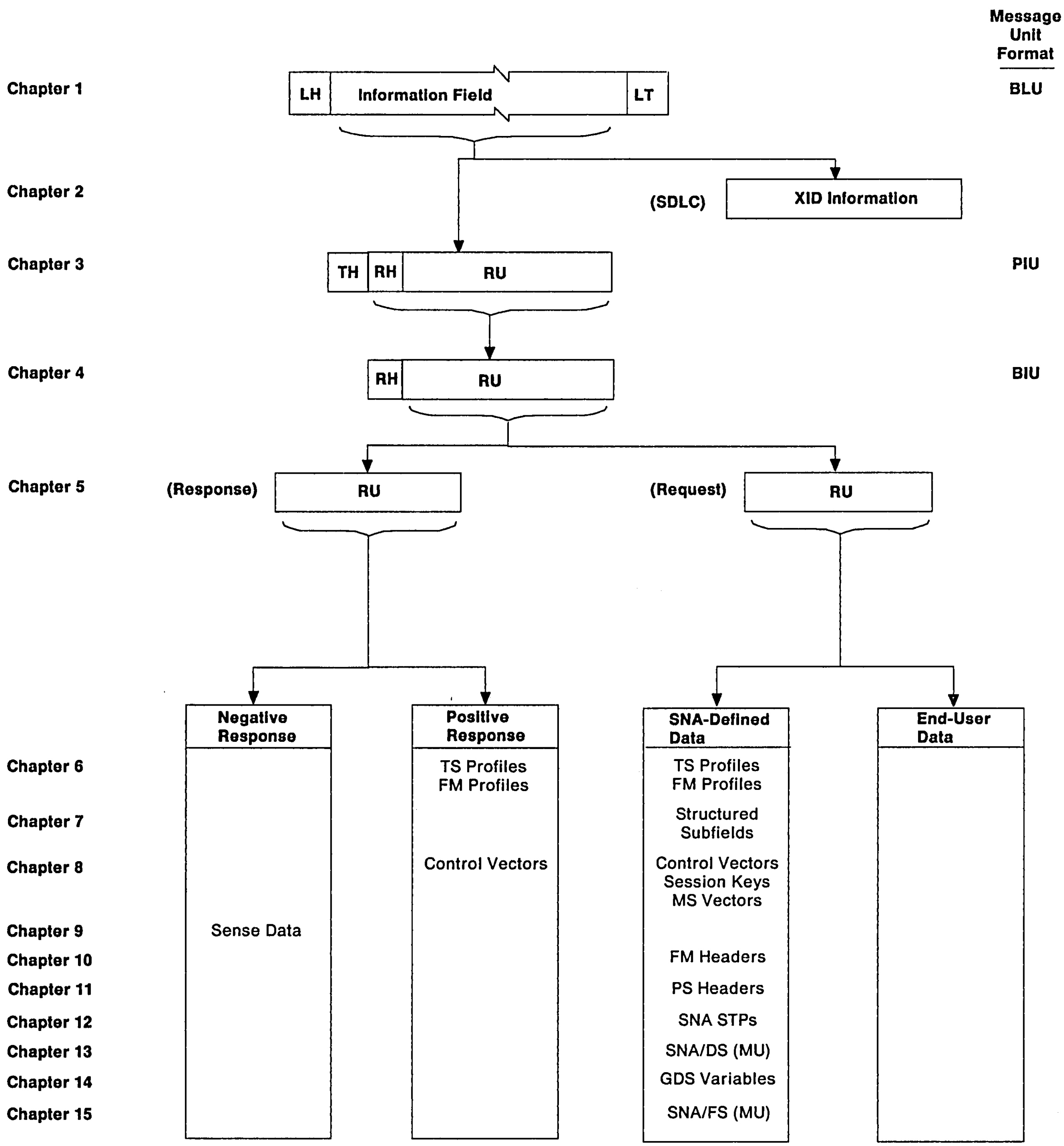
Appendix B provides a summary of general data stream identifier (GDS ID) value assignments.

Appendix C lists the 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 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 (SC30-6808)
- *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)



LH = Link Header
LT = Link Trailer
XID = Exchange Identification
TH = Transmission Header
RH = Request/Response Header
RU = Request/Response Unit
TS = Transmission Services
FM = Function Management
MS = Management Services

PS = Presentation Services
STP = Service Transaction Program
SNA/DS = SNA/Distribution Services
MU = Message Unit
GDS = General Data Stream
BLU = Basic Link Unit
PIU = Path Information Unit
BIU = Basic Information Unit
SNA/FS = SNA/File Services

Figure 1-1. Organization of this Book

Summary of Changes

Additions for GA27-3136-10:

This edition includes information about:

- Network Asset Management
- Change Management
- Common Operations Services
- SNA/File Services

Changes from the previous edition:

- The style of presentation of the formats has been changed.
- An enhanced format set has been provided for SNA/Distribution Services
- Chapters 5.1 and 5.2 have been combined into Chapter 5.

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)

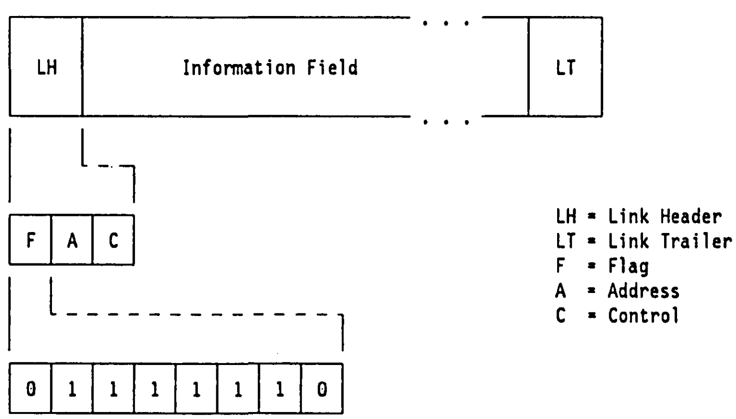


Figure 1-1. Flag Field of Link Header. Always X'7E', B'01111110'.

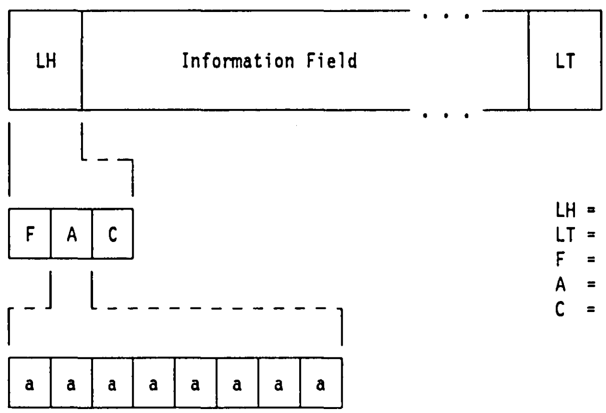
All frames begin with a Flag field. The configuration of the flag is always 01111110 (X'7E'). Because frames also *end* with flags (see link trailer), the trailing flag of one frame may serve as the leading flag of the next frame. When receiving, the last 0 in the trailing flag may also be the first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

```
                |—leading flag-|
 0 1 1 1 1 1 1 0 1 1 1 1 1 1 0
|-trailing flag-|
```

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)




```
LH = Link Header
LT = Link Trailer
F  = Flag
A  = Address
C  = Control
```

Figure 1-3. Address Field of Link Header. B'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', B'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)

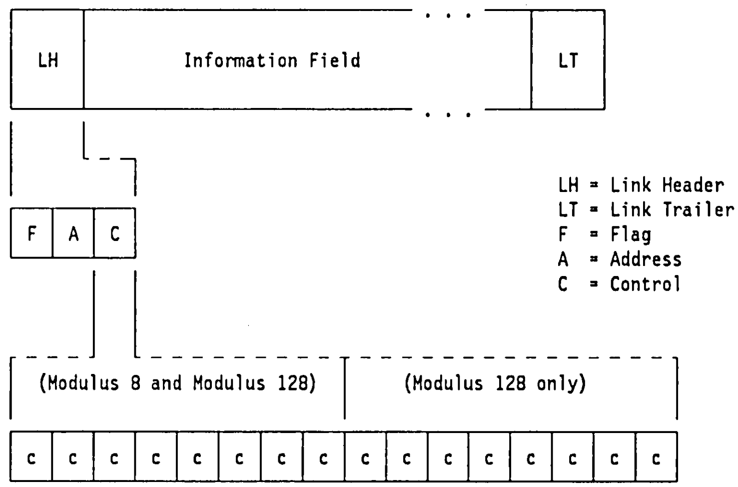


Figure 1-4. Control Field of Link Header. B'cccccccc' for modulus 8; B'cccccccc cccccccc' for modulus 128.

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'x1', X'x1'	Receive Ready	RR
	RRR P/F 0101	X'x5', X'x5'	Receive Not Ready	RNR
	RRR P/F 1001	X'x9', X'x9'	Reject	REJ
Information Format	RRR P/F SSS0	X'xx', X'xx'	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)

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'01xx'	Receive Ready	RR
	0000 0101 RRRR RRR P/F	X'05xx'	Receive Not Ready	RNR
	0000 1001 RRRR RRR P/F	X'09xx'	Reject	REJ
Information Format	SSSS SSS0 RRRR RRR P/F	X'xxxx'	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)				

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

Link Trailer (Frame Check Sequence)

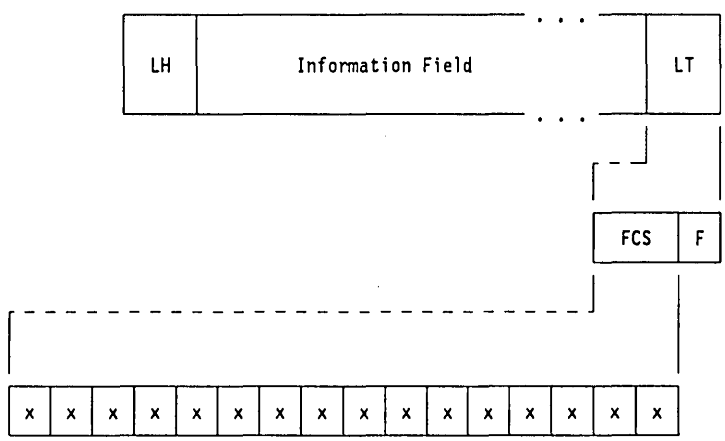


Figure 1-8. Frame Check Sequence Field of Link Trailer

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:

Control Field	Nr	0	Ns	0	0 0 0 0	z y x w
Byte 0	Byte 1				Byte 2	

Modulus 128:

Control Field	Ns	0	Nr	0	0 0 0 0	z y x w
Byte 0 Byte 1	Byte 2		Byte 3		Byte 4	

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 128. In each byte, the low order bit is sent first and the high order bit is sent last.

Link Trailer (Flag)

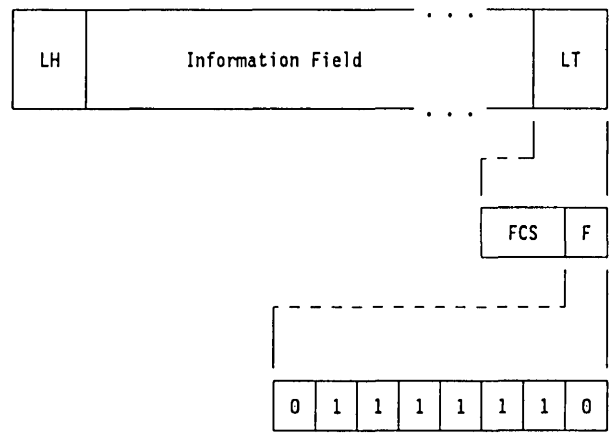


Figure 1-9. Flag Field of Link Trailer. Always X'7E', B'01111110'.

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: 01111110 (X'7E').

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/BF and T2.0 to T5 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		<u>Node Identification</u>
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:* 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:* 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

DLC XID Information Field

Byte	Bit	Content
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
	4−5	Reserved
	6	Short-hold status (reserved if byte 9, bit 7 is set to 0): 0 sender not already engaged in a logical connection using short-hold mode on this port 1 sender already engaged in a logical connection using short-hold mode on this port
	7	Short-hold capability of the XID sender: 0 short-hold mode not supported 1 short-hold mode supported
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
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:

DLC XID Information Field

Byte	Bit	Content
		• 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 the PU in its node receives a DISCONTACT request from its CP. *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 the PU in its node receives a DISCONTACT request from its CP.
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

DLC XID Information Field

Byte	Bit	Content
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≤n≤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≤n≤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:
	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 1 BIND may not be sent to the XID sender *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

DLC XID Information Field

Byte	Bit	Content
	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
	9−11	Reserved
	12−13	XID exchange state: 00 exchange state indicators not supported (set only by implementations not at the current level of SNA) 01 negotiation-proceeding 10 prenegotiation exchange 11 nonactivation exchange
	14−15	Reserved
10		BIND pacing support over the link:
	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 values for 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; 10 is not used; and 11 means adaptive BIND pacing is fully supported.
	2−7	Reserved
11−16		Reserved
17		DLC type: X'01' SDLC X'02' System/370 channel to controller DLC
18−n		<u>DLC Dependent Section</u>
18		Length, in binary, of the DLC Dependent Section field (Length field includes itself in the length specified.)
For SDLC		
19		<u>Link Station and Connection Protocol Flags</u>
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

DLC XID Information Field

Byte	Bit	Content
	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 negotiable (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		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
	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 the PU in its node receives a DISCONTACT request from its CP. *Note 2:* Reject is required only if both sender and receiver have two-way simultaneous transmit-receive capability.

DLC XID Information Field

Byte	Bit	Content
		• 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 the PU in its node receives a DISCONTACT request from its CP.
		• 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 for the send and receive sequence counts—less than 8 implies a modulus of 8; 8 or greater implies a modulus of 128
28(=n)		Reserved
		End of DLC Dependent Section for SDLC

DLC XID Information Field

Byte	Bit	Content
For Channel DLC (CDLC): System/370 Channel between T4 and T2.1 nodes		
		Note: The System/370 node always contains the primary link station for 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 and 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 time-out, and time units
	1	Attention time-out support (set by the secondary; reserved for the primary): 0 not supported 1 supported
	2	Channel data streaming support by the XID sender: 0 not supported 1 supported
	3	Change CDLC parameters support by the XID sender by means of a nonactivation XID exchange (see bit 0): 0 not supported 1 supported
	4−15	Reserved
21−22		Maximum link PIU (LPIU) size: length of the maximum LPIU that the XID sender can receive
23		Buffer pre-fetch: number of buffers suggested for the secondary to pre-allocate 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 time-out (ATO): maximum interval that a secondary awaits a read channel program after presenting an Attention to the primary; if the time-out expires, a secondary-detected inoperative station condition is declared. This time-out value is also used for idle detection (1/2 ATO is used), second-chance Attention (1/2 ATO is used), and primary-detected inoperative station (3/2 ATO is used) *Note:* The secondary has the option of presenting a second Attention, called a *second-chance Attention*, to handle the case of loss of the first Attention.
32−33		Previous number of Read commands: set by the secondary in an XID sent in reply to a change-CDLC-parameters nonactivation XID to specify the number-of-Read-commands parameter (see bytes 24−25) that was active prior to the change; otherwise, reserved

DLC XID Information Field

Byte	Bit	Content
34−35		Previous primary buffer size: set by the secondary in an XID sent in reply to a change-CDLC-parameters nonactivation XID to specify the primary-buffer-size parameter (see bytes 26−27) that was active prior to the change; otherwise reserved
36(=n)		Time units used for Attention time-out and blocking delay: X'00' 100-millisecond time units X'01' 1-millisecond time units
		End of DLC Dependent Section for Channel DLC
n+1−p		Control vectors, as described in "Control Vectors" on page 8-4 *Note:* The following control vectors may be included: X'0E' PU Name control vector: type X'F1', not network-qualified PU name (maximum of 8 bytes may be sent from a T4/T5 node) 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 CP name are 3 to 17 bytes with an imbedded period) X'10' Product Set ID control vector (always present) *Note:* When included in XID, the product set ID is limited to 60 bytes or less in length. X'22' XID Negotiation Error control vector (present when an error during XID negotiation is detected; more than one may be present)

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

Byte		
0	FID2—Format Identification MPF—Mapping Field ODAI—OAF'-DAF' Assignor Indicator EFI—Expedited Flow Ind.	Reserved Byte
2	DAF'—Destination Address	OAF'—Origin Address
4	SNF—Sequence Number Field	

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 Format and Protocol Reference Manual: Architecture Logic for Type 2.1 Nodes* 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 4-9 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 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 6.2's, 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

A: A 16-bit number (1−65,535) incremented by 1 for each request and wrapping through 0 thereafter

FID2 Field Descriptions

Byte	Bit	Content
		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*.

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 5-41 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.

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

Request/Response Header

Byte 0

RRI =0	RU Category	r	FI	SDI	BCI	ECI	Request
RRI =1	RU Category	r	FI	SDI	1	1	Response

Byte 1

Request	DR1I	r	DR2I	ERI	r	RLWI	QRI	PI
Response	DR1I	r	DR2I	RTI	r	r	QRI	PI

Byte 2

Request	BBI	EBI	CDI	r	CSI	EDI	PDI	CEBI
Response	r	r	r	r	r	r	r	r

r = Reserved

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
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 DR11I, 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 10, "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 5-5 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 9, "Sense Data" on page 9-1. 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 9, "Sense Data" on page 9-1.)

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."

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.

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.

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.

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 * subarea type RUs only
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. The formats of character-coded FMD NS requests are implementation dependent. LU → LU FMD requests (e.g., FM headers) are described in Chapter 7, "User Data Structured Subfields" and Chapter 8, "Common Fields."
3. All values for field-formatted requests that are not defined in this section are reserved.
4. 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.
5. 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.
6. 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.
7. User data, control vectors, and session keys referred to in the format descriptions are described in 7-1 and 8-1.

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 and 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(c) Requests

REQDISCONT

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'41038D'	NMVT
X'010604'	NSPE	X'810620'	NOTIFY
X'010681'	INIT-SELF (Format 0)	X'810681'	INIT-SELF (Format 1)
X'010683'	TERM-SELF (Format 0)	X'810683'	TERM-SELF (Format 1)

DFC, NC, and SC Request Codes

X'04'	LUSTAT (DFC)	X'83'	CANCEL (DFC)
X'05'	RTR (DFC)	X'84'	CHASE (DFC)
X'0D'	ACTLU (SC)	X'A0'	SDT (SC)
X'0E'	DACTLU (SC)	X'A1'	CLEAR (SC)
X'11'	ACTPU (SC)	X'A2'	STSN (SC)
X'12'	DACTPU (SC)	X'A3'	RQR (SC)
X'31'	BIND (SC)	X'C0'	SHUTD (DFC)
X'32'	UNBIND (SC)	X'C0'	CRV (SC)
X'70'	BIS (DFC)	X'C1'	SHUTC (DFC)
X'71'	SBI (DFC)	X'C2'	RSHUTD (DFC)
X'80'	QEC (DFC)	X'C8'	BID (DFC)
X'81'	QC (DFC)	X'C9'	SIG (DFC)
X'82'	RELQ (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: 10 ERP (only value defined)
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
	4–7	Type activation requested: X'2' ERP
2	0–3	FM profile: X'0' FM profile 0
	4–7	TS profile: X'1' TS profile 1

ACTPU (ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
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	PU type of the node containing the SSCP
	8−47	Implementation and installation dependent binary identification

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.

The description below is complete for LU 6.2; for other LU types, see *SNA: Sessions between Logical Units* for additional details.

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

BIND (BIND SESSION)

Byte	Bit	Content
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)
		<u>FM Usage—Primary LU Protocols for FM Data</u>
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
	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	Compression indicator (reserved for LU 6.2): 0 compression will not be used on requests from primary 1 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
		<u>FM Usage—Secondary LU Protocols for FM Data</u>
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

BIND (BIND SESSION)

Byte	Bit	Content
	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	Compression indicator (reserved for LU 6.2): 0 compression will not be used on requests from secondary 1 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
		<u>FM Usage—Common LU Protocols</u>
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 sent-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: 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').

BIND (BIND SESSION)

Byte	Bit	Content
	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
	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:* 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). *Note:* Contention winner is also brackets first speaker.

BIND (BIND SESSION)

Byte	Bit	Content
	4−5	Alternate code processing identifier (reserved unless Alternate Code Set Allowed indicator (byte 6, bit 4) is 1): 00 process alternate code FMD RUs as ASCII-7 01 process alternate code FMD RUs as ASCII-8 (only value defined for LU 6.2) *Note:* When the Alternate Code Processing Identifier indicator is set to the value 01, the entire FMD request RU is to be translated using the transforms defined by the ANSI X3.26 Hollerith Card Code.
	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)
	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 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).
		<u>TS Usage</u>
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 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, i.e., when control vector X'60' is not present): 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 in those fields 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 in those fields 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

BIND (BIND SESSION)

Byte	Bit	Content
		Note: Adaptive pacing is supported only in conjunction with one-stage session-level pacing. 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.
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' = $a \times 2^b$ (Notice that, by definition, a≥8 and therefore X'ab' is a normalized floating point representation.) See Figure 5-1 on page 5-15 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 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.)

BIND (BIND SESSION)

Byte	Bit	Content
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
		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.*
15		LU-6 level: X'02' Level 2 (i.e., LU 6.2)
16−22		Reserved
23	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	Reserved *Note:* This byte is used for security information only.

BIND (BIND SESSION)

Byte	Bit	Content
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 indicator: 0 the contention-winner LU will not deactivate the limited resource session 1 the contention-winner LU will deactivate the limited resource session when it is no longer busy
	2−7	Reserved
		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 *Note:* Only values 00 and 11 are defined for LU 6.2.
	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

BIND (BIND SESSION)

Byte	Bit	Content
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
	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 eight-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FMD requests
k+1−m		<u>Primary LU Name Field</u> (always present)
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 the INIT-SELF (or INIT-OTHER), INIT-SELF, the uninterpreted name as carried in that RU (and also in CDINIT for a cross-domain session)
m+1−n		<u>User Data Field</u>
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) ¬X'00' first byte of unstructured user data *Note:* Individual structured subfields may be omitted entirely. When present, they appear in ascending subfield-number order.
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 Chapter 7, "User Data Structured Subfields.")
n+1−p		<u>User Request Correlation Field</u> (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		<u>Secondary LU Name Field</u> (present only for negotiable BINDs and for non-negotiable BINDs that include control vectors)
p+1		Length of secondary LU name (values 1 to 17 are valid) *Note:* Value 0 is retired.
p+2−r		Secondary LU name

BIND (BIND SESSION)

Byte	Bit	Content

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 8-4

Note: The following control vectors may be included; they are parsed according to subfield parsing rule KL:

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' (conditionally present: used 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'2C'	COS/TPF control vector (conditionally present)
X'2D'	Mode control vector (conditionally present, used in non-LU6.2 extended BIND)
X'60'	Fully-qualified PCID control vector (when present, the BIND is called an *extended BIND*)

Note: The receiving LU simply ignores unrecognized control vectors.

Note 1: 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 2: 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). 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		<u>Uninterpreted Name of DLU</u>
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		<u>User Field</u>
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.)
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	0−1	Type: 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
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

INIT-SELF Format 1 (INITIATE-SELF)

Byte	Bit	Content
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

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)

LUSTAT (LOGICAL UNIT STATUS)

Byte	Bit	Content	
		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
		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

NMVT (NETWORK MANAGEMENT VECTOR TRANSPORT)

Byte	Bit	Content
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
	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 "MS Major Vectors and Unique Subvectors" on page 8-12.

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

NOTIFY (NOTIFY)

Byte	Bit	Content
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

NOTIFY Vectors (Described 0-origin)

ILU/TLU Notification NOTIFY Vector

ILU/TLU Notification NOTIFY Vector

Byte	Bit	Content
0		Key: X'03'
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

ILU/TLU Notification NOTIFY Vector

Byte	Bit	Content
		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 Key" on page 8-11 *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

LU-LU Session Services Capabilities NOTIFY Vector

Byte	Bit	Content
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
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

NSPE (NS PROCEDURE ERROR)

Byte	Bit	Content
		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 Key" on page 8-11 *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
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)

(Retired RU)

RECFMS has been retired from SNA for T2 nodes. Consult product documentation for further information and support.

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)

(Retired RU)

REQMS has been retired from SNA for T2 nodes. Consult product documentation for further information and support.

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
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

STSN (SET AND TEST SEQUENCE NUMBERS)

Byte	Bit	Content
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
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

TERM-SELF Format 0 (TERMINATE-SELF)

Byte	Bit	Content
	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 sessions for which DLU is PLU 01 select sessions for which DLU is SLU 10 select sessions 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		<u>Uninterpreted Name of DLU (retired):</u>
4		Type: X'F3' logical unit
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

TERM-SELF Format 1 (TERMINATE-SELF)

Byte	Bit	Content
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 sessions for which DLU is PLU 01 select sessions for which DLU is SLU 10 select sessions 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 Key" on page 8-11 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
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'.

UNBIND (UNBIND SESSION)

Byte	Bit	Content
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 8-4 *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL: X'35' Extended Sense Data control vector (present when the UNBIND Type is X'06', X'FE', or is immediate, i.e., X'07' through X'7F') 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 9, "Sense Data" on page 9-1 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 5-41 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"

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

2. 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.

3. 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 individ-

ually); 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.

4. User data, control vectors, and control lists referred to in the format descriptions are described in Chapter 7, "User Data Structured Subfields" on page 7-1 and Chapter 8, "Common Fields" on page 8-1.

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'02' ERP (only value defined)
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 8-4 *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. When present, they appear in the order specified. X'00' SSCP-LU Session Capabilities control vector (always present, always first) X'0C' LU-LU Session Services Capabilities control vector (always present, 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
1	0−1 2−3 4−7	Reserved Format of response: 00 format 0 (only value defined) Type activation selected: X'2' ERP (only value defined)
2−9		Contents ID: eight-character EBCDIC symbolic name of the load module currently operating in the node; eight space (X'40') characters is the default value

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
1	0−3	Format: 0000 (only value defined)
	4−7	Type: 0000 negotiable (only value defined for LU 6.2) 0001 nonnegotiable
2−25		Bytes 2−25 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
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 as in BIND (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
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

RSP(BIND)

Byte	Bit	Content
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 8-4 *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL: X'0E' Network Name control vector: CP network name (conditionally present, used in extended BIND responses when neither the Fully-Qualified PCID [X'60'] control vector nor the Route Selection [X'2B'] control vector contain the CP(PLU) name) X'60' Fully-Qualified PCID control vector (present if received on the BIND)
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
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

RSP(BIND)

Byte	Bit	Content
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 8-4 *Note:* The following control vectors may be used; they are parsed according to subfield parsing rule KL: X'60' Fully-Qualified PCID control vector (always present)

RSP(STSN)

SLU → PLU, Exp; SC

RSP(STSN)

Byte	Bit	Content
0		X'A2' request code

RSP(STSN)

Byte	Bit	Content
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)
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 sessions, SSCP-LU sessions, and SSCP-PU sessions to Type 1, 2, or 2.1 nodes. 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	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 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 sessions; SSCP-PU sessions to Type 1, 2, or 2.1 nodes; 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	6.2

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

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 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 compression.
- 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)

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)
- 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)

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)
- 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

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)
- 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)

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)
- 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 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 compression.
- 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
- 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
- Alternate code permitted.

The only FM Usage field defining options for Profile 19 is Contention Winner/Loser.

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, Session Qualifier, Random Data, Enciphered Data, 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 (If Session Qualifier subfield is present, 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.

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: 0 to 8 Type-1134 symbol-string characters (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). The symbol string may be assigned to a longer field and padded with blanks on the right, but these trailing blanks are not considered part of the Mode 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		<u>Session Instance Identifier</u>
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).

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 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 Clear Data field 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
- Management services major vectors and subvectors

Substructure Encoding/Parsing Rules

Rules for Common Substructures

The following rules apply to encodings defined in this chapter; they govern the encoding of SNA-defined RU substructures, i.e., structures 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-Dependent 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, 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. 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 Header field.

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 Header field 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")
2–n		Vector Data
When the enclosing structure indicates use of parsing rule KL, the first two bytes are interpreted as:		
0		Key

Example of common substructure format

Byte	Bit	Content
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)

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 key value, the control vector and the message-unit structures that can carry the control vector.

Key	Control Vector	Applicable Message-Unit Structures
X'00'	SSCP-LU Session Capabilities	RSP(ACTLU)
X'0C'	LU Session Services Capabilities	RSP(ACTLU)
X'0E'	Network Name	XID, BIND
X'10'	Product Set ID	XID
X'22'	XID Negotiation Error	XID
X'2C'	COS/TPF	BIND
X'2D'	Mode	BIND
X'35'	Extended Sense Data	UNBIND
X'60'	Fully Qualified PCID	BIND, UNBIND, RSP(BIND)

Control Vector Formats

The control vectors are defined as follows (with 0-origin indexing of the vector bytes—see the individual RU description for the actual displacement within the RU):

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 5-15 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

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 8-2)
2−m		Vector Data
2	0−3	(reserved)
	4−7	Secondary LU capability: 0000 SLU capability is inhibited: sessions can neither be 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 8-2) *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 X'F3' LU name X'F4' CP name X'F7' link station name (not network-qualified)
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 no imbedded blanks and with optional (but not significant) trailing blanks.

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 8-2)
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 8-165, 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 8-2)
2−n		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(=n)		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

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 8-2)
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 of COS Name field
4−m		COS name: 0 to 8 type-A symbol-string characters with optional (but not significant) trailing blanks

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 8-2)
2−n		<u>Vector Data</u>
2		Length of Mode Name field
3−n		Mode name: 0 to 8 type-A symbol-string characters with optional (but not significant) trailing blanks

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 8-2)
2−p		<u>Vector Data</u>
2−5		Sense data *Note:* The shorter abbreviated form (now retired) of the control vector ends here.
6−p		<u>Extended Sense Information</u>
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 (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
	4	Generator of Extended Sense Data control vector (reserved when Termination Procedure Origin Name field not present): 0 the termination procedure origin 1 a node other than the termination procedure origin

Extended Sense Data (X'35') Control Vector

Byte	Bit	Content
	5	Contents of Termination Procedure Origin Name field (reserved when Termination Procedure Origin Name field not present): 0 termination procedure origin name 1 name of node other than termination procedure origin, as described below; termination procedure origin name not known
	6−7	Reserved
7		Length 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 Extended Sense Data control vector.) *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)
m+2−n		Termination procedure origin name: if the field contains the termination procedure origin name (see byte 6, bit 5), network-qualified name of the node that caused the session termination procedure to be executed; otherwise, the network-qualified name of the node that generated the Extended Sense Data control vector, with, when available, a local or network name that indicates the direction from which the RU signaling the termination procedure was received *Note1:* 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 CP to begin session termination, the T4|T5 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]). *Note2:* The network identifier is always included in the termination procedure origin name.
n+1−p		Related resource (If the length in byte n+1 is 0, the Related Resource field may be omitted.)
n+1		Length of Related Resource Name field (values 0 to 17 are valid)
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.

Fully-qualified PCID (X'60') Control Vector

Fully-qualified PCID (X'60') Control Vector

Byte	Bit	Content
0−1		Vector header; Key=X'60' (see "Substructure Encoding/Parsing Rules" on page 8-2)
2−n		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 present)

Session Key

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

MS Major Vectors and Unique Subvectors

Introduction

The following table shows, by key value, the MS major vectors that an NMVT or CP-MSU can carry.

Key	MS Major Vector	Applicable Message-Unit Structures
X'0000'	Alert	NMVT
X'0050'	Change Control	CP-MSU
X'0061'	Reply to Execute Command	NMVT
X'0062'	Reply to Analyze Status	NMVT
X'0063'	Reply to Query Resource Data	NMVT
X'0064'	Reply to Test Resource	NMVT
X'0066'	Reply Activation Acceptance	CP-MSU
X'006F'	Send Message To Operator	NMVT
X'0080'	RTM	NMVT
X'0090'	Reply Product Set ID	NMVT
X'8050'	Request Change Control	CP-MSU
X'8061'	Execute Command	NMVT
X'8062'	Analyze Status	NMVT
X'8063'	Query Resource Data	NMVT
X'8064'	Test Resource	NMVT
X'8066'	Request Activation	CP-MSU
X'8080'	Request RTM	NMVT
X'8090'	Request Product Set ID	NMVT

Note: The 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 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 8-165.
- Subvectors may appear in any order within a major vector unless otherwise stated.

MS Major Vector Formats

Alert (X'0000') MS Major Vector

PU → SSCP

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 8-165 for subvector keys X'00'−X'7F', and in "Alert MS Subvectors" on page 8-16 for subvector keys X'80'−X'FE'. *Note:* The following subvector keys may be used as indicated:

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

Key:

P	Present one time
CP(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.

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.
6. An instance of this subvector describing the PU 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 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 has preserved supporting data, e.g., a storage dump, to which the Alert must be correlated.
9. This subvector is present when the Alert reports an error on a LAN, and the node sending the Alert is attached to the LAN.
10. This subvector is present when the Alert reports a problem with a logical link using the SDLC or LAN LLC protocol.
11. This subvector is present when the Alert reports a problem with a logical link using the SDLC or LAN LLC protocol.
12. 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.
13. 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.
14. This subvector is present in an Alert if and only if none of the X'94', X'95', and X'96' subvectors is present.
15. 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

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

This subvector transports SDLC or LAN LLC link station failure information.

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

Byte	Bit	Content
0		Length (p+1), in binary, of the SDLC Link Station Data subvector
1		Key: X'8C'
2−p		Subfields containing SDLC 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 SDLC Control Field Received X'04' Last SDLC Control Field Sent X'05' Sequence Number Modulus X'06' Link Station State X'07' LLC Reply Timer Expiration Count X'08' Last Received N(R) Count

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

This subfield transports the outstanding frame count.

Outstanding Frame Count (X'02') SDLC 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 SDLC Control Field Received (X'03') SDLC Link Station Data Subfield

This subfield transports the last SDLC control field received from the secondary station before the error occurred.

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

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

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

This subfield transports the last SDLC control field sent to the secondary station before the error occurred.

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

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

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

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

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

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

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

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

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

This subfield transports the number of times the LLC Reply Timer (T1) expired.

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

Byte	Bit	Content
0		Length (q+1), in binary, of the LLC Reply Timer Expiration Count subfield
1		Key: X'07'
2−3		Count, in binary, of LLC Reply Timer (T1) expirations

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

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

Last Received N(R) Count (X'08') SDLC 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

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content
		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 "Special 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 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.

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content	
		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).
		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

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content	
		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
		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

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content
		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
		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.

Basic Alert (X'91') Alert MS Subvector

Byte	Bit	Content
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.
	3−15	Reserved
4		Alert type: a code point 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'10' retired

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

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' PROBLEM RESOLVED
		X'Bxxx' NOTIFICATION
		X'Cxxx' SECURITY
		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
		    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'1010' ADAPTER ERROR: A hardware error has occurred in an adapter, making it inoperable
		
		X'1100' INPUT DEVICE ERROR

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

Byte	Bit	Content		
			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'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'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

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

Byte	Bit	Content		
			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'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'1611'	IMPENDING STORAGE SUBSYSTEM FAILURE
			X'1612'	IMPENDING WORKSTATION SUBSYSTEM FAILURE
			X'1613'	IMPENDING COMMUNICATIONS SUBSYST 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 that resulted in all central processing units (CPUs) of a central processing complex (CPC) entering into a wait state with interrupts disabled.
		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 that originated in a computer program and are intended to direct the operation of a device are incorrect.
		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'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'3200'	LAN ERROR: An error has been detected on a local-area network	

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

Byte	Bit	Content	
		X'3203'	LOOP ERROR: An error has been detected on a communication loop
		X'3204'	LOOP OPEN
		X'3205'	LOOP ADAPTER INOPERATIVE
		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'3217'	OPTICAL FIBER CONVERTERS HAVE WRAPPED: An optical fiber subsystem has wrapped the primary path onto the secondary path due to an error condition on the primary path.
		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'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.

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

Byte	Bit	Content		
			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.
			X'3240'	TOKEN-BUS COMMUNICATIONS LOST
		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.	
			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'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
			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'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, or the AT Command Set) and DCE provided information about the link.	
			X'3601'	NO LPDA RESPONSE RECEIVED
			X'3602'	BAD FCS IN LPDA RESPONSE

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

Byte	Bit	Content		
			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
			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'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'4021'	EXCESSIVE STORAGE SUBSYSTEM ERRORS
			X'4022'	EXCESSIVE WORKSTATION SUBSYSTEM ERRORS
			X'4023'	EXCESSIVE COMMUNICATIONS SUBSYST ERRORS

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

Byte	Bit	Content		
		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
			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'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'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	

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

Byte	Bit	Content		
			X'8001'	CUSTOMIZATION IMAGE WARNING: A customization image parameter is incorrect and has been replaced by a valid value.
		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'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
			X'9206'	OUT OF TONER

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

Byte	Bit	Content		
			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'	PROBLEM RESOLVED: A problem has been resolved *Note:* The problem may have been reported earlier by an Alert.	
			X'A001'	IMPENDING COOLING PROBLEM RESOLVED: An impending cooling problem, reported earlier by an Alert, has been resolved without ever having impacted the availability of any resource
		X'B000'	OPERATOR NOTIFICATION: Problem-related information is being conveyed to a network operator *Note:* An X'Bxxx' code point is used only if no more-specific one is available.	
			X'B001'	MAINTENANCE PROCEDURE: A resource has been taken off-line 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 OFF-LINE
			X'B003'	LAN BRIDGE TAKEN OFF-LINE
			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 OFF-LINE
			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 off-line for maintenance.
		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
			X'C005'	UNKNOWN OSI MANAGEMENT SERVICES REQUEST: An OSI system is attempting to solicit management services from another system without being properly identified.

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

Byte	Bit	Content	
		X'C006'	INVALID PASSWORD
		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.
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')
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'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'0030' SYSTEM MICROCODE: The specific microcode was not identified.
- X'0031' SYSTEM STORAGE MICROCODE
  Note: See also code point X'0421' (STORAGE CONTROLLER MICROCODE)

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			X'0032'	SYSTEM DISPLAY MICROCODE *Note:* See also code point X'0422' (WORKSTATION CONTROLLER MICROCODE)
			X'0033'	SYSTEM COMMUNICATION MICROCODE *Note:* See also code point X'0423' (COMM SUBSYSTEM CONTROLLER MICROCODE)
			X'0034'	SYSTEM PRINTER MICROCODE *Note:* See also code point X'0422' (WORKSTATION CONTROLLER MICROCODE)
			X'0040'	INITIAL PROGRAM LOAD
		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'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'0220'	MAIN AC POWER SUPPLY
			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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
		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
		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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
		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'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).
			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'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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
		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, code-transparent, serial-by-bit information transfer over a link connection
			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: 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'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.

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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
			X'2031'	LINE: The telephone line or transmission link connecting two or more components in the network
			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'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'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'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'2100'	COMMUNICATIONS/REMOTE NODE: Either a communications facility denoted by an X'20xx' code point or a remote node denoted by an 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'	BSC COMMUNICATIONS/REMOTE NODE

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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'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: 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
		X'2600'	ELECTRICAL INTERFERENCE: An electrical disturbance in a communication system that interferes with or prevents reception of a signal or of information	

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
		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
			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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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'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
		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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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'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'3330'	ADAPTER HARDWARE: The hardware comprising an adapter
			X'3331'	ADAPTER MICROCODE: The microcode executing in an 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.)

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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
			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'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'3502'	TERMINAL MULTIPLEXER: The equipment used to connect multiple devices to a single cable

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content	
		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).
		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'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'3530'	ISDN NETWORK COMPONENT
		X'3531'	ISDN NETWORK TERMINATION (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
			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

Probable Causes (X'93') Alert MS Subvector

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'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. *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
			X'3703'	TOKEN-RING FAULT DOMAIN: An adapter, its nearest active upstream neighbor, and the token-ring media between them; the token-ring media consist 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 that converts electrical signals into optical signals and vice-versa
			X'3707'	TOKEN-RING LAN CABLES
			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.

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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'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'4000'	PERFORMANCE DEGRADED	
			X'4001'	STORAGE SUBSYSTEM OVERLOADED
			X'4002'	WORK STATION SUBSYSTEM OVERLOADED
			X'4003'	COMMUNICATIONS SUBSYSTEM OVERLOADED
		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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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	
			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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
		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 personal banking machine (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'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
			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

Probable Causes (X'93') Alert MS Subvector

Byte	Bit	Content		
			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'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'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'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	
		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

User Causes (X'94') Alert MS Subvector

Byte	Bit	Content
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 8-97 for keys X'80'−X'FE'. X'01' User Causes X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index *Note:* Subfields X'01' and X'81' are always present. One or more instances of the X'82' and/or X'83' subfields may be present, depending on the code points present in the X'01' and X'81' subfields.

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'

User Causes (X'01') User Causes Subfield

Byte	Bit	Content
2−q		2-byte user cause co de 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 "(sf82 qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield. The one or more necessary X'82' subfields follow immediately after 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.

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 immediately after 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 X'82' subfields that are associated with the code point, and whether a X'83' subfield is associated with it, as follows:

X'xx0x'−X'xx9x':	No X'82' subfields.
X'xxAx'−X'xxBx':	One X'82' subfield.
X'xxCx':	Two X'82' subfields.
X'xxDx':	Three X'82' subfields.
X'xxEx':	One X'83' subfield.
X'xxFx':	Reserved: code points will not be assigned in this range.

Defined user cause codes are:

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

- X'0111' THE PORTION OF MAIN STORAGE MADE AVAILABLE BY THE USER FOR A PARTICULAR OPERATION HAS BEEN EXHAUSTED

User Causes (X'01') User Causes Subfield

Byte	Bit	Content		
			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
			X'0214'	TERMINAL MULTIPLEXER 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'	(sf82 qualifier) LOCAL MODEM POWER OFF
		Note: The qualifier identifies the link segment level (LSL) on which the local modem belongs.		
			X'02A2'	(sf82 qualifier) REMOTE MODEM POWER OFF *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
			X'02A3'	(sf82 qualifier) LOCAL DSU/CSU POWER OFF *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'02A4'	(sf82 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'2304'	INCORRECT TELEPHONE NUMBER SPECIFIED
			X'2308'	CALLING DTE DOES NOT SUBSCRIBE TO THIS FACILITY: The calling DTE has requested a service that it does not subscribe to
			X'2310'	X.21 CONNECTION INTENTIONALLY CLEARED BY TERMINAL CONTROL UNIT OPERATOR
			X'23A0'	CONNECTION NOT ESTABLISHED: (sf82 qualifier) *Note:* The qualifier indicates the telephone number for the connection that could not be established.
		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'2500'	LINE NOT ENABLED: A communication link has not been prepared for data transmission	
			X'2501'	PORT DEACTIVATED
			X'2510'	LINE NOT ENABLED AT CALLED DTE
			X'2511'	PORT DEACTIVATED AT CALLED DTE
		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'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

User Causes (X'01') User Causes Subfield

Byte	Bit	Content		
			X'34A0'	CABLE NOT CONNECTED: (sf82 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: (sf82 qualifier) *Note:* The qualifier specifies the cable that was not installed.
			X'34A2'	CABLE UNPLUGGED: (sf82 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 INTERFACE
			X'3803'	TC LEAD ACTIVE ON OTHER REMOTE NODE INTERFACE
			X'38A1'	SPEED MISMATCH BETWEEN MODEMS ON (sf82 qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
			X'38A2'	SPEED MISMATCH BETWEEN DSU/CSU'S ON (sf82 qualifier) *Note:* The qualifier identifies the the link segment level (LSL) on which the DSU/CSUs belong.
			X'38A3'	(sf82 qualifier) LOCAL MODEM IN TEST MODE *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
			X'38A4'	(sf82 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'	(sf82 qualifier) REMOTE MODEM IN TEST MODE *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
			X'38A6'	(sf82 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'	(sf82 qualifier) LOCAL MODEM REINITIALIZED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
			X'38A8'	(sf82 qualifier) LOCAL DSU/CSU REINITIALIZED *Note:* The qualifier identifies the link segment level (LSL) on which the local DSI/CSU belongs.
			X'38A9'	(sf82 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'	(sf82 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'	(sf82 qualifier) LOCAL MODEM POWER OFF THEN ON *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.

User Causes (X'01') User Causes Subfield

Byte	Bit	Content		
			X'38AC'	(sf82 qualifier) REMOTE MODEM POWER OFF THEN ON *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
			X'38AD'	(sf82 qualifier) LOCAL DSU/CSU POWER OFF THEN ON *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
			X'38AE'	(sf82 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'	(sf82 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'	(sf82 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 degredation.
			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 degredation.
			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 degredation.
		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

User Causes (X'01') User Causes Subfield

Byte	Bit	Content		
			X'5204'	FILM JAM: There is a jam condition in the medium for a camera device.
		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 medium for a camera device has been exhausted.
			X'5305'	OUT OF BILLS OR DOCUMENTS: The medium 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 medium for a camera device is nearly exhausted.
			X'5505'	LOW ON BILLS OR DOCUMENTS: The medium 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
			X'6010'	DASD DEVICE NOT READY

User Causes (X'01') User Causes Subfield

Byte	Bit	Content		
			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'6020'	SERVICE DOOR OPEN: The door that provides access to the interior of the machine has been opened.
		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'7010'	CALL AUTHORIZATION REQUIRED
		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

User Causes (X'01') User Causes Subfield

Byte	Bit	Content		
			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.
			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'7120'	INCORRECT TEST TOOL USED: The test tool used for servicing the device is incorrect.
		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.
			X'73A0'	FILE FULL (sf82 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 (sf82 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'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

User Causes (X'01') User Causes Subfield

Byte	Bit	Content
		X'74A1' BLOCKED AIR FILTER (sf82 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'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'F8C0' FAILING COMPONENT IS IDENTIFIED BY (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 qualifier) (sf82 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'
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 8-97 for keys X'80'−X'FE'. X'01' Install Causes X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index *Note:* Subfields X'01' and X'81' are always present. One or more instances of the X'82' and/or X'83' subfields may be present, depending on the code points present in the X'01' and X'81' subfields.

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. 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).

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content
		The expression "(sf82 qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield. The one or more necessary X'82' subfields follow immediately after 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.
		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 immediately after 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 X'82' subfields that are associated with the code point, or whether a X'83' subfield is associated with it, as follows:
		X'xx0x'–X'xx9x': No X'82' subfields.
		X'xxAx'–X'xxBx': One X'82' subfield.
		X'xxCx': Two X'82' subfields.
		X'xxDx': Three X'82' subfields.
		X'xxEx': One X'83' subfield.
		X'xxFx': Reserved: code points will not be assigned in this range.
		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.
		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'.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			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'1300'	INCORRECT SOFTWARE GENERATION: A program has been installed incorrectly and the requested function cannot be performed	
			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'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
		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 (sf82 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

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			X'17C0'	THRESHOLD VALUE SET TOO LOW (sf82 qualifier) (sf82 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'2600'	SYSTEM OR TRANSMISSION MEDIA INSTALLED NEAR ELECTRICAL INTERFERENCE	
		X'3400'	CABLE INSTALLED INCORRECTLY: A cable has been incorrectly installed	
			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'34A0'	(sf82 subfield) LOCAL DCE INTERFACE CABLE NOT PROPERLY CONNECTED *Note:* The qualifier identifies the link segment level (LSL) on which the local DCE belongs.
			X'34A1'	(sf82 subfield) 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'	(sf82 qualifier) LOCAL MODEM EXTERNAL CLOCK NOT PROVIDED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
			X'35A1'	(sf82 qualifier) REMOTE MODEM EXTERNAL CLOCK NOT PROVIDED *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'35A2'	2-WIRE, HALF DUPLEX COUPLER ON THE (sf82 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.
			X'35A3'	(sf82 qualifier) MODEMS SPEED MISMATCH *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
			X'35A4'	(sf82 qualifier) DSU/CSU'S SPEED MISMATCH *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSUs belong.
			X'35A5'	(sf82 qualifier) INCOMPATIBLE MODEMS *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
			X'35A6'	SENDING NODE AND (sf82 qualifier) MODEMS CONFIGURATION MISMATCH *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
			X'35A7'	SENDING NODE AND (sf82 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'3704'	TOKEN-RING DUPLICATE STATION ADDRESS ASSIGNED
			X'3724'	CSMA/CD DUPLICATE STATION ADDRESS ASSIGNED
		X'3800'	LPDA CONFIGURATION ERROR: A configuration parameter for an LPDA link has been specified incorrectly	
			X'38A0'	(sf82 qualifier) LOCAL MODEM ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
			X'38A1'	(sf82 qualifier) LOCAL DSU/CSU ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
			X'38A2'	(sf82 qualifier) REMOTE MODEM ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
			X'38A3'	(sf82 qualifier) REMOTE DSU/CSU ADDRESS INCORRECT *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
			X'38A4'	(sf82 qualifier) LOCAL MODEM LPDA-2 DISABLED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
			X'38A5'	(sf82 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'	(sf82 qualifier) LOCAL MODEM NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content		
			X'38A7'	(sf82 qualifier) REMOTE MODEM NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
			X'38A8'	(sf82 qualifier) LOCAL DSU/CSU NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the local DSU/CSU belongs.
			X'38A9'	(sf82 qualifier) REMOTE DSU/CSU NOT CONFIGURED *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
			X'38AA'	(sf82 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'	(sf82 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'	(sf82 qualifier) LOCAL MODEM CONFIGURED AS CONTROL *Note:* The qualifier identifies the link segment level (LSL) on which the local modem belongs.
			X'38AD'	(sf82 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 (sf82 qualifier) AND (sf82 qualifier) *Note:* The qualifiers identify the link segment levels (LSL) where the speed mismatch is.
			X'38C1'	(sf82 qualifier) LOCAL MODEM HAS A 2-WIRE COUPLER AND THE (sf82 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'80C1'	STORAGE CONFIGURATION ERROR (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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.

Install Causes (X'01') Install Causes Subfield

Byte	Bit	Content
		X'80C3' REMOTE WORK STATION CONFIGURATION ERROR (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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'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'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'F8C0' FAILING COMPONENT IS IDENTIFIED BY (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 qualifier) (sf82 quqlifier) *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 8-97 for keys X'80'−X'FE'. X'01' Failure Causes X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index *Note:* Subfields X'01' and X'81' are always present. One or more instances of the X'82' and/or X'83' subfields may be present, depending on the code points present in the X'01' and X'81' subfields.

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'
2−q		2-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.

Failure Causes (X'01') Failure 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 "(sf82 qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield. The one or more necessary X'82' subfields follow immediately after 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.

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 immediately after 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 X'82' subfields that are associated with the code point, or whether a X'83' subfield is associated with it, as follows:

X'xx0x' — X'xx9x':	No X'82' subfields.
X'xxAx' — X'xxBx':	One X'82' subfield.
X'xxCx':	Two X'82' subfields.
X'xxDx':	Three X'82' subfields.
X'xxEx':	One X'83' subfield.
X'xxFx':	Reserved: code points will not be assigned in this range.

Defined failure cause codes are:

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.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			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'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' (LOADABLE STORAGE CONTROLLER MICROCODE)
			X'0032'	SYSTEM DISPLAY MICROCODE *Note:* See also code point X'0422' (LOADABLE WORK STATION CONTROLLER MICROCODE)
			X'0033'	SYSTEM COMMUNICATION MICROCODE *Note:* See also code point X'0423' (LOADABLE COMMUNICATIONS SUBSYSTEM CONTROLLER MICROCODE)
			X'0034'	SYSTEM PRINTER MICROCODE *Note:* See also code point X'0422' (LOADABLE WORK STATION CONTROLLER MICROCODE)
			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'0110'	STORAGE CONTROL: The component that controls access to storage

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			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'0220'	MAIN AC POWER SUPPLY
			X'02C0'	INTERNAL POWER UNIT FOR (sf82 qualifier) (sf82 qualifier) *Note:* The qualifiers identify the adapter numbers of the adapters served by the failing internal power unit.
			X'0230'	POWER DISTRIBUTION UNIT
			X'0231'	MOTOR GENERATOR
			X'0240'	INTERNAL CLOCK: A mechanism that 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

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			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
			X'0443'	COMMUNICATIONS SUBSYSTEM CONTROLLER INTERFACE: The interface between a communication subsystem controller and the main processor in its node
		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'1000'	SOFTWARE PROGRAM: A program implemented in software, as distinguished from one implemented in microcode	
			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.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'1012'	SOFTWARE DEVICE DRIVER: A program designed to control a device.
			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'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'10A1'	UNABLE TO BUILD ALERT REQUESTED BY (sf82 qualifier) *Note:* An Alert builder utility could not complete a request from the program identified by the qualifier.
			X'10E1'	SOFTWARE PROGRAM (sf83 product text)
		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
			X'200A'	ISDN COMMUNICATIONS ERROR

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		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'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'	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'20A0'	NO RESPONSE FROM THE X.21 NETWORK — (sf82 qualifier) EXPIRED *Note:* The qualifier specifies the X.21 timer that has expired.
		X'20A1'	NO RESPONSE FROM THE ISDN NETWORK — (sf82 qualifier) EXPIRED *Note:* The qualifier specifies the ISDN timer that has expired.
		X'20A2'	OSI PROTOCOL ERROR (sf82 qualifier) *Note:* The qualifier specifies the protocol code that defines the error condition which has occurred.
		X'20A3'	SNA COMMUNICATIONS ERROR (sf82 qualifier)
		X'20A4'	NO RESPONSE FROM THE X.25 NETWORK — (sf82 qualifier) EXPIRED *Note:* The qualifier specifies the X.25 timer that has expired.
		X'20A5'	NO RESPONSE FROM THE X.25 NETWORK — (sf82 qualifier) RETRY COUNT EXPIRED *Note:* The qualifier specifies the X.25 timer for which the retry count has expired.
		X'20A6'	(sf82 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'	(sf82 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.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'20A8'	(sf82 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'20B1'	X.25 COMMUNICATIONS ERROR — THE FOLLOWING DIAGNOSTIC PACKET WAS RECEIVED FROM THE NETWORK (sf82 qualifier) *Note:* The qualifier specifies the diagnostic code.
		X'20B2'	X.25 PROTOCOL VIOLATION DETECTED (sf82 qualifier) *Note:* The qualifier specifies the diagnostic code.
		X'20C1'	X.25 COMMUNICATIONS ERROR — THE FOLLOWING INDICATION PACKET WAS RECEIVED FROM THE NETWORK (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 qualifier) *Note:* The first qualifier specifies the packet type (reset, restart, or clear) and cause code and the second qualifier specifies the diagnostic code.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'20D1'	NO RESPONSE FROM THE X.25 NETWORK — (sf82 qualifier) EXPIRED (sf82 qualifier) (sf82 qualifier) *Note:* The first qualifier specifies the timer. The second qualifier specifies the retry count and the third qualifier specifies the timer setting.
		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.	
			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'22A0'	REMOTE NODE (sf82 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

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			X'23C1'	X.25 COMMUNICATIONS NOT ESTABLISHED — THE FOLLOWING INDICATION PACKET WAS RECEIVED FROM THE NETWORK (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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	
			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'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

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'32D1'	LOCAL DCE COMMUNICATIONS INTERFACE (sf82 qualifier) (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 qualifier) (sf82 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.
		X'32D3'	REMOTE DCE COMMUNICATIONS INTERFACE (sf82 qualifier) (sf82 qualifier) (sf82 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.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			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'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 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'33A0'	LINE ADAPTER MULTIPLEXER (sf82 qualifier): A line adapter (scanner) multiplexer in a communication controller *Note:* The qualifier identifies the line address for the failing multiplexer.
			X'33C0'	LINE ADAPTER (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 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.
			X'33C3'	LINE INTERFACE COUPLER (LIC) (sf82 qualifier) (sf82 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)

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			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'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'3480'	TWINAXIAL CABLE DISTRIBUTION ASSEMBLY
			X'34A0'	(sf82 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.
			X'34A1'	(sf82 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.	

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		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
		X'3507'	REMOTE DIGITAL DATA DEVICE: On a particular link segment, the digital data device (DDD) farther from the Alert sender
		X'3510'	CALLED DCE
		X'3511'	LINE: The telephone line or transmission link connecting two or more components in the network
		X'3512'	THE CONNECTION BETWEEN THE CALLING DCE AND ITS LOCAL DSE
		X'3513'	LOCAL LOOP ASSOCIATED WITH THE CALLED DTE
		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'3530'	ISDN NETWORK COMPONENT
		X'3531'	ISDN NETWORK TERMINATION (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

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		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'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'	(sf82 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'	(sf82 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'	(sf82 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'	(sf82 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.

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
		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
			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'	(sf82 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'	(sf82 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'	(sf82 qualifier) LOCAL MODEM FEATURE(S) *Note:* The qualifier identifies the link segment level (LSL) on which the modem belongs.
			X'36A3'	(sf82 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. *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

Failure Causes (X'01') Failure Causes Subfield

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'3706'	OPTICAL FIBER CONVERTER: A device which converts electrical signals into optical signals and vice-versa
			X'3707'	TOKEN-RING LAN CABLES
			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'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'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 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

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content			
		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	
			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	

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
			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'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'6330'	DISK DRIVE ELECTRONICS: The electronic components of a DASD device
			X'6350'	LOCAL CONSOLE
			X'6351'	REMOTE CONSOLE *Note:* "Local" and "remote" are defined with respect to the system with which the console communicates.
		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

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content		
		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'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'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
			X'F01A'	DM RECEIVED

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		X'F01B'	RECEIVE WINDOW SIZE EXCEEDED
		X'F01C'	LLC LEVEL CRC OR CHECK SUM 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'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
		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

Failure Causes (X'01') Failure Causes Subfield

Byte	Bit	Content	
		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
		X'F0A1'	ERROR OCCURRED READING FROM FILE (sf82 qualifier)
		X'F0A2'	ERROR OCCURRED WRITING TO FILE (sf82 qualifier)
		X'F0A3'	FAILURE OCCURRED ON (sf82 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.
		X'F0A4'	XID NEGOTIATION FAILED WITH (sf82 qualifier) *Note:* The qualifier specifies the SNA sense data identifying why the negotiation failed.
		X'F0A5'	COMPONENT OF (sf82 qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the equipment or service belongs.
		X'F0A6'	BAD LINE QUALITY ON (sf82 qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the line belongs.
		X'F0A7'	BOTH MODEMS DETECTED IMPULSE HITS ON (sf82 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 (sf82 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 (sf82 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 (sf82 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 (sf82 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'F0AC'	INCORRECT LPDA RESPONSE FROM THE LOCAL DSU/CSU ON (sf82 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 (sf82 qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the remote DSU/CSU belongs.
		X'F0AE'	BIPOLAR ERRORS DETECTED BY LOCAL DSU/CSU ON (sf82 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 (sf82 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 (sf82 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 (sf82 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 (sf82 qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
		X'F0B3'	MODEMS ON (sf82 qualifier) IN BACKUP SPEED *Note:* The qualifier identifies the link segment level (LSL) on which the modems belong.
		X'F0B4'	(sf82 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'	(sf82 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'	(sf82 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'	(sf82 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'	(sf82 qualifier) LOCAL 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'F0B9'	(sf82 qualifier) REMOTE DSU/CSU DETECTED DDS NETWORK LOOPBACK ACTIVE *Note:* The qualifier identifies the link segment level (LSL) on which the DSU/CSU belongs.
			X'F0BA'	INCORRECT LPDA RESPONSE FROM THE LOCAL MODEM ON (sf82 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 (sf82 qualifier) *Note:* The qualifier identifies the link segment level (LSL) on which the remote modem belongs.
		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'F8C0'	FAILING COMPONENT IS IDENTIFIED BY (sf82 qualifier) (sf82 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 (sf82 qualifier) (sf82 qualifier) (sf82 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	

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

Cause Undetermined (X'97') Alert MS Subvector

Byte	Bit	Content
1		Key: X'97'
2−p		One or more subfields containing recommended action data, as described in "Network Alert (X'0000') Common Subfields" on page 8-97. X'81' Recommended Actions X'82' Detailed Data X'83' Product Set ID Index *Note:* Subfields X'01' and X'81' are always present. One or more instances of the X'82' and/or X'83' subfields may be present, depending on the code points present in the X'81' subfield.

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 8-97 for keys X'80'-X'FE': X'01' Qualified Message Data X'82' Detailed Data *Note:* Any number of instances of the X'01' and X'82' subfields may be present. Each X'01' subfield contains a number indicating how many subsequent X'82' subfields are associated with it.

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'

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

Byte	Bit	Content
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 8-175 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 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)

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

Byte	Bit	Content
5		Qualifier count: a binary number indicating how many qualifiers are associated with this message *Note:* The qualifiers are specified in X'82' subfields following this X'01' subfield, and are substituted into the message in the order in which the X'82' 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

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 predefined 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).

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

Byte	Bit	Content

The expression "(sf82 qualifier)" in the English text indicates a variable-length gap, to be filled in with data passed in a Detailed Data (X'82') subfield. The one or more necessary X'82' subfields follow immediately after this subfield in the X'94' subvector, in the order in which they are to be associated with the gaps specified in the X'81' subfield.

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 immediately after 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 X'82' subfields that are associated with the code point, and whether a X'83' subfield is associated with it, as follows:

X'xx0x'—X'xx9x':	No X'82' subfields.
X'xxAx'—X'xxBx':	One X'82' subfield.
X'xxCx':	Two X'82' subfields.
X'xxDx':	Three X'82' subfields.
X'xxEx':	One X'83' subfield.
X'xxFx':	Reserved: code points will not be assigned in this range.

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
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

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

Byte	Bit	Content	
		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 INSTA
		X'0016'	REVIEW MODEM AND LINE STATUS DATA
		X'0017'	RUN VERIFY COMMAND
		X'00A0'	PERFORM TRANSMISSION LINE PROBLEM DETERMINATION PROCEDURES ON (sf82 qualifier) *Note:* The qualifier identifies the failing transmission line.
		X'00B0'	PERFORM PROBLEM DETERMINATION PROCEDURE FOR (sf82 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 (sf82 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 (sf82 qualifier) *Note:* The qualifier identifies the command, program, error recovery procedure, etc.
		X'00B3'	PERFORM (sf82 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 (sf82 qualifier) REMOTE MODEM AND (sf82 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 (sf82 qualifier) REMOTE DSU/CSU AND (sf82 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 (sf82 qualifier) REMOTE MODEM (FROM KEYPAD) and (sf82 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.

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

Byte	Bit	Content		
			X'00C3'	RUN SELF TEST WITH WRAP PLUG ON (sf82 qualifier) REMOTE DSU/CSU (FROM CONTROL PANEL) and (sf82 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 (sf82 qualifier) AND (sf82 qualifier) *Note:* The qualifiers identify the link segment level (LSL) on which the lines belong.
			X'00C5'	RUN DCE AND LINE STATUS TEST ON (sf82 qualifier) AND (sf82 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
			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'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'01C0'	CHECK FOR CONFIGURATION MISMATCH BETWEEN THE (sf82 qualifier) REMOTE MODEM AND (sf82 qualifier) LOCAL MODEM *Note:* The qualifiers identify the link segment level (LSL) on which the modems belong.

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

Byte	Bit	Content		
			X'01C1'	CHECK FOR CONFIGURATION MISMATCH BETWEEN THE (sf82 qualifier) REMOTE MODEM AND (sf82 qualifier) LOCAL DSU/CSU *Note:* The qualifiers identify the link segment level (LSL) on which the modem and the DSU/CSU This code point is used on a multi-segment link with modems on one segment and DSU/CSUs on the other segment.
		X'0200'	CHECK POWER: Check the electrical power supply for the device	
		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'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
		X'1000'	PERFORM PROBLEM RECOVERY PROCEDURES: Refer to the problem recovery documentation provided for this condition and follow the specified procedures	

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

Byte	Bit	Content		
			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'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
			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'10A1'	PERFORM (sf82 qualifier) *Note:* The qualifier specifies a problem recovery procedure to be performed.
			X'10A2'	FOLLOW PROBLEM RECOVERY PROCEDURE INDICATED AT PRINTER FOR (sf82 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 (sf82 qualifier) *Note:* The qualifier specifies a value that indexes a local problem recovery procedure.
			X'10A4'	FOR CORRECTIVE ACTION REFER TO (sf82 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'1100'	VARY OFFLINE	
		X'1200'	RETRY	

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

Byte	Bit	Content		
			X'1201'	MOVE THE PAGING DATA SETS TO ANOTHER SUB-SYSTEM
			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'12C0'	RETRY AFTER (sf82 qualifier) (sf82 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'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'13A0'	ACTIVATE ONE OR MORE PORTS IN THE ROTARY GROUP ASSOCIATED WITH (sf82 qualifier) *Note:* The qualifier identifies the telephone number associated with the rotary group.
			X'13A1'	ACTIVATE RESOURCES ATTACHED TO (sf82 qualifier)
			X'13A2'	DEACTIVATE RESOURCES ATTACHED TO (sf82 qualifier)
		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'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'14A0'	VARY OR CONNECT (sf82 qualifier) ON-LINE: Start the identified element via local target system control program facilities. *Note:* The qualifier identifies the target functional element.
			X'14A1'	IML the (sf82 qualifier): Initialize the central processor complex (CPC) or a resource within the CPC. *Note:* The qualifier identifies the IML target element.

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

Byte	Bit	Content		
			X'14A2'	ACTIVATE (sf82 qualifier): Issue the Activate command to attempt to bring the target system on-line. *Note:* The qualifier identifies the activation target.
			X'14D0'	IPL (sf82 qualifier) FROM (sf82 qualifier) WITH (sf82 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'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
			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'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

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

Byte	Bit	Content		
			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 (sf82 qualifier) *Note:* The qualifier identifies the card to be replaced, e.g., by its part number.
			X'18A1'	REPLACE CABLE IDENTIFIED BY (sf82 qualifier)
			X'18C0'	REPLACE THE BATTERY IDENTIFIED BY (sf82 qualifier) (sf82 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'19A0'	QUIESE AND MOVE THE (sf82 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 (sf82 qualifier) *Note:* The qualifier identifies the communication control unit (CCU) within the communication controller to which the fallback is to be done.
		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.

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

Byte	Bit	Content		
			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 those of the X'5x' 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'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'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'22C0'	REVIEW SENDING DEVICE LOG (sf82 qualifier) (sf82 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'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.
			X'3002'	CONTACT SECURITY CONTROL REPRESENTATIVE: Contact the customer representative who is responsible for dealing with security concerns for the device.

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

Byte	Bit	Content		
			X'30A0'	DIAL (sf82 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'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'3110'	CONTACT COMMUNICATIONS SYSTEMS PROGRAMMER
			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'3200'	REPORT THE FOLLOWING *Note:* Since replacement code points for reporting one, two, and three (sf82 qualifiers)'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 (sf82 qualifier)
			X'32C0'	REPORT THE FOLLOWING (sf82 qualifier) (sf82 qualifier)
			X'32D0'	REPORT THE FOLLOWING (sf82 qualifier) (sf82 qualifier) (sf82 qualifier)
		X'3300'	IF PROBLEM RECURS 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

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

Byte	Bit	Content		
			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'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
			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'	RECURRENCE OF SAME ERROR INDICATES MEDIA FAILURE
			X'F007'	RECURRENCE OF PROBLEM INDICATES DEVICE OR ATTACHMENT ERROR
			X'F008'	RECURRENCE INDICATES MEDIA PROBLEM
			X'F009'	NON RECURRENCE OF FAILURE INDICATES ORIGINAL DRIVE FAILURE
			X'F00A'	MULTIPLE FAILURES INDICATE LINE ADAPTER MULTIPLEXER FAILURE
			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'F050'	IPL CAPABILITIES LIMITED
			X'F051'	NO IPL CAPABILITIES
			X'F052'	NORMAL OPERATIONS CAN CONTINUE BUT IF AUXILIARY STORAGE IS EXHAUSTED ONSITE ACTION WILL BE NECESSARY

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

Byte	Bit	Content	
		X'F060'	TO RECOVER LOST RESOURCE
		X'F0A0'	FOR (sf82 qualifier)
		X'F0D0'	FAILING COMPONENT LOCATION (sf82 qualifier) (sf82 qualifier) (sf82 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'F0E0'	FOR (sf83 product text)
		X'F0E1'	PREPARE FOR AUTOMATIC SHUTDOWN OF (sf83 product text)
		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.

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'
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.

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content
	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 8-175 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'07' ERROR CODE X'09' EVENT CODE X'0A' LLC ERROR CODE 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

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'16'	BUS STATUS CODE
		X'17'	RING STATUS CODE
		X'18'	CALL PROGRESS SIGNAL: A notification from an X.21 network to a DTE, indicating why a connection could not be established
		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'20'	MESSAGE CODE
		X'21'	PANEL ERROR MESSAGE CODE
		X'22'	SYSTEM MESSAGE CODE
		X'23'	MESSAGE SEVERITY
		X'24'	WAIT STATE CODE
		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'3A'	RACK/UNIT
		X'3B'	RACK
		X'3C'	UNIT
		X'3D'	CARD SLOT
		X'3E'	CABLE POSITION
		X'40'	ERROR RECOVERY PROCEDURE
		X'41'	PDP CODE
		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

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content	
		X'57'	DTE ADDRESS CALLING
		X'58'	LOCAL DTE ADDRESS
		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'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'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
		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, and so forth.
		X'90'	YEAR/MONTH/DAY
		X'91'	TIME
		X'92'	JULIAN DATE
		X'93'	MINUTES
		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'B0'	EIA STANDARD
		X'B1'	CCITT STANDARD
		X'B6'	RETRY COUNT

Detailed Data (X'82') Network Alert Common Subfield

Byte	Bit	Content
		X'B7' TIMER SEETING
		X'B8' LINE SPEED (BITS PER SECOND)
		X'B9' LINE SPEED (KILOBITS PER SECOND)
		X'BA' LINE SPEED (MEGABITS PER SECOND)
		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
		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.
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'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). *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 in byte 4 *Note:* Maximum length of the detailed data is 44 bytes.

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'
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 8-175 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

Product Set ID Index (X'83') Network Alert Common Subfield

Byte	Bit	Content
	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.

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 8-165 for subvector keys X'00'–X'7F', and in "Request Change Control MS Subvectors" on page 8-116 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	
Install (X'81')	CP	Note 1
Remove (X'83')	CP	Note 2
Accept (X'85')	CP	Note 3
Corequisite Change (X'87')	On	Note 4

Key:

P Present one time
CP Conditionally present one time (See notes for conditions.)
On 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'

Install (X'81') Request Change Control MS Subvector

Byte	Bit	Content
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.

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 superseded 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

Post-Test (X'60') Install Subfield

Byte	Bit	Content
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'

Corequisite Change (X'87') Request Change Control MS Subvector

Byte	Bit	Content
2−p		An SNA/File Services file name, as defined by SNA/File Services within the registered GDS code point 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.

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 8-165 for subvector keys X'00'−X'7F', and in "Change Control MS Subvectors" on page 8-123 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	
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')	CPn	Note 4
Reporting Secondary Installation (X'8A')	CP	Note 5
Secondary Installation Change Name (X'8C')	CPn	Note 6
Reporting Back-Level Status (X'8E')	CP	Note 7
Back-Level Change Name (X'90')	CPn	Note 8
Reporting Deletion (X'92')	CP	Note 9
Deleted Change Name (X'94')	CPn	Note 10
Detailed Data (X'98')	On	

Key:

—	Not present
P	Present one time
CP	Conditionally present one time
CPn	Conditionally present more than one time (See notes for conditions.)
O	Optionally present one time
On	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'8E') 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'
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 X'30' not attempted and will not 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') 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 Status (X'30') Install Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Activation Use subfield
1		Key: X'30'

Activation Use Status (X'30') Install Subfield

Byte	Bit	Content
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'
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' not successful 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

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'
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

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

Reporting Removal (X'84') Change Control MS Subvector

Byte	Bit	Content
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' not successful
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

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' not successful

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'

Reported Change Name (X'88') Change Control MS Subvector

Byte	Bit	Content
2–p		An SNA/File Services file name, as defined by SNA/File Services within the registered GDS code point 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 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 registered GDS code point 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 Back-Level Status (X'8E') Change Control MS Subvector

The Reporting Back-Level Status 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 Status (X'8E') Change Control MS Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the Reporting Back-Level Status subvector
1		Key: X'8E'

Reporting Back-Level Status (X'8E') Change Control MS Subvector

Byte	Bit	Content
2−p		One subfield (listed by Key value below and described in detail following): X'10' Back-Level Status

Back-Level Status (X'10') Reporting Back-Level Status Subfield

This subfield reports the back-level status. It is always present once.

Back-Level Status (X'10') Reporting Back-Level Status Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Back-Level Status 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 registered GDS code point 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 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'
2−p		One subfield (listed by Key value below and described in detail following): X'10' Deletion Status

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 registered GDS code point 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)

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

This major vector requests that the message it contains 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 8-165 for subvector keys X'00'–X'7F'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in NMVT Execute Command (X'8061') Major Vector	
Name List (X'06')	P	
Self-Defining Text Msg (X'31')	P	

Key:

P Present one time

Reply to Execute Command (X'0061') MS Major Vector

PU → SSCP

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'

Reply to Execute Command (X'0061') MS Major Vector

Byte	Bit	Content
4−n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 8-165 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 NMVT Reply to Execute Command (X'0061') Major Vector	
Sense Data (X'7D')	CP	Note 1

Key:

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 MS parameter major vector follows in the NMVT.

Analyze Status (X'8062') MS Major Vector

SSCP → PU

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 8-165 for subvector keys X'00' - X'7F'. *Note:* The following subvector key is used as indicated:

Subvector	Presence in NMVT Analyze Status (X'8062') Major Vector	
Name List (X'06')	P	

Key:

P Present one time

Reply to Analyze Status (X'0062') MS Major Vector

PU → SSCP

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 8-165 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 NMVT Reply to Analyze Status (X'0062') Major Vector	
Sense Data (X'7D')	CP	Note 1

Key:

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 MS parameter major vectors follow in the NMVT.

Query Resource Data (X'8063') MS Major Vector

SSCP → PU

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 8-165 for subvector keys X'00'−X'7F'. *Note:* The following subvector key is used as indicated:

Subvector	Presence in NMVT Query Resource Data (X'8063') Major Vector	
Name List (X'06')	P	

Key:

P Present one time

Reply to Query Resource Data (X'0063') MS Major Vector

PU → SSCP

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 8-165 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 130B' major vectors.
Note: The following subvector keys may be used as indicated:

Subvector	Presence in NMVT Reply to Query Resource Data (X'0063') Major Vector	
Sense Data (X'7D')	CP	Note 1

Key:

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 MS parameter major vectors follow in the NMVT.

Test Resource (X'8064') MS Major Vector

SSCP → PU

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 8-165 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 NMVT Test Resource (X'8064') Major Vector	
Name List (X'06')	P	
Test Setup Data (X'80')	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

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'
4−n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 8-165 for subvector keys X'00'−X'7F', and in "Reply to Test Resource Subvectors" on page 8-140 for subvector keys X'80'−X'FF'.

Reply to Test Resource (X'0064') MS Major Vector

Byte	Bit	Content
		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 NMVT Reply to Test Resource (X'0064') Major Vector	
Sense Data (X'7D')	CP	Note 1
Test Result Data (X'81')	CP	Note 2

Key:

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 MS parameter major vectors follow in the NMVT.
2. This subvector is present only when the receiving application has executed the specified test.

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'

Test Result Data (X'81') Reply to Test Resource MS Subvector

Byte	Bit	Content
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

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'

Request Activation (X'8066') MS Major Vector

Byte	Bit	Content
4−n		MS subvectors, as described (using 0-origin indexing) in "MS Common Subvectors" on page 8-165 for subvector keys X'00'−X'7F', and in "Request Change Control MS Subvectors" on page 8-116 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	
Activate (X'81')	P	

Key:

P 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 if sessions are active. 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(=q)		Force indication: X'10' no (do not force) — reject if sessions are active X'20' yes (force) — activate even if sessions are active

Change Management Activation Use (X'20') Activate Subfield

This subfield indicates which installed changes to activate. It is optional.

Change Management Activation Use (X'20') Activate 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

Reply Activation Acceptance (X'0066') MS Major Vector

LU → LU

This major vector is used to reply to Request Activation (X'8066'), to indicate initial acceptance or rejection of the request.

Reply 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 8-165 for subvector keys X'00'−X'7F', and in "Change Control MS Subvectors" on page 8-123 for subvector keys X'80'−X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in Reply Activation Acceptance (X'0066') Major Vector	
	CP-MSU	
Date/Time (X'01')	P	
Activation Acceptance (X'82')	P	
Detailed Data (X'98')	On	

Key:

P Present one time
On Optionally present more than one time

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'

Attempt Status (X'10') Activation Acceptance Subfield

Byte	Bit	Content
2(=q)		Acceptance: X'10' will attempt X'20' will not attempt

Send Message to Operator (X'006F') MS Major Vector

PU → SSCP

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 8-165 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 NMVT Send Message to Operator (X'006F') Major Vector	
Name List (X'06')	P	

Key:

P Present one time

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 8-165 for subvector keys X'00'−X'7F', and in "Request Response Time Monitor Subvectors" on page 8-148 for subvector keys X'80'−X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in NMVT Request RTM (X'8080') Major Vector	
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 8-1.
	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 8-1.
	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

Request Type	Subvectors present in the Request RTM (X'8080') major vector	Bits B1	Bits B3
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

Figure 8-1. Setting of Bits 1 and 3 of Byte 2 of the RTM Request (X'92') Subvector

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
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

RTM Control (X'94') Request RTM MS Subvector

Byte	Bit	Content
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 8-165 for subvector keys X'00'−X'7F', and in "Response Time Monitor Subvectors" on page 8-151 for subvector keys X'80'−X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in NMVT RTM (X'0080') Major Vector	
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:

P Present one time
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
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

RTM Status Reply (X'91') RTM MS Subvector

Byte	Bit	Content
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'
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) 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.
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

RTM Data (X'93') RTM MS Subvector

Byte	Bit	Content
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 ($0 < r1 \leq b1 \times u$), ($b1 \times u < r2 \leq b2 \times u$), ... 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
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
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 8-165 for subvector keys X'00'−X'7F', and in "Request Product Set ID Subvectors" on page 8-155 for subvector keys X'80'−X'FE' *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in NMVT Request PSID (X'8090') Major Vector	
*Node Identification (X'81')	CP	Note 1
*Node and Port-Attached Devices Identification (X'83')	CP	Note 2

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.
2. This subvector is present when requesting the PSID for the node (PU) the major vector is sent to *and* its port-attached devices.

Request Product Set ID Subvectors

Node Identification (X'81') Request PSID MS Subvector

This subvector requests product identification for the node receiving the request.

Request Product Set ID (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.

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'

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.

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 8-165 for subvector keys X'00'−X'7F', and in "Reply Product Set ID Subvectors" on page 8-157 for subvector keys X'80'−X'FE' X'80' = X'FE'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in NMVT Reply PSID (X'0090') Major Vector	
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

Key:

P Present one time
CP Conditionally present one time (See Notes for conditions.)

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 Product Set ID 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 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'
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'

Power-on Since Last Solicitation (X'30') Port-Attached Device Config. Des. Subfield

Byte	Bit	Content
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

Text Data (X'1300') MS Parameter Major Vector

PU → SSCP

This MS parameter major vector accompanies one of two MS major vectors: 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 zero-origin indexing) in "MS Common Subvectors" on page 8-165 for subvector keys X'00'−X'7F'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in NMVT Message Data Parameters (X'1300') Major Vector	
Qualified Message (X'0A')	CP(n)	Note 1
Self-Defining Text Msg (X'31')	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

PU → SSCP

This MS parameter major vector accompanies one of five MS major vectors: 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) in "MS Common Subvectors" on page 8-165 for subvector keys X'00'−X'7F'. *Note:* The following subvector keys may be used as indicated:

Subvector	Presence in NMVT Structured Data (X'1307') Major Vector	
Hierarchy/Resource List (X'05')	P	
Resource Data (X'80')	CP(n)	

Key:

P Present one time
CP(n) Conditionally present one or more times. (See Note for conditions.)

Note:

1. This subvector is present one or more times when the Structured Data MS parameter major vector follows the Reply To Test Resource (X'0064') MS major vector or the Reply To Query Resource Data (X'0063') MS major vector.

Structured Data Subvectors

Resource Data (X'80') Structured Data MS Subvector

This Structured Parameter 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		Subfields containing a resource name and data related to the resource. The X'01' 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

Resource Item Name (X'01') Resource Item Name Subfield

This subfield transports the name of the resource data item, i.e. a label. One of the following may be used: Resource Item Name, Resource Item Address, or Resource Item ID.

Resource Item Name (X'01') Resource Item Name 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

Transparent Coded Datastream(X'1309') MS Parameter Major Vector

PU → SSCP

This MS parameter major vector accompanies one of two MS major vectors: Reply to Execute Command (X'0061') and Send Message to Operator (X'006F'). It contains data whose structure is not defined by SNA, but that 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

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 NMVT Begin Data Parameters (X'130A') Major Vector	
Resource State (X'82')	CP	Note 1
Probable Causes (X'93')	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 (3), 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

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'

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 the PU for time-stamping the NMVT 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

Hierarchy Name List (X'03') MS Common Subvector

Byte	Bit	Content	
		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
		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'

SNA Address List (X'04') MS Common Subvector

Byte	Bit	Content
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)
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 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		The following subfield containing a hierarchical list of resources (listed by key value below and described in detail following): X'10' Hierarchy Name List

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		Always set to X'80'
3−q		Hierarchy Name List Entries (transmission order of entries indicates resources down the hierarchy, i.e., first-sent is highest in 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.

Hierarchy Name List (X'10') Hierarchy/Resource List Subfield

Byte	Bit	Content
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. 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.
	2−7	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 name X'19' program product X'20' storage device X'21' adapter X'25' diskette X'27' loop X'29' keyboard 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

Hierarchy Name List (X'10') Hierarchy/Resource List Subfield

Byte	Bit	Content
		X'3F' port X'80' controller X'81' service point X'82' communication controller X'83' central processing unit 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

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'50' Destination Application Name

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
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.

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.

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

Qualified Message (X'0A') MS Common Subvector

Byte	Bit	Content
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'
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

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.

Sub-field	HW or SW X'11' (Note 1)	X'0000' Alert (Note 2) Sender	Resource	XID3 (Note 3)	FMH 7 (LU6.2)	
X'00'	HW	P	P	P	P	
X'01'	HW	CP	CP	CP	CP	Note 4
X'0B'	HW	O	O	O	O	
X'0E'	HW	O	O	O	O	
X'02'	SW	CP	CP	CP	CP	Note 5
X'04'	SW	CP	CP	CP	CP	Note 6
X'06'	SW	P	P	O	P	
X'07'	SW	—	CP	O	CP	Note 7
X'08'	SW	CP	CP	CP	CP	Note 6
X'09'	SW	—	CP	O	CP	Note 7

Conditions of Subfield Presence in Product Identifier Subvector

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

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. One of the X'07' and X'09' subfields is required in the software X'11' for software products modified by the customer.

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 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

Hardware Product Identifier (X'00') Product ID Subfield

Byte	Bit	Content
		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
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'		

Hardware Product Identifier (X'00') Product ID Subfield

Byte	Bit	Content
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'
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'
2−8(=r)		Program product number: seven characters *Note:* 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)

Software Product Customization Date and Time (X'09') Product ID Subfield

Byte	Bit	Content
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 sub-system 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.
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).

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 ID's are defined in Volume 2 of the *National Language Information and Design Guide*, SE09–8002. For example, the 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 ID's 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

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'

Relative Time (X'42') MS Common Subvector

Byte	Bit	Content
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

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'

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)	Note 2

Conditions of Subfield Presence in Supporting Data Correlation Subvector

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 retrieved by use of the Fully-qualified PCID present in the subfield.
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.

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. When it flows in an Alert, this correlator can then be used by the Alert receiver to retrieve session data from nodes in the session path via the Request Trace (X'8010') major vector—see SIR.

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 network-qualified CP name (values 3 to 17 are valid)
11−q		Network-qualified CP name (NETID is not elided)

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 8-97.

LAN Link Connection Subsystem Data (X'51') Supporting Data Correlation Subfield

This MS common subvector transports data on the elements of the LAN link connection.

LAN Link Connection Subsystem Data (X'51') Supporting Data Correlation Subfield

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') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection Subsystem Data Subfield

This subfield transports the routing information used by a link.

LAN Routing Information (X'05') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection Subsystem Data Subfield

This subfield specifies the type of beacon detected by the LAN adapter.

Beaconing Data (X'07') LAN Link Connection 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') LAN Link Connection Subsystem Data Subfield

This subfield transports the address of the MAC element associated with the failure.

Single MAC Address (X'08') LAN Link Connection Subsystem Data Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Single MAC Address subfield

Single MAC Address (X'08') LAN Link Connection Subsystem Data Subfield

Byte	Bit	Content
1		Key: X'08'
2−7(=q)		Single MAC address, in hexadecimal

Fault Domain Error Weight Pair (X'09') LAN Link Connection 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') LAN Link Connection 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') LAN Link Connection Subsystem Data Subfield

This subfield transports the bridge identifier of a LAN bridge.

Bridge Identifier (X'0A') LAN Link Connection Subsystem Data Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Bridge Identifier subfield
1		Key: X'0A'
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') LAN Link Connection Subsystem Data Subfield
This subfield transports the name of the MAC element within the sending node.

Local Individual MAC Name (X'23') LAN Link Connection 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: a string of no more than 16 upper-case alphanumeric EBCDIC 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') LAN Link Connection 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') LAN Link Connection Subsystem Data Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Remote Individual MAC Name subfield
1		Key: X'24'
2−q		Remote individual MAC name: a string of no more than 16 upper-case alphanumeric EBCDIC 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') LAN Link Connection 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') LAN Link Connection 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: a string of no more than 16 upper-case alphanumeric EBCDIC charac 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') LAN Link Connection Subsystem Data Subfield

This subfield transports the name of the MAC related to the failure.

Single MAC Name (X'28') LAN Link Connection 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: a string of no more than 16 upper-case alphanumeric EBCDIC characters 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

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(=q)		Local link station DLC address, in hexadecimal; e.g., for a LAN, the source service access point (SSAP) address

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
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
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
5(=q)		Point-to-point or multipoint: X'01' point-to-point X'02' multipoint

LPDA Fault LSL Descriptor (X'08') Link Connection Subsystem 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 (X'08') Link Connection Subsystem 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

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 9, "Sense Data"

Chapter 9. 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 9-1) 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.

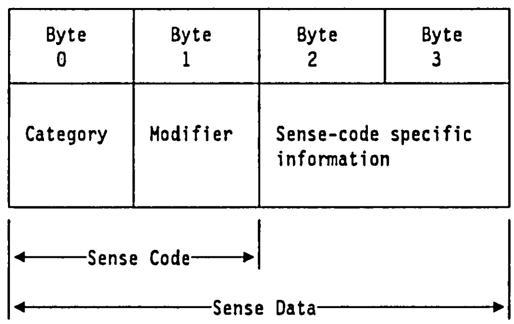


Figure 9-1. Sense Data Format

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.
0003	Name aliasing cannot be performed because the name alias function is not available.
0006	The line is not associated with a line adapater.
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.

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.

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.

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.

0007 The LU address in bytes 8−9 of RNAA type X'4' is already in the free pool.

000A The configuration identifier specified in a management services command is not recognized by the DLC manager at the receiving node.

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.

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.

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.

000E The resource to be dynamically reconfigured (DRed) was defined at system-definition time and is defined as not DR-deletable.

000F-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 control vector (X'15') 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.
0030	An FNA was received for an LU that has an active SSCP-LU 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 LU was not pending receipt of 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 as 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 A cross-domain resource, which was expected to be active, is inactive.

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 gateway 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 gateway resources; the receiving SSCP should attempt to reroute the request to another SSCP.

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.
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 from a change management focal point specifying use of changes installed in production only, but the receiver supports such a request only when it is received locally.
0010	Reserved
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 request that is not valid from an SSCP that is not in the native network of the gateway T4 node.
0003	The link station received a CONTACT from an unauthorized SSCP.
0004	A BFCLEANUP was received from an unauthorized SSCP.
0005	An RNAA was received from an unauthorized SSCP.

080F End User Not Authorized: The requesting end user 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.

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, UNBIND, or in negative response to BIND.

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.

0002 More PUs or LUs are requested by RNAA than the attachment resource will hold.

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.

0005 The Requested Reserved Resources for Sessions Are Not Available: In RNAA, a reservation of session resources exceeded those available; no address was assigned and no change was made to the LU's current reservation.

0007 Insufficient resources are available for LU address allocation.

000B A BFSESSINFO was received for an unknown LU.

000D Insufficient buffers exist to activate a session.

0011 Insufficient storage is available to the SNA component to satisfy the request at this time.

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.

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.

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	Online terminal test in progress.
0007	SDLC link test, level 2, in progress.
0009	Test initiated from the modem panel is 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.

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.

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 Station/SSCP ID: The station ID or SSCP ID in the request was found to be invalid.

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.

0001 No Session Found: The session identified in the BFCLEANUP was not found; the BFCLEANUP is rejected.

0002 The session identified in the BFCINIT was not found; the BFCINIT is rejected.

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 Reserved

0820 Control Vector Error: Invalid data for the control vector specified by the target network address and key.

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.

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 required cryptography is specified, but no SLU cryptographic data key is provided.

0822 Link Procedure Failure: A link-level procedure has failed due to link equipment failure, loss of contact with a link station, or an invalid response to a link command. (This is not a path error, since the request being rejected was delivered to its destination.)

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.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 For LU 6.2, Backout Initiated: 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. This sense data is sent only in FMH-7.

For non-LU 6.2, no specific code applies.

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.

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.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Invalid Request: The specified link station has already received a CONTACT and is therefore under the control of another SSCP. This CONTACT would exceed the share limit (=1).

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, due to 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. They can, however, be accepted if desired (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. The only possibility is to perform data object renewal using Send-and-Install with removability prohibited or desired—but not required (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).
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).
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 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).

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.

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.

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 Reserved

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 named 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:

0000 No specific code applies.

0003 Invalid Link: The link to which the PU is to be added is not an SNA link. Only SNA links are supported.

0004 Invalid Link: A request that is allowed only for a nonswitched link was received for a link that is defined to the receiver as switched.

0005 Resource Not Dynamically Added: This request works only with resources that were added through dynamic reconfiguration.

0009 RNAA(Move) was received for a resource that was added through dynamic reconfiguration; such a resource may not be moved through RNAA(Move).

0010 A SETCV with control vector X'43' was received for a non-switched resource.

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.

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.

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.

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.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

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 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 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 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 (0-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.

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.

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.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 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.

0001 A second BIND has been received from a peripheral node PLU while the session was still in the activation process.

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

0855 Reserved

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.

0858 Reserved

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 customer-defined server that is sensitive to data object contents, has detected an exception.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

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.

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 The 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.

085F Reserved

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.

0003 The CDINIT request or response contains a Session Initiation 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.

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.

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 Subvector Missing: One or more MS subvectors that are required by the receiver to perform some function are missing from the received list of subvectors, 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 subvector key (nn) of one of the 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: An 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 subvector key (nn) of the subvector 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 an MS major vector 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.)

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 orderly deactivation of the receiving node has been received, but sessions are active and their implied deactivation is not allowed; the requested activation will not proceed.

0002 An MS major vector specifying deactivation of the receiving node has been received, but the receiver cannot determine if sessions are active; the requested activation 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.

000D The sending SSCP and the receiving T4 node have conflicting system definitions. A BIND has been received for an LU address that is currently being used by an active LU-LU session. The LU address is primary on this active session. The LU address cannot be used for a secondary role on a new session.

000E The sending SSCP and the receiving T4 node have conflicting system definitions. A BIND has been received for an independent LU, but the LU specified is not in a T2.1 node.

000F The sending SSCP and the receiving T4 node have conflicting system definitions. The SSCP owner is the same as the SSCP sending the nonactivation CONTACT PIU, but the node to be contacted is not a T2.1. The CONTACT is for a T2.1 node, but the node to be contacted is not defined as a T2.1 to the receiver.

0010 The BFCLEANUP is for an independent LU, but the LU specified is not an independent LU.

0011 The subarea address portion of an addressed LU is not equal to the subarea address of the T4 node. The LU is not in the same subarea as the T4 node.

0012 A BFCLEANUP is for a resource that is not a BF LU, and hence the request is rejected. This is a situation where the function is not supported by the target resource. It can be caused by a system definition mismatch between the T4 node and the SSCP.

0013 The network ID in the BIND SLU name is not equal to the network ID of the boundary function, or the SLU name is not equal to the LU name in the boundary function control block for the LU.

0014 The LU specified in the FNA is not associated with the PU specified in the FNA; that is, an LU address (bytes 7−n) is not associated with the PU target address specified.

0015 BFCINIT Name Mismatch: The BIND cannot be built from the BFCINIT because the network-qualified PLU name does not match. The session activation is rejected by the boundary function with a BFTERM.

0016 Invalid Target Address: Either of the following conditions holds:

- The PU with which the specified LUs are to be associated is not type 1 or type 2; i.e., the SSCP attempts to add an LU to a PU, but the boundary function has defined that PU as a type 4.
- The SSCP sent an RNAA assignment type X'0' or X'5' with a PU or LU specified instead of a link. This is caused by a system definition mismatch.

0017 An entire network address including subarea and element is required for Pre-ENA address assignment: If an entire network address is not specified and an RNAA requesting a pre-ENA address is received, the RNAA is rejected.

0018 An RNAA type 4 was received requesting an auxiliary address on a dependent LU.

001A The target LU specified in BFCLEANUP or BFCINIT is not associated with the same link station that is associated with the session indicated in the URC control vector.

001B The target link station specified in a BFCLEANUP is not the same link station as the session indicated in the URC control vector.

001C Resource Definition Mismatch for BFCINIT: The sending SSCP and the receiving T4 node have conflicting system definition. A BFCINIT has been received for an LU address that is currently being used by an active LU-LU session. The LU address is primary on this already active session. The LU address cannot be used for a secondary role on a new session.

001D The LU address in a BFCINIT is a secondary address; the BFCINIT is rejected.

001E The subject LU specified in a BFSESSINFO RU is not defined to the SSCP as an independent LU; this is a mismatch between the SSCP and the BF.

001F A dependent LU is attached to a PU that indicates ACTPU is to be suppressed; the SSCP cannot activate the LU because ACTLU is not supported.

0020 A peripheral node supporting independent LUs has received an ACTLU request for an LU. This request is rejected, as an independent LU does not support ACTLU.

0021 An RNAA(Add) was received by a boundary function for a resource defined at system definition time, which is not allowed.

0025 The receiving node is unable to process a BIND for the LU type specified for the given LU name.

0028 An RNAA(Move) was received for a link station, and the link station's primary-secondary role is incompatible with the target link. on the target link are defined with a different link station role (primary or secondary) than those of the source.

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.

002C BFSESSINFO received reporting a subject LU in another network.

002D BFSESSINFO received for an (independent) subject LU, but the reported LU is considered by the receiver as a dependent LU.

002E BFSESSINFO received reporting a dynamic subject LU that the receiver considers to be located under a different ALS than that reported in the BFSESSINFO. The SSCP will attempt to correct this configuration mismatch.

002F BFSESSINFO received reporting a subject LU that the receiver considers to be located under a different ALS than that reported in the BFSESSINFO. The SSCP cannot correct this configuration mismatch.

0030 BFSESSINFO received for a subject LU, but the receiver has the address associated with a different LU, which it considers to be static.

0031 BFSESSINFO 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 received for an LU. The subject LU 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 for an other-domain resource that is not active or an application program that is not active.

0034 An SSCP is unable to associate the information received in a BFSESSINFO 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 nonswitched 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.

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 non-recoverable 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: A gateway SSCP-PU session that is needed to establish an address transform for the intended cross-network LU-LU session was not active.

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. Th 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, no adjacent SSCP table corresponds 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 Reserved

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.)

087F Reserved

0880 Reserved

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.

0882 Reserved

0883 Reserved

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.

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 protocols.

088C Missing Control Vector: The RU did not contain a control vector that was expected to appear.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

nn00 A required control vector is missing. Byte 2 contains the key (nn) of the required control vector that is missing. If more than one control vector is missing, only the first omission is reported. The second byte of the sense code specific field is set to X'00'.

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.

0003 A SWITCH request specifying a switch to the already existing state was received.

0004 A SWITCH request was received that was invalid.

0005 The SLU has received SWITCH(Conditional, to backup) and no current XRF-backup sessions exist that can replace this session (that is, become the XRF-active.)

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 system definition mismatch between the OLU and the SSCP(OLU).
- 000F Invalid backup command.
- 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.

0890 Reserved

0891 Invalid Network ID (NETID)

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

- 0000 No specific code applies.
- 0001 PLU NETID Invalid: The NETID of the PLU is not the same as that of the SSCP(PLU).
- 0002 Invalid NETID: The NETID field in CONNOUT does not match the NETID defined in the link station receiving the CONNOUT.
- 0003 Invalid NETID: The NETID field in the RNAA is not the same as the native NETID. There is a mismatch between the system definitions of the SSCP and the type 4 node.

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 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 an SSCP—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 an SSCP—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 may contain the following sense code specific information:

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 contained a control vector that was in error.

xxyy The first byte (xx) of the sense code specific data contains the hex 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. The second byte (yy) of the sense code specific data contains the (zero-origin) byte offset of the error within the control vector.

0896 Control Vector Too Long.

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.

0001 The BFCLEANUP specifies that it is for an independent LU, but the LU specified is not an independent LU. This also could be caused by a resource mismatch.

0002 The target LU is not in the same subarea as the type 4 node.

0003 The function is not supported by the target resource.

0004 Invalid SLU Name: The network ID (if present) in the SLU Name field, is not equal to the network ID of the type 4 node, or the SLU name is not equal to the LU name contained in the T4 node system definition.

0005 The LU address specified in the FNA is not associated with the PU target address specified in the FNA.

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.

000A The PU of an independent PLU named in BFINIT does not have the same element address as the one in the ALS field of BFINIT.

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.

0010 An adjacent SSCP has the same SSCP name as the SSCP that controls the DLU, but a different network identifier from the DLU.

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'.)

0899 Invalid Address: An address 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.

0002 If the address requested in the RNAA is an existing address and an FNA has been received for this address, reject the RNAA.

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 control vector (X'5F') 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 control vector (X'5F') cannot be correlated to any previous session.

089C Reserved

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:

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.

0001 The LU is sending an UNBIND with a reason code of X'0A' (SSCP gone); the identified LU-LU session had to be deactivated because of a forced deactivation of the associated SSCP-PU or SSCP-LU session, for example, because of a DACTPU, DACTLU, or DISCONTACT.

0002 The LU is sending UNBIND with a reason code of X'0F' (cleanup).

0003 The gateway node is sending UNBIND with a reason code of 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, for example, a session setup error or session takedown failure had occurred.

08A2 Resource Active. The requested function must be performed on an inactive resource, and the resource is active.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 RNAA(MOVE) was received for an active resource.

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.

Request Error (Category Code = X'10')

This category indicates that the RU 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.

0005 An RNAA type 4 was received, in which the local address field length is greater than 1. The implementation does not support a length other than 1.

0006 An RNAA type 4 was received, in which the link station address field length is greater than 1. The implementation does not support a length other than 1.

0007 On BFCINIT, the network name portion of the network qualified name field has a format error.

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.

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 session-controller PLU.

0016 Service parameter not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report containing the one or more service parameter triplets that are 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 containing the one or more service parameter triplets that are 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 one or more unsupported functions. 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 one or more unsupported functions. 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 the SNA/DS 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. An LU cannot establish a session with itself.

0021 There is a mismatch between session initiation request type and the protocols (SSCP-independent or SSCP-dependent) used by the designated LU partner. For example, a session initiation request other than BFINIT identifies an independent LU as a session partner.

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 and 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.

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 indicated in UNBIND.

Bytes 2 and 3 following the sense code contain sense code specific information. Figure 9-2 on page 9-54 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.
6046	An SNA/DS transaction program is unable to allocate a conversation with an 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.

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

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 9-2. Usage of X'1008' Sense Code Specific Information by LU Type

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 is 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.

1011 RNAA Request Error: The RNAA is rejected because there is a mismatch between the sending and receiving nodes' system definitions, or capabilities.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 No Available Pre-ENA Addresses: An RNAA that requests an address that is pre-ENA compatible is rejected, as no pre-ENA addresses are available.

0002 RNAA Takeover Error: In a takeover situation, a system definition mismatch was detected between the SSCP currently controlling a resource and the SSCP taking over. For example, an RNAA will be rejected if the LU name in the RNAA is not the same as the LU name contained in the T4 node system definition; or an existing LU with the same local address is found, but the LU is system-defined (not dynamically added); or if the adjacent link station name given in the RNAA does not match the link station name provided in the T4 system definition.

0003 Invalid Network ID: If the network ID field in the RNAA is not the same as the native network ID of the receiving node, the RNAA is rejected.

0004 Invalid PU or LU Type: The RNAA is rejected if the PU to which the LUs are to be added is not type 1 or type 2, but instead was defined at the receiving PU as a type 4, or if the type of request is appropriate for a link station, but the resource specified in the request is a PU or an LU.

0005 Pre-ENA Address Cannot Be Assigned: An RNAA requesting a pre-ENA address assignment has been received and rejected because the system definition required for pre-ENA address assignment is missing.

1012 SNA/DS Receiver Exception MU Format Exception: Parsing or building of the SNA/DS Receiver Exception MU Format was unsuccessful.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

1013 Unknown Server Parameters: The specified parameters are not recognized by the server.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

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.

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.

2011 Pacing Protocol Error.

- 0000 A normal-flow request was received by a half-session after the pacing count had been reduced to 0 and before a pacing response had been sent.
- 0001 Unexpected ISOLATED PACING MESSAGE (IPM) Received: An IPM was received when the receiver was in a state that did not allow it.
- 0002 Unexpected Pacing Request Received: A request with the pacing indicator set was received when the receiver was in a state that did not allow it.

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.

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.
401A through 4020	Reserved
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.

Bytes 2 and 3 following the sense code contain sense code specific information. Settings allowed are:

0000 No specific code applies.

0001 Hierarchical Reset: The identified LU-LU session is being deactivated; an ACTLU/ACTPU(Cold) or DACTLU/DACTPU was received, or the PU has failed.

0003 Unrecoverable LU Failure: 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.

0004 Recoverable LU Failure: 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.

0005 Hierarchical Reset: Backup session reset resulted from a hierarchical reset.

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.

0007 Session not activated: A BIND was received for a dependent LU that has not received an ACTLU to activate the SSCP-LU session.

8006 Invalid FID: Invalid FID for the receiving node. (See Note 1 located at the end of this chapter).

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. (See Note 2 located at the end of this chapter).

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.

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 chapter).

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 COS Not Available: A session activation request cannot be satisfied because none of the virtual routes requested for the session is available.

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 session-activation 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 through 8016 Reserved

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.

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 one or more service parameter triplets for which a connection could not be found.

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.

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 10. Function Management (FM) Headers

The request header (RH) contains a format indicator (FI) that, when *on*, indicates that an FM header is at the beginning of the 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 10-1 and Figure 10-2 show the placement of FM headers within an RU:

RH: FMH, *BC,*EC	FM header	Data

Figure 10-1. FM Header Contained in One RU

RH: FMH, *BC,¬EC	First of FM header

RH: ¬FMH,¬BC,*EC	Rest of FM header	Data

Figure 10-2. FM Header Contained in Two Contiguous RUs of a Chain

Figure 10-3 shows some instances where FM headers are used and Figure 10-4 identifies the logical unit (LU) types that use each FM Header.

```
| TH | RH | FMH |

| TH | RH | FMH | Data |

| TH | RH | FMH | FMH | Data |  (see Note)

| TH | RH | FMH | GDS | Data | GDS | Data |
```

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 10-3. 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 10-4. 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

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: compression indicator (see Notes 4 and 5): 0 no compression 1 compression (the first byte following 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)

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.

FM Header 1

Byte	Bit	Content
		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): NNNNNNN 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*.)

Byte	Bit	Content
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 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 FMH and the Prime Compression Character FMH, can be sent as an FMH-2 or FMH-3. They are sent as an FMH-2 when they apply to a specific destination at the half-session and as 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 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	FMH concatenation (must be 0)
	1–7	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.
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

FM Header 4

Byte	Bit	Content
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)
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 LU.

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)

FM Header 5: Attach (LU 6.2)

Byte	Bit	Content
4	0	Security indicator: 0 user ID is not already verified 1 user ID is already verified
	1−3	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 10-10 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
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 10-10 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 (format described in Chapter 7, "User Data Structured Subfields")
w+1−w+6		Logical-unit-of-work instance number, in binary

FM Header 5: Attach (LU 6.2)

Byte	Bit	Content
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, or DB, 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.

PIP Variable

The PIP variable following FMH-5 Attach has the following format:

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

FM Header 5: Attach (Not LU 6.2)

Byte	Bit	Content
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.

FM Header 6

Byte	Bit	Content
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
2−5		SNA-defined sense data (see below)
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 (LU 6.2)

Byte	Bit	Content
Note:		Only the following sense data (in hexadecimal) can be sent in an LU 6.2 FMH-7. Sense data carried in non-LU 6.2 FMH-7 varies by implementation. See Chapter 9, "Sense Data" for additional details on sense data. The phrases following the sense data 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.

Sense Data	Return Code
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
080F6051	ALLOCATION_ERROR—SECURITY_NOT_VALID
08240000	BACKED_OUT
084B6031	ALLOCATION_ERROR—TRANS_PGM_NOT_AVAIL_RETRY
084C0000	ALLOCATION_ERROR—TRANS_PGM_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

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 9, "Sense Data" on page 9-1)
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 CD set *on* 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
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.

FM Header 12: Security

Byte	Bit	Content
1	0	Reserved
	1–7	Type: 0001100
2–9		Enciphered version of the random data received in RSP(BIND)

Chapter 11. Presentation Services (PS) Headers

Presentation Services (PS) Headers

Presentation services (PS) headers convey information between PS component sync point managers when the conversation using the session is allocated with the sync-point synchronization level. These headers are used only by LU type 6.2.

Transaction program data is delimited using a 2-byte length field called an LL, containing a value that is the number of bytes contained in the transaction program data plus 2 (the length of the LL field itself).

LL	transaction program data

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. Therefore, 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)
2−3		Sync point command type: X'0005' Prepare X'0006' Request Commit X'0007' Committed X'0008' Forget X'0009' Heuristic Mixed

PS Header 10: Sync Point Control

Byte	Bit	Content
4−5		Modifier specifying next flow (present only if bytes 2−3 = X'0005' or X'0006'; reserved when bytes 2−3 = X'0006' and 2-phase sync point being used): 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, Forget command type when Prepare was the first PS header received, and Committed command type when Request Commit was the first PS header received.

Chapter 12. GDS Variables for SNA Service Transaction Programs (STPs)

List of SNA Service Transaction Programs

Logical Unit type 6.2 service transaction programs are identified by a transaction program name (TPN) that begins with a value of X'06'. Other SNA service transaction programs are identified similarly. Figure 12-1 identifies the transaction program names that SNA currently defines. These TPNs are specified in an FM header type 5 (FMH-5 Attach).

TP Name	TP Description
X'06F1'	LU 6.2 Change number of sessions
X'06F2'	LU 6.2 SYNC POINT resynchronization
X'07F0F0F1'	DDM Synchronous Conversation
X'20F0F0F0'	DIA Process Destination TP
X'20F0F0F1'	DIA Server TP
X'20F0F0F2'	DIASTATUS TP
X'21F0F0F1'	DS_SEND TP (FS1)
X'21F0F0F2'	DS_RECEIVE TP (FS1)
X'21F0F0F3'	DS_ROUTER_DIRECTOR TP
X'21F0F0F6'	DS_General Server TP
X'21F0F0F7'	DS_SEND TP (FS2)
X'21F0F0F8'	DS_RECEIVE TP (FS2)
X'24F0F0F0'	FS server TP
X'30F0F0F2'	Object Distribution TP
X'30F0F0F3'	Object Distribution Server TP

Figure 12-1. SNA-Defined Service Transaction Programs

Refer to Chapter 13, "SNA/Distribution Services (DS)" for additional SNADS information and refer to Chapter 14, "GDS Variables for General Use" for information about GDS variables that are not specific to SNA service transaction programs.

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.
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

Change Number of Sessions (X'1210') GDS Variable

Byte	Bit	Content
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 = 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), 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 1001 reply, function completed normally
5		Sync point manager flags:
	0–6	Reserved
	7	Log status: 0 cold 1 warm
6		Length (values 1 to 17 are valid), in binary, of network-qualified LU network name
7–n		Network-qualified LU name (format described in Chapter 7, "User Data Structured Subfields")
n+1		Length (values 1 to 64 are valid), in binary, of log name
n+2–p		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 MS major vector, as described (using 0-origin indexing) in "MS Major Vector Formats" on page 8-13, and/or one or more of the following GDS variables if appropriate: X'1532' SNA Condition Report: documented in Chapter 15, "SNA/File Services (FS)." Present if an SNA-registered condition was recognized by the management services 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. *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, in binary, of Compare States GDS variable, including this Length field
2−3		GDS ID: X'1213'
4		Service flag:
	0−3	Reserved
	4−7	Request/reply indicator: 0010 request 1000 reply, function completed abnormally 1001 reply, function completed normally

Compare States (X'1213') GDS Variable

Byte	Bit	Content
5		Sync point manager 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		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 name (format described in Chapter 7, "User Data Structured Subfields")
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 0 to 8 are valid), in binary, of conversation correlator
n+2−q		Conversation correlator of transaction program that allocated the conversation that failed: see FMH-5 for format of this correlator
q+1		Length (values 2 to 8 are valid), of session instance identifier
q+2−p		Session instance identifier of session being used by conversation at time of failure (See Chapter 7, "User Data Structured Subfields" for the format of this identifier.)

Chapter 13. SNA/Distribution Services (DS)

Introduction

This appendix contains the format descriptions of the FS1 and 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 (DS) requires a byte-accurate description of the formats that must be understood by all DSUs. The DS 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 codepoint). In either case, the length bytes include the length of the header itself and the following information, if any. For FS1, a header may be either two bytes in length, referred to as an "LT," or five bytes in length, referred to as an "LLIDF" (length, GDS codepoint, and format byte).

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 nonatomic 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.

For FS1, segmentation is indicated by the contents of the F byte (the fifth byte of the LLIDF header). Structures may be segmented when the most significant bit of the F byte is *on*. If the most significant bit is *on*, then three more bytes, the ISS bytes, follow the LLIDF header. The ISS bytes indicate whether a particular segment is the last segment of a structure. In each segment except the last segment of a structure, the I byte contains X'20'. In the last segment of a structure, the I byte contains X'00'. The SS bytes contain X'0000'.

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 DS structures.

Structure Name

The first column of the header description table identifies DS 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 a DS message unit.

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 same notation is applied to the Format Set 1 encodings. Structures in FS1 are classified as either length-bounded LLIDFs (IDF), length-bounded LTs (T), delimited LLIDFs (Del-IDF), delimited LTs (Del-T), implied LLIDFs (Imp-IDF), or implied LTs (Imp-T).

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 that 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 DS 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.

Header Description Tables for FS2 Message Units

DISTRIBUTION TRANSPORT MESSAGE UNIT (DTMU)

Figure 13-1 (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	13-13	Del-ID	pfx	≥53*	1	Y	Y	4-12	—
Transport_Prefix	13-13	ID/pfx	1570	8-18	1	N	Y	1-3	—
Hop_Count	13-13	T	01	4	1	—	—	—	—
MU_ID	13-13	T	03	6	0-1*	—	—	—	—
MU_Instance_Number	13-13	T	06	4	0-1*	—	—	—	—
Transport_Command	13-13	ID/seg	1571	29-4096*	1	Y	Y	3-30	—
Dist_Flags	13-14	T	01	5	0-1	—	—	—	—
Service_Parms	13-15	T	02	5-32	0-1	—	—	—	—
Server_Obj_Byte_Count	13-18	T	03	10	0-1*	—	—	—	—
Origin_Agent	13-18	T	04	3-10	1	—	—	—	—
Server	13-18	T	05	3-10	0-1*	—	—	—	—
Origin_DSU	13-18	T	06	8-22	1	N	Y	2	—
Origin_RGN	13-19	T	01	3-10	1	—	—	—	—
Origin_REN	13-19	T	02	3-10	1	—	—	—	—
Origin_User	13-19	T	07	8-22	0-1	N	Y	2	—
Origin_DGN	13-19	T	01	3-10	1	—	—	—	—
Origin_DEN	13-20	T	02	3-10	1	—	—	—	—
Seqno_DTM	13-20	T	08	14-17*	1	—	—	—	—
Supplemental_Dist_Info1	13-21	T	09	3-10	0-1	—	—	—	—
Agent_Correl	13-22	T	0A	3-130	0-1	—	—	—	—
Report-To_DSU	13-22	T	0B	8-22	0-1	N	Y	2	—
Report-To_RGN	13-22	T	01	3-10	1	—	—	—	—
Report-To_REN	13-22	T	02	3-10	1	—	—	—	—
Report-To_User	13-23	T	0C	8-22	0-1	N	Y	2	—
Report-To_DGN	13-23	T	01	3-10	1	—	—	—	—
Report-To_DEN	13-23	T	02	3-10	1	—	—	—	—
Report_Service_Parms	13-24	T	0D	5-32	0-1	—	—	—	—
Report-To_Agent	13-26	T	0E	3-10	0-1	—	—	—	—
Dest_Agent	13-27	T	0F	3-10	0-1	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-3728	—	—	—	—	—
Dest_List	13-27	ID/seg	1572	12-11268	1	N	Y	1	—
Dest	13-27	Imp-T	idc	8-5654	≥1	N	Y	1-2	—
Dest_DSU	13-27	T/idc	01	8-22	1	N	Y	2	—
Dest_RGN	13-28	T	01	3-10	1	—	—	—	—
Dest_REN	13-28	T	02	3-10	1	—	—	—	—
Dest_User	13-28	T	02	8-22	≥0	N	Y	2	—
Dest_DGN	13-28	T	01	3-10	1	—	—	—	—

Figure 13-1 (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_DEN	13-29	T	02	3-10	1	—	—	—	—
Agent_Object	13-29	ID/seg	1573	5-32767	0-1	—	—	—	—
Server_Object	13-29	ID/seg	1574	≥5	0-1	—	—	—	—
Supplemental_Dist_Info2	13-29	ID/seg	1580	5-32767	0-1	—	—	—	—
Unrecognized_Reserve	13-48	ID/seg	—	4-32767	—	—	—	—	—
DS_Suffix	13-29	ID/sfx	157F	4	1	—	—	—	—
Note: * Refer to FS2 Structure Descriptions starting on page 13-13 for presence rules and length restrictions.									

DISTRIBUTION REPORT MESSAGE UNIT (DRMU)

Figure 13-2. 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	13-30	Del-ID	pfx	≥77*	1	Y	Y	6-12	—
Report_Prefix	13-30	ID/pfx	157C	8-18	1	N	Y	1-3	—
Hop_Count	13-13	T	01	4	1	—	—	—	—
MU_ID	13-13	T	03	6	0-1	—	—	—	—
MU_Instance_Number	13-13	T	06	4	0-1*	—	—	—	—
Report_Command	13-30	ID/seg	1575	25-4096*	1	Y	Y	3-20	—
Service_Parms	13-15	T	02	5-32	0-1	—	—	—	—
Report-To_Agent	13-26	T	04	3-10	1	—	—	—	—
Reporting_DSU	13-30	T	06	8-22	1	N	Y	2	—
Reporting_RGN	13-30	T	01	3-10	1	—	—	—	—
Reporting_REN	13-30	T	02	3-10	1	—	—	—	—
Report_DTM	13-31	T	09	10-13*	1	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-4015	—	—	—	—	—
Report-To_DSU_User	13-32	ID	1583	12-48	1	N	Y	1-2	—
Report-To_DSU	13-22	T	01	8-22	1	N	Y	2	—
Report-To_RGN	13-22	T	01	3-10	1	—	—	—	—
Report-To_REN	13-22	T	02	3-10	1	—	—	—	—
Report-To_User	13-23	T	02	8-22	0-1	N	Y	2	—
Report-To_DGN	13-23	T	01	3-10	1	—	—	—	—
Report-To_DEN	13-23	T	02	3-10	1	—	—	—	—
Report_Information	13-32	ID/seg	1576	18-4096	1	Y	Y	1-24	—
Reported-On_Origin_DSU	13-32	T	06	8-22	0-1*	N	Y	2	—
Reported-On_Origin_RGN	13-32	T	01	3-10	1	—	—	—	—
Reported-On_Origin_REN	13-32	T	02	3-10	1	—	—	—	—
Reported-On_Origin_User	13-33	T	07	8-22	0-1*	N	Y	2	—
Reported-On_Origin_DGN	13-33	T	01	3-10	1	—	—	—	—
Reported-On_Origin_DEN	13-33	T	02	3-10	1	—	—	—	—
Reported-On_Seqno_DTM	13-34	T	08	14-17	1	—	—	—	—
Reported-On_Supp_Dist_Info1	13-35	T	09	3-10	0-1*	—	—	—	—
Reported-On_Agent_Correl	13-36	T	0A	3-130	0-1	—	—	—	—
Reported-On_Origin_Agent	13-36	T	0B	3-10	0-1*	—	—	—	—
Reported-On_Dest_Agent	13-36	T	0C	3-10	0-1*	—	—	—	—
Receiving_DSU	13-44	T	10	8-22	0-1	N	Y	2	—
Receiving_RGN	13-44	T	01	3-10	1	—	—	—	—
Receiving_REN	13-45	T	02	3-10	1	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-3849	—	—	—	—	—
SNA_Condition_Report	13-37	ID/seg	1532	10-32767	1	Y	Y	1-10	13-10
Reported-On_Supp_Dist_Info2	13-36	ID/seg	1582	5-32767	0-1*	—	—	—	—
Unrecognized_Reserve	13-48	ID/seg	—	4-32767	—	—	—	—	—
DS_Suffix	13-29	ID/sfx	157F	4	1	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 13-13 for presence rules and length restrictions.

DISTRIBUTION CONTINUATION MESSAGE UNIT (DCMU)

Figure 13-3. 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	13-37	Del-ID	pfx	≥18	1	Y	Y	2-10	—
Continuation_Prefix	13-37	ID/pfx	157B	14-24	1	N	Y	2-3	—
MU_ID	13-13	T	03	6	1	—	—	—	—
MU_Instance_Number	13-13	T	06	4	1	—	—	—	—
Restarting_Byte_Position	13-37	T	02	10	0-1	—	—	—	—
Agent_Object	13-29	ID/seg	1573	5-32767	0-1	—	—	—	—
Server_Object	13-29	ID/seg	1574	≥5	0-1	—	—	—	—
Supplemental_Dist_Info2	13-21	ID/seg	1580	5-32767	0-1*	—	—	—	—
Unrecognized_Reserve	13-48	ID/seg	—	4-32767	—	—	—	—	—
DS_Suffix	13-29	ID/sfx	157F	4	1	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 13-13 for presence rules.

SNA CONDITION REPORT

Figure 13-4. SNA Condition Report

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
SNA_Condition_Report	13-37	ID	1532	10-32767	1	Y	Y	1-10	—
SNA_Report_Code	13-37	T	7D	6	1	—	—	—	—
Structure_Report	13-38	T	01	14-255	0-10*	Y	Y	2-10	—
Structure_State	13-38	T	01	3	1	—	—	—	—
Structure_Contents	13-38	T	02	3-100	0-1*	—	—	—	—
Parent_Spec	13-38	T	03	5-17	0-7	N	Y	1-4	—
Parent_ID_Or_T	13-39	T	01	3-4	1	—	—	—	—
Parent_Class	13-39	T	02	3	0-1*	—	—	—	—
Parent_Position	13-39	T	03	4	0-1	—	—	—	—
Parent_Instance	13-39	T	04	4	0-1	—	—	—	—
Structure_Spec	13-40	T	04	5-17	0-1*	N	Y	1-4	—
Structure_ID_Or_T	13-40	T	01	3-4	0-1*	—	—	—	—
Structure_Class	13-40	T	02	3	0-1*	—	—	—	—
Structure_Position	13-40	T	03	4	0-1	—	—	—	—
Structure_Instance	13-41	T	04	4	0-1	—	—	—	—
Structure_Segment_Number	13-41	T	05	4	0-1*	—	—	—	—
Structure_Byte_Offset	13-41	T	06	4	0-1	—	—	—	—
Sibling_List	13-41	T	07	3-100	0-1*	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-241	—	—	—	—	—
Reported-On_Dest_List	13-41	Del-T	pfx	12-11268	0-1*	N	Y	3	—
Reported-On_Dest_Prefix	13-41	T/pfx	08	2	1	—	—	—	—
Reported-On_Dest	13-42	Imp/T	idc	8-5654	≥1	N	Y	1-2	—
Reported-On_Dest_DSU	13-42	T/idc	09	2-22	1	N	Y	0-2	—
Reported-On_Dest_RGN	13-42	T	01	3-10	0-1*	—	—	—	—
Reported-On_Dest_REN	13-42	T	02	3-10	0-1*	—	—	—	—
Reported-On_Dest_User	13-42	T	0A	8-22	≥0	N	Y	2	—
Reported-On_Dest_DGN	13-43	T	01	3-10	1	—	—	—	—
Reported-On_Dest_DEN	13-43	T	02	3-10	1	—	—	—	—
Reported-On_Dest_Suffix	13-43	T	0B	2	1	—	—	—	—
Supplemental_Report	13-44	T	03	3-255	0-5*	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-17664	—	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 13-13 for presence rules and length restrictions.

SENDER EXCEPTION MESSAGE UNIT (SEMU)

Figure 13-5. 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	13-44	ID	1578	10-256	1	Y	Y	1-10	—
SNA_Report_Code	13-37	T	7D	6	1	—	—	—	—
MU_ID	13-13	T	03	6	0-1	—	—	—	—
MU_Instance_Number	13-13	T	06	4	0-1*	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-236	—	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 13-13 for presence rules.

RECEIVER EXCEPTION MESSAGE UNIT (REMU)

Figure 13-6. 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	13-44	Del-ID	pfx	≥25	1	Y	Y	2-10	—
Receiver_Exception_Command	13-44	ID/pfx	1577	15-512	1	Y	Y	2-8	—
Sender_Retry_Action	13-44	T	01	3	1	—	—	—	—
MU_ID	13-13	T	03	6	0-1	—	—	—	—
MU_Instance_Number	13-13	T	06	4	0-1*	—	—	—	—
Receiving_DSU	13-44	T	16	8-22	1	N	Y	2	—
Receiving_RGN	13-44	T	01	3-10	1	—	—	—	—
Receiving_REN	13-45	T	02	3-10	1	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-473	—	—	—	—	—
Unrecognized_Reserve	13-48	ID	—	≥4	—	—	—	—	—
SNA_Condition_Report	13-37	ID/sfx	1532	10-1024	1	Y	Y	1-10	13-10

Note: * Refer to FS2 Structure Descriptions starting on page 13-13 for presence rules.

COMPLETION QUERY MESSAGE UNIT (CQMU)

Figure 13-7. 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	13-45	ID	1579	14-256	1	Y	Y	2-10	—
MU_ID	13-13	T	03	6	1	—	—	—	—
MU_Instance_Number	13-13	T	06	4	1	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-242	—	—	—	—	—

COMPLETION REPORT MESSAGE UNIT (CRMU)

Figure 13-8. 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	13-45	ID	157A	7-256	1	Y	Y	1-10	—
Indicator_Flags	13-45	T	01	3	1	—	—	—	—
MU_ID	13-13	T	03	6	0-1	—	—	—	—
MU_Instance_Number	13-13	T	06	4	0-1*	—	—	—	—
Last_Structure_Received	13-46	T	04	4	0-1*	—	—	—	—
Last_Byte_Received	13-46	T	05	10	0-1*	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-225	—	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 13-13 for presence rules.

PURGE REPORT MESSAGE UNIT (PRMU)

Figure 13-9. 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	13-46	ID	157E	10-256	1	Y	Y	1-10	—
MU_ID	13-13	T	03	6	1	—	—	—	—
Unrecognized_Reserve	13-48	T	—	2-246	—	—	—	—	—

RESET REQUEST MESSAGE UNIT (RRMU)

Figure 13-10. 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	13-46	ID	1585	21-23	1	N	Y	2	—
MU_ID	13-13	T	03	6	1	—	—	—	—
Reset_DTM	13-46	T	09	11-13	1	—	—	—	—

RESET ACCEPTED MESSAGE UNIT (RAMU)

Figure 13-11. 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	13-47	ID	1586	21-23	1	N	Y	2	—
MU_ID	13-13	T	03	6	1	—	—	—	—
Reset_DTM	13-46	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 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 (1-origin)

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 (1-origin)

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 (1-origin)

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.

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 13-24.

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 are defined by or reserved for SNA/DS.
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. For FS2, the capacity triplet is not used in the DRMU. For FS1, the capacity triplet is used. For FS2, all service parameters are optional in both the DTMU and the DRMU. For FS1, the first three parameters are present in both Dist_MU *types* TRANSPORT and REPORT. 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.

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	one megabyte
X'16'	4MB	
X'18'	16MB	(default when Priority SP is LT X'E0')

Note: All other values are reserved.

1. In FS2, 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.

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; for FS1, the byte count need not be accurate.
Presence Rule:	Optional when the *server_object* is present; otherwise, precluded.
Format:	Unsigned binary integer (1-origin)

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:	01134-00500 (character set AR)
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.

Server

Description:	The *server* is the name to be used to store the *server_object* at the destination.
Presence Rule:	In FS2, optional when the *server_object* is present; otherwise, precluded. If optional and absent, the general server TP name is the default. In FS1, required when the *server_object* is present.
Format:	Character string, except for first byte

CGCSGID:	01134-00500 (character set AR)
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.

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: 01134-00500 (character set AR)

String Conventions: 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: 01134-00500 (character set AR)

String Conventions: 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.
Note:	For FS1, when the Dist_MU is of type REPORT and the distribution report was generated by DS, null user names will occur.
Format:	Character string

CGCSGIDs: 01134-00500 (base), 00930-00500 (enhanced character set)

String Conventions:

Base — Leading, imbedded, and trailing space (X'40') characters are not allowed.

ECS — 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 DS, null user names will occur.
Format:	Character string

CGCSGIDs: 01134-00500 (base), 00930-00500 (enhanced character set)

String Conventions:

Base	Leading, imbedded, and trailing space (X'40') characters are not allowed.
ECS	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. For FS2, the number ranges from 1 to (2**31)-1. For FS1, the number ranges from 0 (for report MUs) to 9999. 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 GMT for local time is included.
Note:	FS2 tolerates sequence numbers with value 0 in message units that had, at some point, come from an FS1 network and had already specified a sequence number of 0 (i.e., DIA application status reports). However, sequence numbers with value 0 are never originated from within an FS2 network.
Length Restriction:	Originating FS2 DSUs generate a GMT-based time. The minimum length for *seqno_DTM* is therefore 15 (1-origin).
Format:	Byte string

Byte	Content
0-1	LT header
	SEQNO
2-5	Signed binary integer limited 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)
	GMT FLAGS
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 Note: All other values are reserved.
15	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 23 are valid)
16	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (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, 12:00GMT 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 1-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00am on 2 January 1988 in New York would be 12:00GMT - 5 hours, or

```
X'07C401020C000000600500'
  yyyyMMddHHmmsshh- HHmm
```

Supplemental_Dist_Info1

Description:	The *supplemental_dist_info1* structure is reserved for future use.
Format:	Character string

Agent_Correl

Description:	The *agent_correlation* is a string supplied by the origin agent. 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*. For FS1, this information is valid only if Dist_MU is of 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. For FS1, this information is valid only if Dist_MU is of type TRANSPORT. This is typically, but not necessarily, the network ID.
Format:	Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	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. For FS1, this information is valid only if Dist_MU is of type TRANSPORT. This is typically, but not necessarily, the LU name.
Format:	Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Note:	If a product chooses to implement DGN=REN, the enhanced character set (ECS) subset is implemented in a particular network, and any DGN contains an ECS character that is not an element of character set AR, 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*. For FS1, this information is valid only if Dist_MU is of type TRANSPORT.

Report-To_DGN

Description:	The *report-to_DGN* is the first part of the user name to which distribution reports are to be sent. For FS1, this information is valid only if Dist_MU is of type TRANSPORT.
Format:	Character string

CGCSGIDs: 01134-00500 (base), 00930-00500 (enhanced character set)

String Conventions:

Base	Leading, imbedded, and trailing space (X'40') characters are not allowed.
ECS	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. For FS1, this information is valid only if Dist_MU is of type TRANSPORT.
Format:	Character string

CGCSGIDs: 01134-00500 (base), 00930-00500 (enhanced character set)

String Conventions:

Base	Leading, imbedded, and trailing space (X'40') characters are not allowed.
ECS	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.

For FS2, the comparison operators and values derived for the protection and security parameters are the same as those specified (explicitly or implicitly) in the DTMU. For FS1, 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 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 13-15.

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. For FS2, the capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified. For FS1, the capacity triplet is used. For FS2, all service parameters are optional in both the DTMU and the DRMU. For FS1, the first three parameters—priority, protection, and capacity—are present if report service parameters are to be specified.

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 (not present in FS2)

Byte	Content
0	X'03'
1	X'C0' REQUIRE_LEVEL_GE
2	X'00' ZERO

Notes: All other values are reserved.
Also, All FS1 implementations are able to receive distribution reports of FOUR_K capacity (X'0C'). New FS1 implementations always send distribution reports of ZERO capacity.

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 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: 01134-00500 (character set AR)

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.

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:	01134-00500 (character set AR)
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.

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:	01134-00500 (character set AR)
String Conventions:	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:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Note:	If a product chooses to implement DGN=REN, the enhanced character set (ECS) subset is implemented in a particular network, and any DGN contains an ECS character that is not an element of character set AR, 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

CGCSGIDs:	01134-00500 (base), 00930-00500 (enhanced character set)	
String Conventions:		
	Base	Leading, imbedded, and trailing space (X'40') characters are not allowed.
	ECS	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

CGCSGIDs: 01134-00500 (base), 00930-00500 (enhanced character set)

String Conventions:

Base	Leading, imbedded, and trailing space (X'40') characters are not allowed.
ECS	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*. For FS1, the *server_object* 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

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. 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: 01134-00500 (character set AR)

String Conventions: Leading, trailing, and imbedded 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: 01134-00500 (character set AR)

String Conventions: Leading, trailing, and imbedded 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 products support the offset from GMT for local time.
Length Restriction:	Originating FS2 DSUs always generate a GMT-based time. The minimum length for *report_DTM* is therefore 11 (1-origin).
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)
	GMT FLAGS
10	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 Note: All other values are reserved.
11	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 23 are valid)
12	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (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, 12:00GMT 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 1-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00am on 2 January 1988 in New York would be 12:00GMT - 5 hours, or

```
X'07C401020C000000600500'
  yyyyMMddHHmmsshh- HHmm
```

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:	01134-00500 (character set AR)
String Conventions:	Leading, trailing, and imbedded 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:	01134-00500 (character set AR)
String Conventions:	Leading, trailing, and imbedded 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

CGCSGIDs: 01134-00500 (base), 00930-00500 (enhanced char set)

String Conventions:

- Base: Leading, trailing, and imbedded space (X'40') characters are not allowed.
- ECS: Leading space (X'40') characters are disallowed, 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

CGCSGIDs: 01134-00500 (base), 00930-00500 (enhanced char set)

String Conventions:

- Base: Leading, trailing, and imbedded space (X'40') characters are not allowed.
- ECS: Leading space (X'40') characters are disallowed, trailing space characters are not significant, and imbedded space characters are significant.

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. For FS2, the number ranges from 1 to (2**31)-1. For FS1, the number ranges from 1 to 9999. The date-time is the date and time generated at the origin of the distribution. FS2 products support the offset from GMT for local time.
Length Restriction:	Originating FS2 DSUs always generate a GMT-based time. The minimum length for *reported-on_seqno_DTM* is 15 (1-origin).
Format:	Byte string

Byte	Content
0-1	LT header
	SEQNO
2-5	Signed binary integer limited 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)
	GMT FLAGS
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 Note: All other values are reserved.
15	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 23 are valid)
16	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (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, 12:00GMT 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 1-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00am on 2 January 1988 in New York would be 12:00GMT - 5 hours, or

```
X'07C401020C000000600500'
  yyyyMMddHHmmsshh- HHmm
```

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 subnetwork, the *reported-on_origin_agent* structure is discarded.

Format: Character string, except for first byte

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading, trailing, and imbedded 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.

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: 01134-00500 (character set AR)

String Conventions: Leading, trailing, and imbedded 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.

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 (1-origin)

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*.

SNA_Report_Code

Description:	The *SNA_report_code* is an SNA registered code identifying the condition that is being reported. Refer to Chapter 9, "Sense Data" on page 9-1 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_spec*s 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:	Presence governed by the *SNA_report_code*.

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 codepoints. 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 codepoints. 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:	Presence is governed by the *SNA_report_code*.
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:	Presence is governed by the *SNA_report_code*.

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_dest_users* with a *reported-on_dest_DSU* for those destinations specified in the original distribution request being reported on. For flat destination lists (i.e., lists containing only DSUs and/or DSU-user pairs), there are zero or one user names per DSU list. For factored destination lists, there can be multiple user names per DSU list.

Reported-On_Dest_DSU

Description:	The *reported-on_destination_DSU* is one of the original destination DSUs being reported on.

Reported-On_Dest_RGN

Description:	The *reported-on_destination_RGN* is the first part of the name of one of the original destination DSUs being reported on. This is typically, but not necessarily, the network ID.
Presence Rule:	Absent when passed through an FS1 subnetwork.
Format:	Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reported-On_Dest_REN

Description:	The *reported-on_destination_REN* is the second part of the name of one of the original destination DSUs being reported on. This is typically, but not necessarily, the LU name.
Presence Rule:	Absent when passed through an FS1 subnetwork.
Format:	Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Note:	If a product chooses to implement DGN=REN, the ECS subset is implemented in a particular network, and any DGN contains an ECS character that is not an element of Character Set AR, then ECS characters may occur in this structure.

Reported-On_Dest_User

Description:	The *reported-on_destination_user* is the name of one of the original destination users being reported on.

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 DS condition code of X'000D' (lost user names), user names will be null.
Format:	Character string

CGCSGID: 01134-00500 (base), 00930-00500 (enhanced character set)

String Conventions:

Base Leading, imbedded, and trailing space (X'40') characters are not allowed.

ECS Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

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 DS condition code of X'000D' (lost user names), user names will be null.
Format:	Character string

CGCSGID: 01134-00500 (base), 00930-00500 (enhanced character set)

String Conventions:

Base Leading, imbedded, and trailing space (X'40') characters are not allowed.

ECS Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Reported-On_Dest_Suffix

Description:	The *reported-on_destination_suffix* is the suffix of the *reported-on_destination_list*.

Supplemental_Report

Description:	The *supplemental_report* contains other information pertaining to a condition. The contents of the *supplemental_report* are governed by the *SNA_report_code.*
Presence Rule:	Presence is governed by the *SNA_report_code.*

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:	01134-00500 (character set AR)
String Conventions:	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:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Note:	If a product chooses to implement DGN=REN, the enhanced character set (ECS) subset is implemented in a particular network, and any DGN contains an ECS character that is not an element of SNA Character Set AR, 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 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:	*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:	*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. A byte count of X'FFFFFFFFFFFFFFFF' indicates that the structure was fully received. The byte count contains only atomic data and does not contain the segmenting LLs for segmented structures.
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 (1-origin)

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 always generates a GMT-based time. The minimum length for *reset_DTM* is 11 (1-origin).
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)
	GMT FLAGS
10	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 Note: All other values are reserved.
11	Hour offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (values from 0 to 23 are valid)
12	Minute offset from GMT, in binary, occurs when GMT flag ≠ X'E9' (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, 12:00GMT 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 1-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00am on 2 January 1988 in New York would be 12:00GMT - 5 hours, or

```
X'07C401020C000000600500'
  yyyyMMddHHmmsshh- HHmm
```

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 FS2 DSUs pass *unrecognized_reserve* structures through unchanged in outgoing DMUs.
Format:	Undefined byte string

Header Description Tables for FS1 Message Units

DISTRIBUTION MESSAGE UNIT (DIST_MU)

Figure 13-12 (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	13-53	Del-IDF	pfx	≥148	1	N	Y	3-4	—
Prefix	13-53	IDF/pfx	C00102	5-21	1	—	—	—	—
Dist_Command	13-53	IDF/seg	C10502	138-32511	1	N	Y	2-3	—
Service_Desc_Operands	13-53	Imp-IDF	idc	58-774	1	N	N	2-5	—
Dist_ID	13-53	IDF/idc	C34041	28-107	1	N	N	5-7	—
Origin_RGN	13-19	T	01	3-10	0-1	—	—	—	—
Origin_REN	13-19	T	02	3-10	1	—	—	—	—
Origin_DGN	13-19	T	03	2-10	1	—	—	—	—
Origin_DEN	13-20	T	04	2-10	1	—	—	—	—
Origin_Seqno	13-54	T	05	6	1	—	—	—	—
Origin_DTM	13-54	T	06	10	1	—	—	—	—
Agent_Correl	13-22	T	07	3-46	0-1	—	—	—	—
Dist_Gen_Options	13-54	IDF	C33D41	30-58	1	N	N	5	—
Dist_Flags (FS1)	13-55	T	01	3	1	—	—	—	—
Hop_Count	13-13	T	02	4	1	—	—	—	—
Service_Parms	13-15	T	03	11-32	1	—	—	—	—
Server_Object_Ind	13-55	T	04	4	1	—	—	—	—
Origin_Agent	13-18	T	05	3-10	1	—	—	—	—
Report-To_Address	13-55	IDF	C36041	14-45	0-1*	N	N	3-4	—
Report-To_RGN	13-22	T	01	3-10	0-1	—	—	—	—
Report-To_REN	13-22	T	02	3-10	1	—	—	—	—
Report-To_DGN	13-23	T	03	3-10	1	—	—	—	—
Report-To_DEN	13-23	T	04	3-10	1	—	—	—	—
Report-To_Options	13-56	IDF	C34341	8-47	0-1*	N	N	1-2	—
Report_Service_Parms	13-24	T	01	11-32	0-1	—	—	—	—
Report-To_Agent	13-26	T	02	3-10	0-1	—	—	—	—
Agent_Object	13-29	IDF	C32D01	6-517	0-1	—	—	—	—
Destination_Operands	13-56	Imp-IDF	idc	≥75	1	N	Y	3	—
Begin_Dest_Operands	13-57	IDF/idc	C35001	8	1	—	—	—	—
Dest_RGN_List	13-57	Imp-IDF	idc	≥62	≥1	N	Y	4	—
Dest_RGN	13-28	IDF/idc	C35201	5-13	1	—	—	—	—
Begin_REN_List	13-57	IDF	C35001	8	1	—	—	—	—
Dest_REN_List	13-57	Imp-IDF	idc	≥44	≥1	N	Y	4	—
Dest_REN	13-28	IDF/idc	C35301	6-13	1	—	—	—	—
Begin_DGN_List	13-57	IDF	C35001	8	1	—	—	—	—
Dest_DGN_List	13-58	Del-IDF	pfx	≥25	≥1	N	Y	4	—
Dest_DGN	13-28	IDF/pfx	C35401	6-13	1	—	—	—	—
Begin_DEN_List	13-58	IDF	C35001	8	1	—	—	—	—
Dest_DEN	13-29	IDF	C35501	6-13	≥1	—	—	—	—

Figure 13-12 (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
End_DEN_List	13-58	IDF/sfx	C35101	5	1	—	—	—	—
End_DGN_List	13-58	IDF	C35101	5	1	—	—	—	—
End_REN_List	13-58	IDF	C35101	5	1	—	—	—	—
End_Dest_Operands	13-58	IDF	C35101	5	1	—	—	—	—
Dist_Report_Operands	13-59	Imp-IDF	idc	≥63	0-1*	N	Y	2-4	13-51
Dist_Server_Operands	13-58	Imp-IDF	idc	≥14	0-1*	N	Y	2	—
Server_Prefix	13-58	IDF/idc	C90A41	8-280	1	N	N	1-3	—
Server_Obj_Byte_Count	13-18	T	01	10	0-1	—	—	—	—
Server	13-18	T	02	3-10	1	—	—	—	—
Server_Parms	13-58	T	03	3-255	0-1	—	—	—	—
Server_Object	13-29	IDF/seg	C90801	≥6*	1	—	—	—	—
DS_Suffix (FS1)	13-59	IDF	CF0100	5	1	—	—	—	—

Note:

- * Refer to FS1 Structure Descriptions starting on page 13-53 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 13-13. Distribution Report Operands

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children: Unrec	Children: Order	Children: Num	Children: Sub Table
Dist_Report_Operands	13-59	Imp-IDF	idc	≥63	0-1	N	Y	2-4	—
Report_Operands	13-59	Imp-IDF	idc	27-112	1	N	N	1-2	—
Report_Correlation	13-59	IDF/idc	C34041	27-87	1	N	N	4-5	—
Reported-On_Origin_DGN	13-33	T	03	3-10	1	—	—	—	—
Reported-On_Origin_DEN	13-33	T	04	3-10	1	—	—	—	—
Reported-On_Seqno	13-59	T	05	6	1	—	—	—	—
Reported-On_DTM	13-60	T	06	10	1	—	—	—	—
Reported-On_Agent_Correl	13-36	T	07	3-46	0-1	—	—	—	—
Receiving_DSU	13-44	IDF	C36141	8-25	0-1	N	N	1-2	—
Receiving_RGN	13-44	T	01	3-10	0-1	—	—	—	—
Receiving_REN	13-45	T	02	3-10	1	—	—	—	—
Gen_SNADS_Report	13-60	Imp-IDF	idc	16	0-1*	N	Y	2	—
Gen_SNADS_Type	13-60	IDF/idc	C35601	7	1	—	—	—	—
Gen_SNADS_Contents	13-61	IDF	C35741	9	1	N	Y	1	—
Gen_SNADS_Cond_Code	13-61	T	01	4	1	—	—	—	—
Gen_DIA_Report	13-61	Imp-IDF	idc	14-524	0-1*	N	Y	2	—
Gen_DIA_Type	13-62	IDF/idc	C35601	7	1	—	—	—	—
Gen_DIA_Contents	13-62	IDF	C35741	7-517*	1	—	—	—	—
Specific_Report	13-62	Imp-IDF	idc	≥36	1	N	Y	3	—
Begin_Report_DGN_List	13-62	IDF/idc	C35001	8	1	—	—	—	—
Report_DGN_List	13-62	Imp-IDF	idc	≥23	≥1	N	Y	4	—
Reported-On_Dest_DGN	13-43	IDF/idc	C35401	5-13	1	—	—	—	—
Begin_Report_DEN_List	13-62	IDF	C35001	8	1	—	—	—	—
Report_DEN_List	13-63	Imp-IDF	idc	5-553	≥1	N	Y	1-3	—
Reported-On_Dest_DEN	13-43	IDF/idc	C35501	5-13	1	—	—	—	—
Spec_SNADS_Report	13-63	Imp-IDF	idc	16	0-1*	N	Y	2	—
Spec_SNADS_Type	13-63	IDF/idc	C35601	7	1	—	—	—	—
Spec_SNADS_Cont	13-63	IDF	C35741	9	1	N	Y	1	—
Spec_SNADS_CC	13-64	T	01	4	1	—	—	—	—
Spec_DIA_Report	13-64	Imp-IDF	idc	14-524	0-1*	N	Y	2	—
Spec_DIA_Type	13-65	IDF/idc	C35601	7	1	—	—	—	—
Spec_DIA_Contents	13-65	IDF	C35741	7-517*	1	—	—	—	—
End_Report_DEN_List	13-65	IDF	C35101	5	1	—	—	—	—
End_Report_DGN_List	13-65	IDF	C35101	5	1	—	—	—	—

Note: * Refer to FS1 Structure Descriptions starting on page 13-53 for presence rules and length restrictions.

SENDER EXCEPTION MESSAGE UNIT (TYPE FS1)

Figure 13-14. 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)	13-66	IDF	CF0201	8	1	—	—	—	—

RECEIVER EXCEPTION MESSAGE UNIT (TYPE FS1)

Figure 13-15. 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)	13-44	Del-IDF	pfx	59-863	1	N	Y	3	—
Prefix	13-53	IDF/pfx	C00102	5	1	—	—	—	—
Receiver_Exception_Command	13-66	IDF	C10101	49-853	1	N	Y	2	—
Receiver_Exception_Correl	13-67	IDF	C32801	7-23	1	—	—	—	—
Exception_And_Reply_Data	13-67	Imp-IDF	idc	37-825	1	N	N	2	—
Receiver_Exception_Code	13-67	IDF/idc	C32201	8-255	1	—	—	—	—
Reply_Data	13-68	IDF	C34501	29-570	1	N	Y	2-3	—
Receiving_DSU	13-44	IDF	C36141	8-25	1	N	N	1-2	—
Receiving_RGN	13-44	T	01	3-10	0-1	—	—	—	—
Receiving_REN	13-45	T	02	3-10	1	—	—	—	—
SNADS_Report	13-69	Imp-IDF	idc	16	1	N	Y	2	—
SNADS_Report_Type	13-69	IDF/idc	C35601	7	1	—	—	—	—
SNADS_Report_Cont	13-69	IDF	C35741	9	1	N	Y	1	—
SNADS_Report_CC	13-69	T	01	4	1	—	—	—	—
DIA_Report	13-70	Imp-IDF	idc	14-524	0-1	N	Y	2	—
DIA_Report_Type	13-70	IDF/idc	C35601	7	1	—	—	—	—
DIA_Report_Cont	13-70	IDF	C35741	7-517	1	—	—	—	—
DS_Suffix (FS1)	13-59	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_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.
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)

Example:

The date-time encoding for 12:00 noon on 2 January 1988 is:

```
X'07C401020C000000'
  yyyyMMddHHmmsshh
```

Dist_Gen_Options

Description:	The *distribution_general_options* contains structures used by 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 DS is requested to generate a report in case of an exception. 1 A report will not be generated by 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.

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.

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_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.

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.

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.

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*.

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'

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_Parms

Description:	The *server_parameters* structure contains parameters passed by DS to the destination server. This structure is never sent, and is retired in FS2.
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_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.

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)

Example:

The date-time encoding for 12:00 noon on 2 January 1988 is:

```
X'07C401020C000000'
  yyyyMMddHHmmsshh
```

Gen_SNADS_Report

Description:	The *general_SNADS_report* contains the 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 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 DS condition is being reported.
Format:	Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' 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 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 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*.

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 DS and/or DIA report.

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 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 DS condition is being reported.
Format:	Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' 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 DS report for multiple destinations.
Presence Rule:	This information occurs when *spec_DIA_type* ≠ 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

Byte	Bit	Content
		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 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 DS exception that caused the Dist_MU to be rejected.

SNADS_Report_Type

Description:	The *SNADS_report_type* indicates that a DS exception is being reported.
Format:	Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' 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 DS condition in the Dist_MU.

SNADS_Report_CC

Description:	The *SNADS_report_condition_code* describes the particular 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'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 for IBM System 36 and System 38.
X'30F0F0F3'	Object Distribution server transaction program for IBM System 36 and System 38.

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
1532	SNA Condition Report
1570	Transport Prefix
1571	Transport Command
1572	Destination List
1573	Agent Object
1574	Server Object
1575	Report Command
1576	Report Information
1577	Receiver Exception Command
1578	Sender Exception Message Unit (type FS2)
1579	Completion Query Message Unit
157A	Completion Report Message Unit
157B	Continuation Prefix
157C	Report Prefix
157E	Purge Report Message Unit
157F	Suffix
1580	Supplemental Distribution Info2
1582	Reported-On Supplemental Distribution Info2
1583	Report-To DSU/User
1585	Reset Request Message Unit
1586	Reset Accepted Message Unit

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 DS, Prefix within DIST_MU or within REMU (type FS1)
C101*	in DIA, MU CMD NO REPLY ACKNOWLEDGE; in DS, Command within REMU (type FS1)
C105	Command, DIST_MU
C322*	in DIA, MU OPERAND IMM DATA EXCEPTION-CODE; in DS, Exception Code, within REMU (type FS1)
C328*	in DIA, MU OPERAND IMM DATA DATA CORRELATION; in DS, Correlation, within REMU (type FS1)
C32D*	in DIA, MU OPERAND IMM DATA USER-DATA; in DS, Agent Object within DIST_MU
C33D*	in DIA, MU OPERAND IMM DATA STATUS-INFORMATION; in DS, Distribution General Options, within DIST_MU
C340*	in DIA, MU OPERAND IMM DATA DISTRIBUTION-IDENTIFIER; in DS, Distribution Identifier, within DIST_MU
C343*	in DIA, MU OPERAND IMM DATA GENERAL-ROUTING-DATA; in DS, Report-To Options within DIST_MU
C345*	in DIA, MU OPERAND IMM DATA REPLY DATA; in 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
C361	Receiving DSU, within DIST_MU or within REMU (type FS1)
C908	Server Object, within DIST_MU
C90A	Server Prefix, within DIST_MU

[1] The asterisk following the ID indicates that that identifier is used by both DIA (Document Interchange Architecture) and DS.

CF01* in DIA, MU SUFFIX NORMAL-TERMINATION; in DS, Suffix within DIST_MU or within REMU (type FS1)

CF02* in DIA, MU SUFFIX ABNORMAL-TERMINATION; in DS, SEMU (type FS1)

Terminology Mappings

Figure 13-16 (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
Supplemental_Dist_Info1	—	—
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
Supplemental_Dist_Info2	—	—
DS_Suffix	DS_Suffix	Suffix
Dist_Report_MU	Dist_MU (type Report)	Dist_IU (type Status)

Figure 13-16 (Page 2 of 3). Terminology Mappings		
FS2 TERMINOLOGY	**Current FS1 TERMINOLOGY**	**Old FS1 TERMINOLOGY**
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_Supp_Dist_Info1	—	—
Reported-On_Supp_Dist_Info2	—	—
Reported-On_Agent_Correl	Reported-On_Agent_Correl	Orig_Correl
Reported-On_Dest_Agent	—	—
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	—	—
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	—	—

Figure 13-16 (Page 3 of 3). Terminology Mappings		
FS2 TERMINOLOGY	**Current FS1 TERMINOLOGY**	**Old FS1 TERMINOLOGY**
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 14. GDS Variables for General Use

The following chart indicates (using an "X") each GDS variable code point (with first byte = X'12') used by LU 6.2.

First hexadecimal digit
Second hexadecimal digit

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1	X	X		X												
2																
3																
4																
5																
6																
7																
8																
9																
A	X															
B																
C																
D																
E		X	X													
F		X	X	X	X	X										X

Figure 14-1. 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'1213'	Compare States (see Note 1)
X'12A0'	Workstation Display Passthrough
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 12, "GDS Variables for SNA Service Transaction Programs (STPs)" for the formats and meanings of these GDS variables.
2. See Chapter 10, "Function Management (FM) Headers" for the formats and meanings of these GDS variables.

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.

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
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

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
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

Chapter 15. SNA/File Services (FS)

This appendix 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 15-1. 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	15-12	ID	1549	27-128	1	Y	Y	2-8	—
U_of_W_Requester_DSU	15-12	T	01	8-22	1	N	Y	2	—
U_of_W_Requester_RGN	15-12	T	01	3-10	1	—	—	—	—
U_of_W_Requester_REN	15-12	T	02	3-10	1	—	—	—	—
U_of_W_Requester_User	15-12	T	03	8-22	0-1	N	Y	2	—
U_of_W_Requester_DGN	15-13	T	01	3-10	1	—	—	—	—
U_of_W_Requester_DEN	15-13	T	02	3-10	1	—	—	—	—
U_of_W_Requester_Agent	15-13	T	04	3-10	0-1*	—	—	—	—
U_of_W_Seqno_DTM	15-13	T	02	15-17	1	—	—	—	—
Unrecognized_Reserve	15-15	T	—	2-53	—	—	—	—	—

Note: * Refer to the structure description for presence rule(s).

SNA/FS Agent Request

Figure 15-2. 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	15-15	ID	1530	9-13321	1	N	Y	1-2	—
Command	15-15	ID	1546	5	1	—	—	—	—
Command_Parms	15-15	ID	1547	7-13312	0-1	Y	N	1-15	—
Source_Reporting_Action	15-16	T	02	3	0-1*	—	—	—	—
Target_Agent	15-16	T	03	3-10	0-1*	—	—	—	—
Target_Reporting_Action	15-16	T	04	3	0-1*	—	—	—	—
Report-To_Agent	15-17	T	05	3-10	0-1*	—	—	—	—
Report-To_DSU	15-17	T	07	8-22	0-1*	N	N	2	—
Report-To_RGN	15-18	T	08	3-10	1	—	—	—	—
Report-To_REN	15-18	T	09	3-10	1	—	—	—	—
Report-To_User	15-18	T	0A	8-22	0-1	N	N	2	—
Report-To_DGN	15-18	T	0B	3-10	1	—	—	—	—
Report-To_DEN	15-18	T	0C	3-10	1	—	—	—	—
Unrecognized_Reserve	15-15	T	—	2-13238	—	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

SNA/FS Server Request

Figure 15-3. 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	15-19	Del-ID	pfx	≥28	1	Y	Y	3-11	—
FS_Server_Request_Prefix	15-19	ID/pfx	1531	8-19	1	N	Y	1-3	—
Decoder_Instruction	15-19	T	01	4-5	0-1*	—	—	—	—
Source_Instruction	15-19	T	02	4-5	0-1*	—	—	—	—
Target_Instruction	15-19	T	03	4-5	0-1*	—	—	—	—
Data_Object_Group	15-20	Del-ID	pfx	≥16	1	N	Y	3-5	—
Group_Prefix	15-20	ID/pfx	1533	4	1	—	—	—	—
Supplemental_FS_Info1	15-21	ID	1534	4-1024	0-1	Y	Y	1-9	—
Unrecognized_Reserve	15-15	T	—	2-1020	—	—	—	—	—
Supplemental_FS_Info2	15-21	ID	1535	8-32767	0-1	Y	Y	1-15	—
Supplemental_FS_Info3	15-21	ID	153C	9-283	0-1	—	—	—	—
Supplemental_FS_Info4	15-21	ID	1550	12-2048	0-1	—	—	—	—
Unrecognized_Reserve	15-15	ID	—	4-30432	—	—	—	—	—
Data_Object	15-21	Del-ID	pfx	≥18	1	Y	Y	3-19	—
D_O_Prefix	15-21	ID/pfx	1536	4	1	—	—	—	—
D_O_Acceptance	15-21	ID	1537	10-1024	1	Y	Y	1-9	—
D_O_Class	15-21	T	81	6	1	—	—	—	—
Unrecognized_Reserve	15-15	T	—	2-1014	—	—	—	—	—
D_O_Global_Name	15-21	ID	1538	9-283	1	N	Y	1-7	15-9
Supplemental_FS_Info5	15-21	ID	1539	12-2048	0-1	—	—	—	—
D_O_Allocation_Info	15-21	ID	153F	14-1024	0-1*	Y	Y	1-7	15-9
D_O_Contents	15-22	ID/seg	1541	≥5	0-1*	—	—	—	—
Unrecognized_Reserve	15-15	ID	—	4-32767	—	—	—	—	—
D_O_Suffix	15-22	ID/sfx	1542	4	1	—	—	—	—
Group_Suffix	15-22	ID/sfx	1543	4	1	—	—	—	—
Unrecognized_Reserve	15-15	ID	—	4-32767	—	—	—	—	—
FS_Suffix	15-22	ID/sfx	154C	4	1	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

SNA/FS Agent Report

Figure 15-4. 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	15-22	ID	154A	14-32763	1	N	Y	2-3	—
Command	15-15	ID	1546	5	1	—	—	—	—
SNA_Condition_Report	15-28	ID	1532	10-32749	0-1*	Y	Y	1-10	15-11
FS_Action_Summary	15-22	ID	1548	5	1	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

SNA/FS Server Report

Figure 15-5. 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	15-23	Del-ID	pfx	≥22	1	N	Y	3-4	—
FS_Server_Report_Prefix	15-23	ID/pfx	154B	8-9	1	N	Y	1	—
Decoder_Instruction	15-19	T	01	4-5	1	—	—	—	—
SNA_Condition_Report	15-28	ID/seg	1532	10-32749	0-1*	Y	Y	1-10	15-11
Data_Object_Group	15-20	Del-ID	pfx	≥16	0-1*	N	Y	3-4	—
Group_Prefix	15-20	ID/pfx	1533	4	1	—	—	—	—
Supplemental_FS_Info2	15-21	ID	1535	8-32767	0-1	Y	Y	1-7	—
Supplemental_FS_Info3	15-21	ID	153C	9-360	0-1	—	—	—	—
Supplemental_FS_Info4	15-21	ID	1550	9-2045	0-1	—	—	—	—
Unrecognized_Reserve	15-15	ID	—	4-30358	—	—	—	—	—
Data_Object	15-21	Del-ID	pfx	≥8	1	Y	Y	2-13	—
D_O_Prefix	15-21	ID/pfx	1536	4	1	—	—	—	—
D_O_Global_Name	15-21	ID	1538	9-360	1	N	Y	1-8	15-9
Supplemental_FS_Info5	15-21	ID	1539	9-2045	0-1	—	—	—	—
Unrecognized_Reserve	15-15	ID	—	4-30354	—	—	—	—	—
D_O_Suffix	15-22	ID/sfx	1542	4	1	—	—	—	—
G_Suffix	15-22	ID/sfx	1543	4	1	—	—	—	—
FS_Suffix	15-22	ID/sfx	154C	4	1	—	—	—	—

Note: * Refer to the structure description for presence rule(s)

Subtables

Global Names

Figure 15-6. 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	15-23	T	01	3-12	0-1*	—	—	—	—
To_Be_Fetched_Name	15-24	T	02	5-77*	0-1*	N	Y	1-10	15-10
Fetching_Match_Flags	15-24	T	03	3-12	0-1*	—	—	—	—
To_Be_Stored_Name	15-25	T	04	5-77*	0-1*	N	Y	1-10	15-10
To_Be_Deleted_Name	15-25	T	05	5-77*	0-1*	N	Y	1-10	15-10
Deleting_Match_Flags	15-26	T	06	3-12	0-1*	—	—	—	—
Supplemental_FS_Info6	15-21	T	07	3-12	0-1*	—	—	—	—
Fetched_Name	15-26	T	08	5-77*	0-1*	N	Y	1-10	15-10
Stored_Name	15-26	T	09	5-77*	0-1*	N	Y	1-10	15-10
Deleted_Name	15-27	T	0A	5-77*	0-1*	N	Y	1-10	15-10
Reported-On_Name	15-27	T	0B	5-77*	0-1*	N	Y	1-10	15-10

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.

Allocation Information

Figure 15-7. 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	15-27	T	08	10	1	—	—	—	—
Unrecognized_Reserve	15-15	T	—	2-1010	—	—	—	—	—

Note: This subtable is referenced by the *FS_server_request*.

Tokens

Figure 15-8. 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	15-27	T	01	3-18	1	—	—	—	—
Second_Token	15-27	T	02	3-18	0-1	—	—	—	—
Third_Token	15-28	T	03	3-18	0-1	—	—	—	—
Fourth_Token	15-28	T	04	3-18	0-1	—	—	—	—
Fifth_Token	15-28	T	05	3-18	0-1	—	—	—	—
Sixth_Token	15-28	T	06	3-18	0-1	—	—	—	—
Seventh_Token	15-28	T	07	3-18	0-1	—	—	—	—
Eighth_Token	15-28	T	08	3-18	0-1	—	—	—	—
Ninth_Token	15-28	T	09	3-18	0-1	—	—	—	—
Tenth_Token	15-28	T	0A	3-18	0-1	—	—	—	—
Note: This subtable is referenced by the *global_names* table.									

SNA Condition Report

Figure 15-9. The SNA/FS Use of the SNA_Condition_Report

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
SNA_Condition_Report	15-28	ID	1532	10-32749	0-1*	Y	Y	1-10	—
SNA_Report_Code	15-28	T	7D	6	1	—	—	—	—
Structure_Report	15-29	T	01	14-255	0-10*	Y	Y	2-10	—
Structure_State	15-29	T	01	3	1	—	—	—	—
Structure_Contents	15-29	T	02	3-100	0-1*	—	—	—	—
Parent_Spec	15-29	T	03	5-17	0-7	N	Y	1-4	—
Parent_ID_Or_T	15-30	T	01	3-4	1	—	—	—	—
Parent_Class	15-30	T	02	3	0-1*	—	—	—	—
Parent_Position	15-30	T	03	4	0-1	—	—	—	—
Parent_Instance	15-30	T	04	4	0-1	—	—	—	—
Structure_Spec	15-31	T	04	5-17	0-1*	N	Y	1-4	—
Structure_ID_Or_T	15-31	T	01	3-4	0-1*	—	—	—	—
Structure_Class	15-31	T	02	3	0-1*	—	—	—	—
Structure_Position	15-31	T	03	4	0-1	—	—	—	—
Structure_Instance	15-32	T	04	4	0-1	—	—	—	—
Structure_Segment_Number	15-32	T	05	4	0-1*	—	—	—	—
Structure_Byte_Offset	15-32	T	06	4	0-1	—	—	—	—
Sibling_List	15-32	T	07	3-100	0-1*	—	—	—	—
Unrecognized_Reserve	15-15	T	—	2-241	—	—	—	—	—
Reported-On_Dest_List	15-32	Del-T	pfx	12-11268	0-1*	N	Y	3	—
Reported-On_Dest_Prefix	15-32	T/pfx	08	2	1	—	—	—	—
Reported-On_Dest	15-33	Imp/T	idc	8-5654	≥1	N	Y	1-2	—
Reported-On_Dest_DSU	15-33	T/idc	09	2-22	1	N	Y	0-2	—
Reported-On_Dest_RGN	15-33	T	01	3-10	0-1*	—	—	—	—
Reported-On_Dest_REN	15-33	T	02	3-10	0-1*	—	—	—	—
Reported-On_Dest_User	15-33	T	0A	8-22	≥0	N	Y	2	—
Reported-On_Dest_DGN	15-33	T	01	3-10	1	—	—	—	—
Reported-On_Dest_DEN	15-34	T	02	3-10	1	—	—	—	—
Reported-On_Dest_Suffix	15-34	T	0B	2	1	—	—	—	—
Reported-On_Token_String	15-34	T	02	5-182	0-10*	N	Y	1-10	15-10
Supplemental_Report	15-34	T	03	3-255	0-5*	—	—	—	—
Unrecognized_Reserve	15-15	T	—	2-15826	—	—	—	—	—

Note: * Refer to the structure description for presence rule(s) and length restriction.

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 (Character Set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

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 (Character Set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

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 (Character Set AR)
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 (Character Set AR)
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 (Character Set AR)
String Convention:	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

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

Notes:

1. All other values are reserved.
2. 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

Note: All other values are reserved.

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 (Character Set AR)
String Convention:	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

Note: All other values are reserved.

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 (Character Set AR)
String Convention:	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 (Character Set AR)
String Conventions:	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 the SNA/FS reports are to be delivered. This is typically, but not necessarily, the LU name.
Format	Character string

CGCSGID:	01134-00500 (Character Set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

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 (Character Set AR)
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 (Character Set AR)
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 15-20.

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 15-20.

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 15-20.

Server Instructions:

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	source or target
		0010 SUMMARY_OR_EXCEPTIONS	decoder, source, or target
		0011 ONLY_IF_EXCEPTIONS	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	

Notes:

1. All other values are reserved.
2. 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_Acceptance

Description:	The *data_object_acceptance* contains 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 server to determine whether or not the request can be honored.
Format:	Byte string
Note:	Refer to "SNA/FS Data Object Classification Codes" on page 15-35 for the value descriptions.

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_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

Byte	Bit	Contents
0-3		LLID header
4	0-1	01 ALL_SUCCESSFUL (see Note 2) 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

Notes:

1. All other values are reserved.
2. 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 canonical 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 performing a create operation (e.g., CREATE_LOAD; CREATE_LOAD_OR_REPLACE), and the data object is to be involved in partial matching or token value generation.
Format:	Bit string

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
	10000	ORDERED, ≤16 CHARACTERS
	10001	ORDERED, ≤16 DECIMAL NUMERICS
	10010	ORDERED, DATE
	10011	ORDERED, TIME
	10100	ORDERED, G00V00

Notes:

1. All other values are reserved.
2. 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.

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 *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 Flag Values

Values

X'00'	FIND_A_MATCH
X'01'	IGNORE
X'02'	SELECT_HIGHEST
X'03'	SELECT_LOWEST

Note: All other values are reserved.

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 Flag Values

Values

X'00'	FIND_A_MATCH
X'01'	IGNORE
X'02'	SELECT_HIGHEST
X'03'	SELECT_LOWEST

Note: All other values are reserved.

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.

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)

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.

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 Chapter 9, "Sense Data" on page 9-1 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:	Presence governed by the *SNA_report_code*.

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: Presence is governed by the *SNA_report_code*.

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: Presence is governed by the *SNA_report_code*.

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_dest_user*s with a *reported-on_dest_DSU* for those destinations specified in the original distribution request being reported on. For flat destination lists (i.e., lists containing only DSUs and/or DSU-user pairs), there are zero or one user names per DSU list. For factored destination lists, there can be multiple user names per DSU list.

Reported-On_Dest_DSU

Description: The *reported-on_destination_DSU* is one of the original destination DSUs being reported on.

Reported-On_Dest_RGN

Description: The *reported-on_destination_RGN* is the first part of the name of one of the original destination DSUs being reported on. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reported-On_Dest_REN

Description: The *reported-on_destination_REN* is the second part of the name of one of the original destination DSUs being reported on. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reported-On_Dest_User

Description: The *reported-on_destination_user* is the name of one of the original destination users being reported on.

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.

Format: Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

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.
Format:	Character string

CGCSGID:	01134-00500 (character set AR)
String Conventions:	Leading, imbedded, and trailing space (X'40') characters are not allowed.

Reported-On_Dest_Suffix

Description:	The *reported-on_destination_suffix* is the suffix of the *reported-on_destination_list*.

Reported-On_Token_String

Description:	The *reported-on_token_string* contains the FS canonical identifier of a data object related to the detected condition.
Presence Rule:	Presence is governed by the *SNA_report_code*.

Supplemental_Report

Description:	The *supplemental_report* contains other information pertaining to a condition. The contents of the *supplemental_report* are governed by the *SNA_report_code*.
Presence Rule:	Presence is governed by the *SNA_report_code*.

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]	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
			Product Specific	10	10	60	Ex
	Microcode Customization	Unspecified	Unspecified	10	20	00	00
		Product Specific	Unspecified	10	20	Ex	00
Unspecified	Unspecified	Unspecified	Unspecified	00	00	00	00
			Product Specific	00	00	00	Ex
			Customer Specific	00	00	00	Fx

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 Info1
1535	Supplemental FS Info2
1536	Data Object Prefix
1537	Data Object Acceptance
1538	Data Object Global Name

[1] Microcode may be classified as IBM Licensed Internal Code. See "Special Notices" at the beginning of this document for more information.

1539	Supplemental FS Info5
153C	Supplemental FS Info3
153F	Data Object Allocation Info
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 Info4

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 first identifier tokens that have been registered by SNA/FS on behalf of SNA/FS-capable agents.

First Identifier	Agent
C'MCODE'	SNA/MS change management
C'MCUST'	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 the following fields:

- LU name
- Network-qualified LU name
- Mode 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 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.
- Symbol-string 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.

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, and USS 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.

For current and future SNA formats, two new character sets are used: character sets 00640 and 01134, both encoded using code page 00500. 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
80	Ø	O Slash, Capital			X			
81	a	a, Small		X				X
82	b	b, Small		X				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
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	ª	a Underscore, Small			X			
9B	º	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	Đ	D Stroke, Capital			X			
AD	Y	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
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

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
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	Ö	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 12, "GDS Variables for SNA Service Transaction Programs (STPs)," Chapter 14, "GDS Variables for General Use," Chapter 5, "Request/Response Units (RUs)," and Chapter 10, "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.

```
     ┌───────┬────────────┬─────────//────────┐
     │LENGTH │ IDENTIFIER │ INFORMATION ...   │
     │ (LL)  │    (ID)    │                   │
     └───────┴────────────┴─────────//────────┘
Byte  0       2            4                   N
```

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 a 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) encompassing 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.

Figure B-2 (Page 1 of 3). Identifier Registry

GDS ID	Structured Field Owner
00**	3270
01**	3270
03**	3270
06**	3270
09**	3270
0B**	3270
0C**	3270
0D**	3270
0E**	3270
0F00—0FFF	3270
1010—101F	3270
1030—1034	Print Job Restart

Figure B-2 (Page 2 of 3). Identifier Registry

GDS ID	Structured Field Owner
1100−1104	SNA Character String
1200−12FF	LU 6.2 (APPC)
1300−13FF	SNA/Management Services
1400−140F	3820 Page Printer
1570−158F	SNA/Distribution Services
40**	3270
41**	3270
4A**	3270
4B**	3270
4C**	3270
7100−71FF	3250
80**	3270
8100−81FF	3270
C000−C00F	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−C3FF	Document Interchange Architecture
C400−C46F	Document Interchange Architecture
C500−C56F	Document Interchange Architecture
C600−C66F	Document Interchange Architecture
C700−C7FF	Graphical Display Data Manager
C800−C87F	Document Interchange Architecture
C900−C97F	Document Interchange Architecture

Figure B-2 (Page 3 of 3). Identifier Registry

GDS ID	Structured Field Owner
C980—C9FF	Document Interchange Architecture
CA00—CA7F	Document Interchange Architecture
CA80—CAFF	Document Interchange Architecture
CB00—CB0F	Document Interchange Architecture
CC00—CC3F	Document Interchange Architecture
CD00—CD3F	Document Interchange Architecture
CF00—CF0F	Document Interchange Architecture
D000—D0FF	Distributed Data Management
D300—D3FF	Document Content Architecture
D600—D6FF	Intelligent Printer Data Stream
D780—D7BF	Facsimile Architecture
D820—D821	AS/400 (5250)
D930—D95F	AS/400 (5250)
E100—E10F	Level-3 Document Content Architecture
E200—E20F	Level-3 Document Content Architecture
E300—E30F	Level-3 Document Content Architecture
E400—E40F	Level-3 Document Content Architecture
E500—E50F	Level-3 Document Content Architecture
E600—E60F	Level-3 Document Content Architecture
E700—E70F	Level-3 Document Content Architecture
E800—E80F	Level-3 Document Content Architecture
E900—E90F	Level-3 Document Content Architecture
EA00—EA0F	Level-3 Document Content Architecture
EFFF	IBM Token-Ring Network PC Adapter
F000—FEFF	Non-IBM Reserved Block
FF00—FFFF	Context-Dependent Block

Appendix C. List of Abbreviations and Symbols

A

A	address (SDLC)
ACT	active, activate

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
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
ENA	extended network addressing
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)
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
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

D

E

F

G

H

I

L

Q

R

S

T

U

V

X

Reader's Comment Form

Systems Network Architecture
Formats

Publication No. GA27-3136-10

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: Copies of IBM Publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are: clarity, accuracy, completeness, organization, coding, retrieval, and legibility.

Comments: ______________________________

____

____

____

____

____

____

What is your occupation? ______________________________

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

GA27-3136-10

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. E89
P.O. Box 12195
Research Triangle Park, N.C. 27709-9990

Fold and tape | Please Do Not Staple | Fold and tape

IBM®

Reader's Comment Form

Systems Network Architecture
Formats

Publication No. GA27-3136-10

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Note: Copies of IBM Publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.

Possible topics for comment are: clarity, accuracy, completeness, organization, coding, retrieval, and legibility.

Comments: ____________________

____

____

____

____

____

____

What is your occupation? ____________________

If you wish a reply, give your name, company, mailing address, and date:

____

____

____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

GA27-3136-10

Reader's Comment Form

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. E89
P.O. Box 12195
Research Triangle Park, N.C. 27709-9990

Fold and tape | Please Do Not Staple | Fold and tape

IBM ®

Publication Number
GA27-3136-10

Printed in USA

GA27-3136-10