Systems Network Architecture Formats

GA27-3136-18

IBM

Systems Network Architecture Formats

GA27-3136-18

Note!

See “Notices” on page v.

Nineteenth Edition (July 1999)

This is a revision of GA27-3136-17, which is now obsolete. **Changes to the previous edition are indicated by change bars in the lefthand margin.**

Additional copies of this publication can be downloaded via anonymous FTP from Internet node **ftp.networking.ibm.com** under the name of **aiw/formats/formats18.psbin** - the file is a PostScript file, but treat it as binary when you get it and print it. There are also compressed versions of the file (**aiw/formats/formats18.zipbin** and **aiw/formats/formats18.exebin**) as well as a BookManager &gem. version (**aiw/formats/formats18.bookbin**).

Order other 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 reader′s comments is provided at the back of this publication. If the form has been removed, comments may be addressed to:

IBM Corporation
Networking Software
Department BRQA/Building 502
PO Box 12195
Research Triangle Park, North Carolina 27709-9990

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Contents

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:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

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

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

Trademarks

The following terms are trademarks of the IBM Corporation in the United States and/or other countries.

Advanced Peer-to-Peer Networking
APPN
IBM
Intelligent Printer Data Stream
S/390
System/390
SystemView
VTAM

About This Book

This book describes Systems Network Architecture (SNA) formats, including those used between subarea nodes, between subarea nodes and peripheral nodes, and between nodes implementing Advanced Peer-to-Peer Networking (APPN) and/or low-entry networking (LEN) protocols. All significant technical and editorial changes from the previous edition are marked by change bars or asterisks in the lefthand margin. Because of their sheer volume, the formats for SNA management services appear in a standalone book, *SNA Management Services Formats*. That book and this provide full coverage for SNA formats.

The specific implementation by the product you are using may differ from the description contained in this book. For specific implementation information, refer to the appropriate product publications.

Who Should Use This Book

This manual is directed to system programmers and program support personnel, particularly those doing network problem diagnostics, and to implementers, including vendors incorporating APPN into their networking products.

Softcopy

This book will also be made available on an electronic bookshelf as part of *IBM Networking Softcopy Collection Kit* (SK2T-6012) on compact disk read-only memory (CD-ROM).

How This Book is Organized

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

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

Chapter 2 identifies the formats and meanings of the bytes in a High-Performance Routing (HPR) header.

Chapter 3 identifies the formats and meanings of the information-field bytes in a data link control (DLC) Exchange Identification (XID) command and response.

Chapter 4 identifies the formats and meanings of the bytes in a transmission header.

Chapter 5 identifies the formats and meanings of the bytes in a request or response header.

Chapter 6 identifies the formats and meanings of the bytes in request units and response units.

Chapter 7 explains the transmission services and function management profiles that SNA defines to describe session characteristics.

Chapter 8 identifies the formats and meanings of the bytes in user-structured subfields that appear in a request or response unit.

Chapter 9 identifies the formats and meanings of the control vectors, control lists and session keys that appear in a request or response unit.

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

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

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

Chapter 13 provides a summary of general data stream identifier (GDS ID) value assignments, identifies the formats and meanings of the general data stream (GDS) variables that are specific to SNA service transaction programs, and identifies the GDS variables that are for general use.

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

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

Chapter 16 identifies the 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 lists the common structures for SNA condition reports.

Related Publications

Related publications, providing overview and protocol information, are:

- *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 Management Services Formats* (GC31-8302)
- *Systems Network Architecture APPN Architecture Reference* (SC30-3422)
- *Systems Network Architecture APPN Branch Extender Architecture Reference* (SV40-0129) — available in softcopy only on the CD-ROM described in “Softcopy” on page vii

- *Systems Network Architecture APPN Dependent LU Requester Architecture Reference* (SV40-1010) — available in softcopy only on the CD-ROM described in “Softcopy” on page vii
- *Systems Network Architecture APPN Extended Border Node Architecture Reference* (SV40-1018) — available in softcopy only on the CD-ROM described in “Softcopy” on page vii
- *Systems Network Architecture APPN High-Performance Routing Architecture Reference* (SV40-1018) — available in softcopy only on the CD-ROM described in “Softcopy” on page vii
- *Systems Network Architecture: Sessions Between Logical Units* (GC20-1868)
- *Systems Network Architecture: Transaction Programmer's Reference Manual for LU Type 6.2* (GC30-3084)
- *Systems Network Architecture Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2* (SC30-3269)
- *Systems Network Architecture LU 6.2 Reference: Peer Protocols* (SC31-6808)
- *Systems Network Architecture Sync Point Services Reference* (SC31-8134)
- *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)
- *Inside APPN and HPR: The Essential Guide to New SNA* (SG24-3669)
- ISO/IEC 8802-2:1994 (ANSI/IEEE Std 802.2, 1994 Edition), *Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical Link Control*.
- ISO/IEC 8802-3:1996 (ANSI/IEEE Std 802.3, 1996 Edition), *Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*.
- RFC 768, *User Datagram Protocol*, Internet Engineering Task Force (August 1980)
- RFC 791, *Internet Protocol*, Internet Engineering Task Force (September 1981)
- RFC 1034, *Domain Names - Concepts and Facilities*, Internet Engineering Task Force (November 1987)
- RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, Internet Engineering Task Force (July 1993)

- RFC 1490, *Multiprotocol Interconnect over Frame Relay*, Internet Engineering Task Force (July 1993)

For detailed instructions for obtaining most Internet Engineering Task Force (IETF) Requests for Comments (RFC), including lists of FTP and mail server hosts in various countries, send an electronic-mail message to “rfc-info@isi.edu” with the message body “help:ways_to_get_rfcs”.

Summary of Changes

Major additions and changes for GA27-3136-18:

This edition introduces formats for the following functions: increasing Locate reply length beyond 1024 bytes, Responsive mode Adaptive Rate-Based (ARB) congestion control, Triple Data Encryption Standard (Triple DES), HPR support for Frame Relay SVCs, subarea FID4 and FIDF PIU checksum support, DLUR CP-SVR pipe persistence, and LU 6.2 sync point do know/don't know protocol.

Change bars indicate technical material that has not appeared previously in GA27-3136-17.

Chapter 1. DLC Links

Introduction

Three data link controls are described in summary form in this chapter: “Synchronous Data Link Control (SDLC),” beginning on this page, “ATM DLC” on page 1-11, and the “Token-Ring Network DLC” on page 1-10. In addition, two other sections describe HPR’s use of frame-relay and IEEE 802.2 LLC formats.

Synchronous Data Link Control (SDLC)

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

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

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

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

Link Header (Flag)

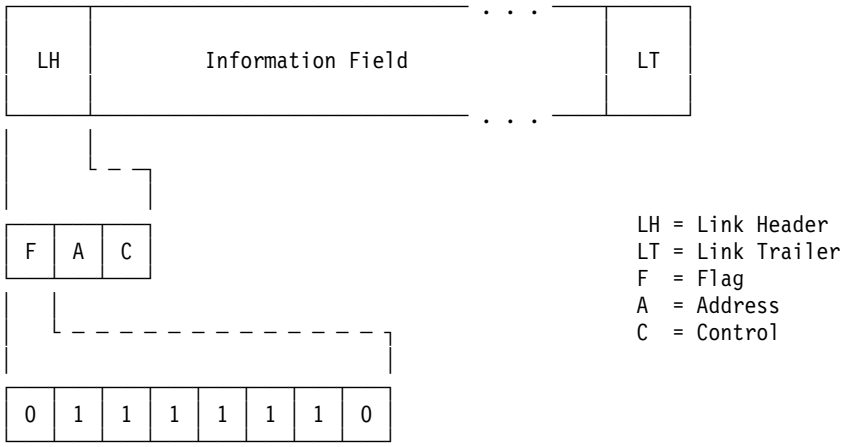


Figure 1-1. Flag Field of Link Header. Always X′7E′ (01111110)

All frames begin with a Flag field. The configuration of the flag is always 01111110 (X′7E′). Because frames also *end* with flags (see link trailer), the trailing flag of one frame may serve as the leading flag of the next frame. When receiving, the last 0 in the trailing flag may also be the first 0 in the next leading flag, as Figure 1-2 on page 1-4 illustrates.

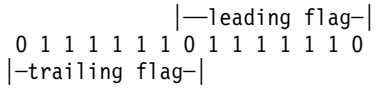


Figure 1-2. Shared Trailing/Leading 0 in SDLC Flags

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

Link Header (Address)

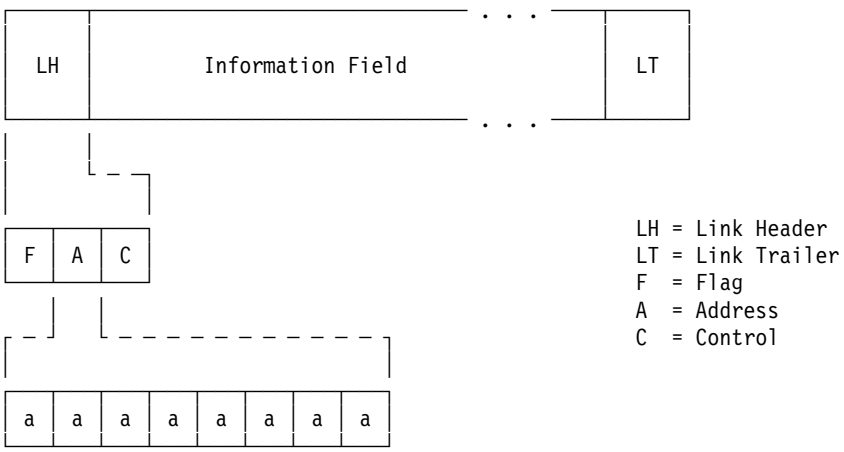


Figure 1-3. Address Field of Link Header. (aaaaaaaa)

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

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

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

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

Link Header (Control)

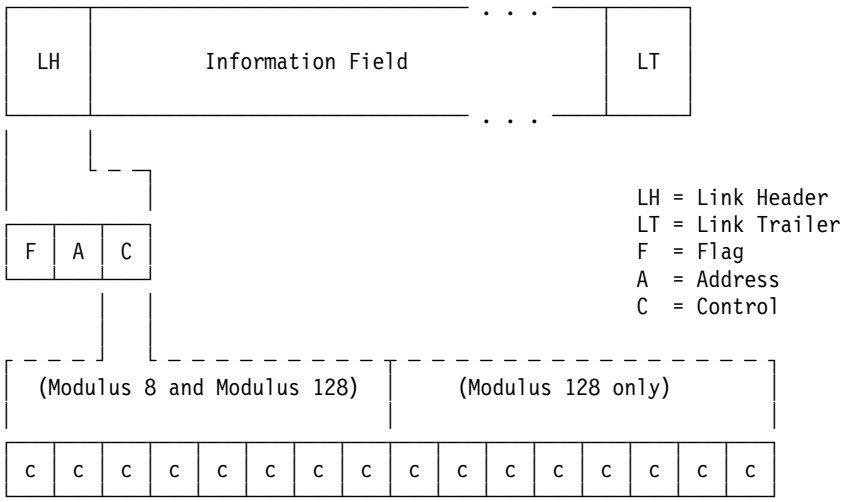


Figure 1-4. Control Field of Link Header. (Eight bits for modulus 8; sixteen bits for modulus 128)

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

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

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

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

The Information Format is identified by a 0 in the low-order bit of the first or only byte of the Control field. In an Information Format SDLC command or response, the Information field contains a PIU (path information unit). The remaining chapters of this book, with the exception of Chapter 2, discuss the contents of the PIU.

Figure 1-5 lists the SDLC commands and responses for modulus 8 (one-byte) Control fields; Figure 1-6 lists them for modulus 128 (two-byte) Control fields. Figure 1-7 describes the Information field of the Frame Reject (FRMR) response frame, which is one of the unnumbered formats listed in Figure 1-5.

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

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

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

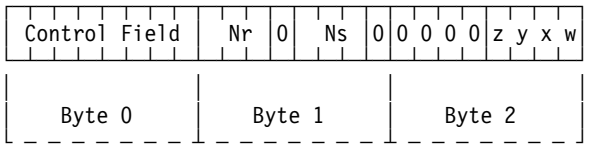
FORMAT	BINARY CONFIGURATION	HEX EQUIVALENT	COMMAND NAME	ACRO–NYM
Unnumbered Format	same as modulus 8 (one–byte), as in Figure 1-5.			
Supervisory Format	0000 0001 RRRR RRR P/F	X'01**'	Receive Ready	RR
	0000 0101 RRRR RRR P/F	X'05**'	Receive Not Ready	RNR
	0000 1001 RRRR RRR P/F	X'09**'	Reject	REJ
Information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

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

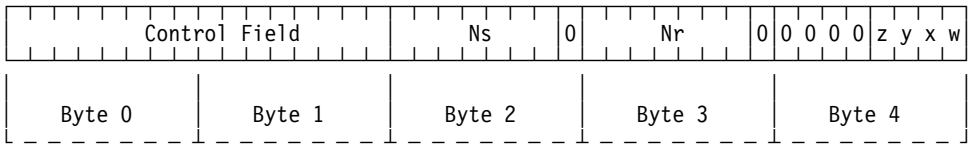
Figure 1-6. Control Fields for SDLC Commands and Responses—Modulus 128

Information Field of the FRMR Response Frame

Modulus 8:



Modulus 128:



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

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

Figure 1-7. Information Field of the FRMR Response Frame – Modulus 8 and Modulus 128. In each byte, the low-order bit is sent first and the high-order bit is sent last.

Link Trailer (Frame Check Sequence)

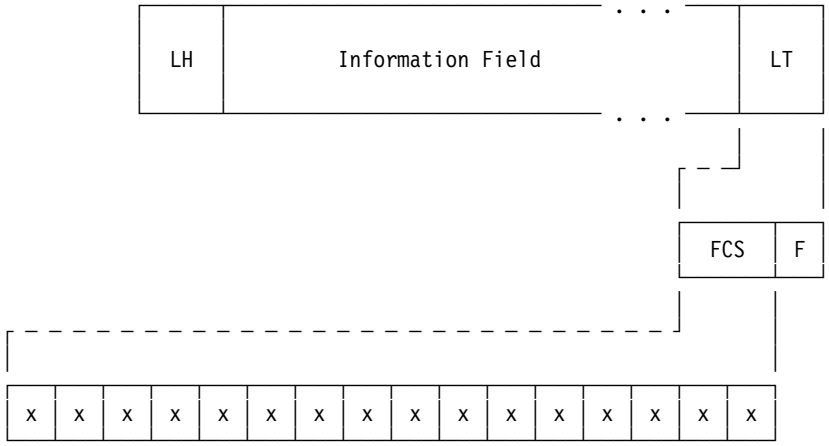


Figure 1-8. Frame Check Sequence Field of Link Trailer

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.

Link Trailer (Flag)

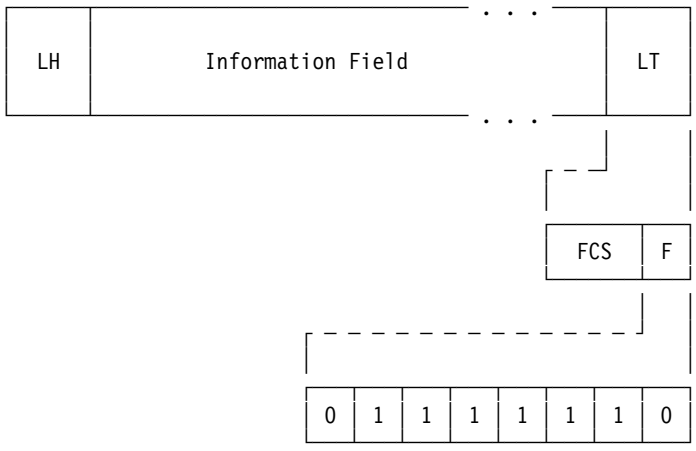


Figure 1-9. Flag Field of Link Trailer. Always X'7E' (01111110)

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

Token-Ring Network DLC

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

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

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

Figure 1-10. LLC Commands and Responses

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

The token-ring network DLC, in contrast to SDLC, transmits the high-order bit first and the low-order bit last within each byte. Also, zero-bit insertion is ***not*** 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*.

ATM DLC

Frames on ATM TGs

All transmissions on an ATM TG are organized in an IEEE 802.2 LLC frame. Frames begin with an 8-byte RFC 1483 header. The value X′4C80′ in the Layer-2 Protocol Indentifier field of the RFC 1483 header indicates that the RFC 1483 header is followed by an additional IEEE 802.2 LLC header; unnumbered information and the commands and responses defined for token-ring network DLC (see Figure 1-10 on page 1-10) may be present. When LDLC is in use, only unnumbered information, XID command and response, TEST command and response, DISC command, and DM response may be present. The value X′5081′ indicates the additional IEEE 802.2 LLC header is not present; this format is used to support service interworking with frame relay DLC nodes. The value X′7085′ in the Layer-3 Protocol Identifier field of the RFC 1483 header indicates that an HPR network layer packet (NLP) follows the RFC 1483 header and the additional IEEE 802.2 LLC header (if present).

RFC 1483 Header

The contents of the RFC 1483 header are defined by RFC 1483 and ATM Forum Implementation Agreement 94-0615. The header begins with 1-byte DSAP, SSAP, and Control fields. When these fields are coded X′FEFE03′ indicating the format of the RFC 1483 header, the fourth byte is a network layer protocol identifier (NLPID). An NLPID value of X′09′ indicates that the NLPID is followed by a 2-byte layer-2 protocol identifier (L2) and a 2-byte layer-3 protocol identifier (L3); the format of the L2 and L3 fields is the same as that specified for broadband low-layer information in ITU-T Recommendation Q.2931. The values for the L2 and L3 fields are defined in ATM Forum contribution 94-0615.

RFC 1483 Header

Byte	Bit	Content
0		DSAP: X′FE′
1		SSAP: X′FE′
2		Control: X′03′
3		NLPID: X′09′
4– 5		Layer-2 protocol identifier: X′4C80′ IEEE 802.2 LLC header present X′5081′ No layer-2 header present
6– 7		Layer-3 protocol identifier: X′7083′ SNA APPN (FID2) including XID3, and IEEE 802.2 LLC commands and responses X′7085′ SNA APPN/HPR (NLP) including LDLC frames

HPR Use of Frame Relay Formats

Note: For a general reference on frame relay, see *User-to-Network Implementation Agreement (UNI) FRF 1.1*, Frame Relay Forum Technical Committee, January 19, 1996.

Frame Relay Format for LLC Commands and Responses and FID2 PIUs (Including FID2 Route Setup)

The format described here is documented further in *Multiprotocol Encapsulation Implementation Agreement FRF.3.1*, Frame Relay Forum Technical Committee, June 22, 1995. It is the same format as used for the base APPN (FID2). For LDLC, this format is used for XID3, TEST, DISC, and DM.

Frame Relay Format for LLC Commands and Responses and FID2 PIUs

Byte	Bit	Content
0– 1		T1.618 address (DLCI)
2		Control: X′03′
3		NLPID: X′08′
4– 5		L2 protocol identifier: X′4C80′ 802.2 header present
6– 7		L3 protocol identifier: X′7083′ SNA APPN(FID2)
8– 1 1		802.2 header
8		DSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
9		SSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
10– 1 1		Control fields: set as appropriate
12 – n		Remainder of PDU: XID3 or TEST information field, FID2 PIU, or null for other LLC commands and responses
n + 1 – n + 2		Frame check sequence

Frame Relay Format for NLPs When Doing No Error Recovery with No 802.2 Header

The format described here is documented further in *Multiprotocol Encapsulation Implementation Agreements FRF.3.1*, *Frame Relay Forum Technical Committee*, June 22, 1995.

Frame Relay Format for NLPs When Doing No Error Recovery with No 802.2 Header

Byte	Bit	Content
0– 1		T1.618 address (DLCI)
2		Control: X′03′
3		NLPID: X′08′
4– 5		L2 protocol identifier: X′5081′ indicates no L2 protocol used
6– 7		L3 protocol identifier: X′7085′ SNA APPN/HPR(NLP)
8 – n		Remainder of PDU, i.e., the HPR NLP
n + 1 – n + 2		Frame check sequence

Frame Relay Format for NLPs When Doing No Error Recovery in 802.2 UI Frames

The format described here is documented further in *Multiprotocol Encapsulation Implementation Agreements FRF.3.1*, *Frame Relay Forum Technical Committee*, June 22, 1995. For LDLC, this format is used for either LDLC specific messages or HPR session and control traffic.

Frame Relay Format for NLPs When Doing No Error Recovery with 802.2 Header

Byte	Bit	Content
0– 1		T1.618 address (DLCI)
2		Control: X′03′
3		NLPID: X′08′
4– 5		L2 protocol identifier: X′4C80′ L2 protocol (802.2) used
6– 7		L3 protocol identifier: X′7085′ SNA APPN/HPR(NLP)
8– 1 0		802.2 header
8		DSAP: the destination SAP obtained from subfield X′80′ in control vector X′61′ in the received XID3
9		SSAP: the source SAP obtained from subfield X′80′ in control vector X′61′ in the sent XID3
10		Control field: X′03′ unnumbered information
11 – n		Remainder of PDU, i.e., the HPR NLP
n + 1 – n + 2		Frame check sequence

Frame Relay Format for HPR NLPs When Doing Error Recovery

The format described here is documented further in *Multiprotocol Encapsulation Implementation Agreements FRF.3.1*, *Frame Relay Forum Technical Committee*, June 22, 1995.

Frame Relay Format for HPR NLPs When Doing Error Recovery

Byte	Bit	Content
0– 1		T1.618 address (DLCI)
2		Control: X′03′
3		NLPID: X′08′
4– 5		L2 protocol identifier: X′4C80′ 802.2 header present
6– 7		L3 protocol identifier: X′7085′ SNA APPN/HPR(NLP)
8– 1 1		802.2 header
8		DSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
9		SSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
10– 1 1		Control fields: set as appropriate
12 – n		Remainder of PDU, i.e., the HPR NLP
n + 1 – n + 2		Frame check sequence

HPR Use of Token-Ring and Ethernet IEEE 802.2 LLC Formats

Token-Ring and Ethernet IEEE 802.2 LLC Formats for HPR NLPs When Doing No Error Recovery

Token-Ring and Ethernet IEEE 802.2 LLC Format for HPR NLPs When Doing No Error Recovery

Byte	Bit	Content
0		DSAP: the LLC destination SAP value used for transmitting HPR NLPs without performing link-level error recovery on them; same as the one received on XID3 from the adjacent node during the XID3 link activation exchange, or the default X′C8′ if none was received
1		SSAP: the LLC source SAP value used for transmitting HPR NLPs without performing link-level error recovery on them; same as the one sent in XID3 by this node during the XID3 link activation exchange, or the default X′C8′ if none was sent
2		Control field: X′03′ unnumbered information (only value used)

Token-Ring and Ethernet IEEE 802.2 LLC Formats for XID, FID2 PIUs, and HPR NLPs When Doing Error Recovery

The format of this field is the same as used for base APPN.

Token-Ring and Ethernet IEEE 802.2 LLC 802.2 Formats for HPR NLPs When Doing Error Recovery

Byte	Bit	Content
0		DSAP: same as for base APPN (i.e., X′04′ or an installation-defined value)
1		SSAP: same as for base APPN (i.e., X′04′ or an installation-defined value)
2– 3		Control fields: set as appropriate

HPR Use of ATM Formats

Note: See “ATM DLC” for the general ATM format description.

ATM Format for LLC Commands and Responses and FID2 PIUs

The formats described here are used for either LDLC or LLC2. For LDLC, this format is used for XID, TEST, DISC, and DM. For LLC2, this format is used for LLC2 commands and responses and reliable delivery of FID2 PIUs.

ATM Format for LLC Commands and Responses and FID2 PIUs

Byte	Bit	Content
0		DSAP: X′FE′
1		SSAP: X′FE′
2		Control: X′03′
3		NLPID: X′09′
4– 5		Layer-2 protocol identifier: X′4C80′ IEEE 802.2 LLC header present
6– 7		Layer-3 protocol identifier: X′7083′ SNA APPN(FID2)
8 – p		IEEE 802.2 LLC header (see note 1)
8		DSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
9		SSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
10 – p		Control (1 or 2 bytes): set as appropriate
p + 1 – n		Remainder of PDU: XID3 or TEST information field, FID2 PIU, or null for other LLC commands and responses
		Note 1: Rules for encoding the IEEE 802.2 LLC header can be found in ISO/IEC 8802-2:1994 (ANSI/IEEE Std 802.2, 1994 Edition), *Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical Link Control*.

ATM Format for NLPs in UI Frames

For LDLC, this format is used for either LDLC specific messages or HPR session and control traffic. For LLC2, this format is used for HPR session and control traffic if SAPs are used with no link-level error recovery.

ATM Format for NLPs in UI Frames

Byte	Bit	Content
0		DSAP: X′FE′
1		SSAP: X′FE′
2		Control: X′03′
3		NLPID: X′09′
4– 5		Layer-2 protocol identifier: X′4C80′ IEEE 802.2 LLC header present
6– 7		Layer-3 protocol identifier: X′7085′ SNA APPN/HPR (NLP)
8– 1 0		IEEE 802.2 LLC header
8		DSAP: the destination SAP obtained from the IEEE 802.2 LLC (X′80′) subfield in the HPR Capabilities (X′61′) control vector in the received XID3 (see note 1)
9		SSAP: the source SAP obtained from the IEEE 802.2 LLC (X′80′) subfield in the HPR Capabilities (X′61′) control vector in the sent XID3 (see note 2)
10		Control: X′03′ UI with P/F bit off
11 – n		Remainder of PDU: NLP
		Note 1: The User-Defined Address bit is considered part of the DSAP. The Individual/Group bit in the DSAP field is set to 0 by the sender and ignored by the receiver.
		Note 2: The User-Defined Address bit is considered part of the SSAP. The Command/Response bit in the SSAP field is set to 0 by the sender and ignored by the receiver.

ATM Format for NLPs with No IEEE 802.2 LLC Header

When LLC2 is in use, the format described here is for HPR session and control traffic when neither SAPs nor link-level error recovery is used.

ATM Format for NLPs with No LLC Header

Byte	Bit	Content
0		DSAP: X′FE′
1		SSAP: X′FE′
2		Control: X′03′
3		NLPID: X′09′
4– 5		Layer-2 protocol identifier: X′5081′ No layer-2 header present
6– 7		Layer-3 protocol identifier: X′7085′ SNA APPN/HPR (NLP)
8 – n		Remainder of PDU: NLP

ATM Format for NLPs when Doing ERP

When LLC2 is in use, the format described here is for HPR session and control traffic when link-level error recovery is required.

ATM Format for NLPs with ERP

Byte	Bit	Content
0		DSAP: X′FE′
1		SSAP: X′FE′
2		Control: X′03′
3		NLPID: X′09′
4– 5		Layer-2 protocol identifier: X′4C80′ IEEE 802.2 LLC header present
6– 7		Layer-3 protocol identifier: X′7085′ SNA APPN/HPR (NLP)
8 – p		IEEE 802.2 LLC header (see note 1)
8		DSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
9		SSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
10– 1 1		Control: set as appropriate
12 – n		Remainder of PDU: NLP
		Note 1: Rules for encoding the IEEE 802.2 LLC header can be found in ISO/IEC 8802-2:1994 (ANSI/IEEE Std 802.2, 1994 Edition), *Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical Link Control*.

HPR Use of IP Formats

IP Format for LLC Commands and Responses

The formats described here are used for the following LLC commands and responses: XID command and response, TEST command and response, DISC command, and DM response.

IP Format for LLC Commands and Responses

Byte	Bit	Content
0 – p		IP header (see note 1)
p+1 – p+8		UDP header (see note 2)
p+9 – p+11		IEEE 802.2 LLC header (see note 3)
p+9		DSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
p+10		SSAP: same as for the base APPN (i.e., X′04′ or an installation-defined value)
p+11		Control: set as appropriate
p+12 – n		Remainder of PDU: XID3 or TEST information field, or null for DISC command and DM response
		Note 1: Rules for encoding the IP header can be found in *RFC 791*.
		Note 2: Rules for encoding the UDP header can be found in *RFC 768*.
		Note 3: Rules for encoding the IEEE 802.2 LLC header can be found in ISO/IEC 8802-2:1994 (ANSI/IEEE Std 802.2, 1994 Edition), *Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 2: Logical Link Control*.

IP Format for NLPs in UI Frames

This format is used for either LDLC specific messages or HPR session and control traffic.

IP Format for NLPs in UI Frames

Byte	Bit	Content
0 – p		IP header (see note 1)
p+1 – p+8		UDP header (see note 2)
p+9 – p+11		IEEE 802.2 LLC header
p+9		DSAP: the destination SAP obtained from the IEEE 802.2 LLC (X′80′) subfield in the HPR Capabilities (X′61′) control vector in the received XID3 (see note 3)
p+10		SSAP: the source SAP obtained from the IEEE 802.2 LLC (X′80′) subfield in the HPR Capabilities (X′61′) control vector in the sent XID3 (see note 4)
p+11		Control: X′03′ UI with P/F bit off
p+12 – n		Remainder of PDU: NLP
		Note 1: Rules for encoding the IP header can be found in *RFC 791*.
		Note 2: Rules for encoding the UDP header can be found in *RFC 768*.
		Note 3: The User-Defined Address bit is considered part of the DSAP. The Individual/Group bit in the DSAP field is set to 0 by the sender and ignored by the receiver.
		Note 4: The User-Defined Address bit is considered part of the SSAP. The Command/Response bit in the SSAP field is set to 0 by the sender and ignored by the receiver.

End of Chapter 1

Chapter 2. High-Performance Routing (HPR) Headers

High-Performance Routing (HPR) Introduction

This chapter describes the header formats for High-Performance Routing (HPR).

Other HPR formats are described in Chapter 1, “DLC Links,” Chapter 4, “Transmission Headers (THs),” Chapter 6, “Request/Response Units (RUs),” Chapter 9, “Common Fields,” and Chapter 13, “GDS Variables.”

HPR Link Frame

This is the format for all frames flowing over HPR links (i.e., links where both sides agree to run HPR).

HPR Link Frame

Byte	Bit	Content
0 – q		DLC header, where the format depends on the DLC type: The DLC header indicates whether the I-field contains an XID or not.
q+1 – r		An XID3 I-field or an APPN/HPR packet (a network layer packet [NLP] or a FID2 PIU)
r+1 – s		DLC trailer: The exact format depends on the DLC type, but within every DLC trailer a cyclic redundancy check (CRC) covers the DLC header and packet fields (bytes 0 – r). This CRC is the *only* data integrity check used by HPR, and is required for every link (i.e., exactly as for the base APPN).

APPN/HPR Packet

This is the format for all packets flowing over HPR links.

2.1 Packet

Byte	Bit	Content
0	0– 3	Packet type, indicating the format of the packet: 0010 FID2 PIU (TH(FID2)-RH-RU) used for base APPN traffic 110r NLP (network layer packet) used for HPR traffic (The r-bit is currently reserved and so should not be checked)
	4– 7	Bits 4–7 of either the FID2 PIU or NLP packet
1 – n		Rest of FID2 PIU or NLP packet

Note: The Packet Type field is actually the first four bits of either the FID2 PIU or NLP packet. The packet type of the FID2 PIU is 0010, which is the FID type field of the TH (where 0010 indicates a FID type of 2). The NLP is defined such that it can never have its first four bits equal to the value 0010. The packet type should be checked for FID2 PIU (0010) and, if equal, routed to the FID2 path control process. If not equal (¬0010), the packet is an NLP and is routed to the network layer to be processed.

Network Layer Packet (NLP)

This contains the network layer header, the RTP transport header or an NLP Function Routing header, and session traffic or HPR control traffic.

Network Layer Packet

Byte	Bit	Content
0 – k		Network layer header (NHDR), as described in a following section
k+1 – m		RTP transport header (THDR) or NLP Function Routing header, as described in a following section
m+1 – n		Data, including FID5 PIUs for session traffic (see Chapter 4, “Transmission Headers (THs)” for FID5 TH information) or HPR GDS variables (e.g., Route Setup) for nonsession traffic

Network Layer Header (NHDR)

The NHDR is used between HPR nodes. It is constructed by the **origin** node, processed by each **intermediate** node, and received and processed by the final **destination** node.

Network Layer Header (NHDR)

Byte	Bit	Content
0	0– 2	Switching mode (SM): 001 reserved (to avoid conflict with FID2 PIUs) 010 reserved (to avoid conflict with FID4 PIUs) 101 function routing 110 automatic network routing (ANR) 111 reserved (to avoid conflict with FIDF PIUs) **Origin node:** Always sets this field to ANR (110) or function routing (101). **Intermediate and Destination node:** If this field does not indicate ANR (110) or function routing (101), the packet is discarded and (optionally) the error is reported locally or to a network management focal point. The sense data associated with this error is X′801E0000′.
	3– 4	Reserved
	5– 6	Transmission priority field (TPF): 00 low (L) 01 medium (M) 10 high (H) 11 network (N) **Origin and Intermediate node:** The origin node sets this field to the priority associated with the RTP connection (the RTP connection identifier is in the THDR). Priority routing is implemented by both origin and intermediate nodes by giving preference to higher-priority packets when sending them out over a link. Note: The transmission priority values are the same for FID2 and NLP packets; therefore, priority routing can be done even when sending both FID2 and NLP packets over the same link. **Destination node:** The destination node ignores this field.
	7	Reserved
1	0– 3	Function type (when Switching Mode field is set to 101; otherwise, reserved): 0001 logical data link control (LDLC)
	4	Time-sensitive packet indicator: Indicates whether this NLP is time sensitive, meaning it cannot tolerate excessive delays along the path. For example, any product-specific blocking functions should flush all data currently blocked when a time-sensitive NLP is received so that it may be processed and forwarded immediately. Two conditions cause the time-sensitive indicator to be set to 1. The first is an NLP that contains a Status Requested Indicator (SRI in the THDR) and Respond ASAP Indicator (RASAPI in the THDR) each with a value of 1. The second is an NLP that contains a Status segment (in the THDR). 0 This NLP is not time sensitive. 1 This NLP is time sensitive.
	5– 6	Slowdown 1 and 2 congestion indicators: Indicate whether a minor (slowdown 1) or significant (slowdown 2) congestion condition exists along the path (e.g., in a frame-relay subnet). These indicators may (optionally) be set to 1 (never changed back to 0) by any node along the path (origin, intermediate, or destination nodes). This information is used by the RTP endpoints to regulate the adaptive rate-based (ARB) flow/congestion control algorithm. Slowdown 1 causes ARB to reduce the rate of sending data on the RTP connection by 12.5% and slowdown 2 causes a 25% reduction. If both the slowdown 1 and slowdown 2 indicators are set, the value of slowdown 2 is used by ARB.

Network Layer Header (NHDR)

Byte	Bit	Content
	5	Slowdown 1 congestion indicator: 0 Slowdown 1 congestion condition does not exist. 1 Slowdown 1 congestion condition exists.
	6	Slowdown 2 congestion indicator: 0 Slowdown 2 congestion condition does not exist. 1 Slowdown 2 congestion condition exists.
	7	Reserved
For SM = ANR (110):		
2 – m		ANR routing field (ANRF): AL1-AL2-...-ALn-X′FF′, where AL1, AL2, ..., ALn are ANR labels associated with the *n* TGs (links) in the route (path). Each TG has two ANR labels (one for each direction). A string of these labels (AL1-AL2-...-ALn) represents a path through the network. No delimiters separate the labels. *Label assignment:* ANR labels are assigned by each node for the outbound direction (i.e., towards the adjacent node) and are unique within the assigning node. *Label size:* An ANR label may vary from 1 to 8 bytes, but in order to conserve header space it should be only as long as necessary. Typically, ANR labels are 1 or 2 bytes long. The size of ANR labels may vary for labels assigned within a single node as long as they remain unambiguous within that node. *High-order bit:* The high-order bit of each label is currently always set to 1. *Label for network connection endpoint (NCE):* The last ANR label in the string (ALn) is an NCE identifier and identifies a component within the final destination node that is to process this packet. An NCE identifier is always present. Only nodes that support the RTP 1401 option set need to understand NCE identifiers. *X′FF′ delimiter:* The X′FF′ delimits the end of the ANRF. This means ANR labels themselves may never contain an X′FF′ character. **Origin node:** The origin node includes in this field an ANR label string that represents the entire route, i.e., with an ANR label for each link along the path followed by the NCE identifier. The origin node strips off the ANR label representing the first link of the route before sending the packet. **Intermediate node:** When an intermediate node receives a packet, it examines the first (“next”) ANR label to determine the link the packet is to be sent out on. It strips off the label before sending the packet. If the ANR label is unknown, it discards the packet. **Destination node:** The destination node routes the packet to the internal component identified by the first (in this case, the *final*) ANR label, which indicates an NCE. If this label is unknown, it discards the packet. *Subarea only:* **NCP and VTAM labels:** See Figure 2-1 on page 2-8 for ANR label conventions used by NCP and VTAM.
m+1		Reserved
For SM = function routing (101):		
2– 4 (= m)		Function routing field (FRF): a 2-byte function routing address (FRA) followed by a X′FF′ value delimiting the end of the FRF (meaning an FRA never contains the X′FF′ value). *Note:* When the Function Type field is set to LDLC (0001), the default value (a base-level support requirement for all implementations) for FRA is X′0001′; other system-defined values may be carried to identify multiple logical nodes at the same ATM address.
m+1		Reserved

ANR Label Type	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
NCP ANR Labels								
Subarea routing	1000 wxTT	ssss ssss	ssss ssss	0nng EEEE				
BNN-BNN HBN TP	1001 0000	eeee eeee	eeee eeee	0nn0 wxTT				
BNN-BNN	1100 wxTT	eeee eeee	eeee eeee					
Outboard ANR routing	111a aaaa	dddd dddd	rrrr rrrr	rrrr rrrr	rrrr rrrr			
VTAM ANR Labels								
Subarea routing	1010 p0yz	TTg0 EEEE	ssss ssss	ssss ssss	0000 0000	mnn0 0000	uuuu uuuu	uuuu uuuu
BNN	1000 p0yz	kkkk kkkk	eeee eeee	eeee eeee	0000 0000	mnn0 0000	uuuu uuuu	uuuu uuuu
NCE	1101 qcRv	0000 0000	LLLL LLLL	LLLL LLLL	LLLL LLLL	LLLL LLLL	LLLL LLLL	LLLL LLLL

Legend:

a...a	Outboard DLC adapter number
c	CP-CP RTP (if 1, this NCE is for a CP-CP NCE)
d...d	Outboard DLC line number
e...e	destination link station element address (NCP) or PU element address (VTAM)—but see w, x, y, and z flags
EEEE	explicit route number (ERN)
g	TG reorder not required (if 1, the ER can support PIUs being sent "TG reorder not required")
k...k	reuse count (wraps at X'FE')
L...L	MNPS condensed LU name (present if q flag set)
m	TPF mapping indicator (if 1, overlay NLP byte 0 TPF field with nn)
nn	new TP bit setting (for NLP byte 0)
p	MNPS label (if 1, two index bytes have been appended as bytes 6–7)
q	MNPS label (if 1, bytes 2–7 contain a condensed MNPS LU name)
r...r	Local resource identifier (LIM.LRID) of the link station
R	RTP Route Setup (if 1, this NCE is for an RTP Route Setup)
s...s	subarea address
TT	subarea Transmission Priority Field
u...u	MNPS index value
v	resource may be owned by an NCP (if 0, resource is not owned by an NCP; if 1, resource may be owned by an NCP)
w	substitution flag: If byte 1 was X'FF', this bit is set to 1 and byte 1, bit 4 is set to 0.
x	substitution flag: If byte 2 was X'FF', this bit is set to 1 and byte 2, bit 5 is set to 0.
y	substitution flag: If byte 2 was X'FF', this bit is set to 1 and byte 2 is replaced with X'00'.
z	substitution flag: If byte 3 was X'FF', this bit is set to 1 and byte 3 is replaced with X'00'.

Figure 2-1. Summary of NCP's and VTAM's ANR labels

NLP Function Routing Header

Ordinarily, an HPR network layer packet is composed of a network layer header (NHDR), an RTP transport header (THDR), and a data field. (The format for the NHDR is shown in “Network Layer Header (NHDR)” on page 2-7.) However, a value of 101 in the Switching Mode field of the NHDR indicates that the mode is function routing. For function routing, a value of X′1′ in the Function Type field of the NHDR indicates the function type is logical data link control (LDLC). When LDLC is specified, there is no THDR, a 1-byte NLP function routing header follows the NHDR, and the NLP contains no data field.

NLP Function Routing Header

Byte	Bit	Content
0		LDLC request/response unit identifier: X′03′ XID complete request (XID_DONE_RQ) — LDLC frame indicating that an activation XID exchange is complete X′04′ XID complete response (XID_DONE_RSP) — LDLC frame indicating agreement that an activation XID exchange is complete and acknowledging an allied XID complete request

RTP Transport Header (THDR)

The THDR contains information necessary for creating and maintaining an RTP transport connection. Its length is always an integral multiple of four bytes (e.g., 20, 24, 28). Variable-length data is contained in control vectors (CVs) or RTP optional segments, which carry additional information about the RTP transport connection. CVs may also be contained within the segments. For performance reasons, the segments and CVs are aligned on word (4-byte) boundaries relative to the beginning of the THDR. So if the THDR begins on a word boundary, all segments and CVs contained within the THDR will also be on word boundaries.

Alignment and padding rules: The format alignment and padding rules for RTP segments and CVs contained within the THDR are as follows:

- All RTP segments and CVs always begin on a word (4-byte aligned) boundary. RTP segments are always a multiple of four bytes in length (using ending pad bytes if necessary to achieve this). In order to align segments and CVs on word boundaries, reserved or pad bytes (X′00′) are used as follows:
  - Up to three trailing pad bytes are inserted after a CV to align a following segment or CV.
  - Reserved bytes are included in a segment or a CV that imbeds a series of CVs in order to align the first nested CV. Pad bytes are used where necessary to align each subsequent CV in the imbedded series.
- The length field of an RTP segment indicates the **number of words** (a word being four bytes) in the segment and includes all imbedded CVs (and any pad bytes between them), and up to three pad bytes that may have been added to the end of the segment to reach the next word boundary.
- The length field of a CV indicates the **number of bytes** in the CV, and includes all imbedded CVs and any pad bytes between them (but not any trailing pad bytes added to align a following segment or CV).

RTP Transport Header (THDR)

Byte	Bit	Content
0–7		Transport connection identifier (TCID) field
	0	TCID assignor: 0 TCID was assigned by the receiving RTP partner, who can thus identify the connection without referring to the Connection Qualifier field. 1 TCID was assigned by the sending RTP partner and is further qualified by a connection qualifier in the Connection Qualifier and/or Source Identifier field (bytes 20 – k).
	1	Reserved
	2–63	A 62-bit transport connection identifier that along with the Connection Qualifier and/or Source Identifier field uniquely identifies an RTP connection. **Suggested use**: Connection identifiers may be chosen so that the receiver can use the low-order bits to directly index into a table of connection records.
8	0	Reserved

RTP Transport Header (THDR)

Byte	Bit	Content
	1	Connection setup indicator (SETUPI): 0 Connection setup segment is not present (¬ SETUP). 1 Connection setup segment is present (SETUP).
	2	Start-of-message indicator (SOMI)—used by RTP for segmenting/reassembling: 0 not start of message (¬ SOM) 1 message starts with first byte of user's data (SOM)
	3	End-of-message indicator (EOMI)—used by RTP for segmenting/reassembling: 0 not end of message (¬ EOM) 1 message ends with last byte of user's data (EOM)
	4	Status requested indicator (SRI): 0 Receiver need not reply with a status segment (¬ SR). 1 Receiver must reply with (at least) a status segment (SR).
	5	Respond as soon as possible indicator (RASAPI), indicating when status should be sent (meaningful only when SRI=SR; otherwise, reserved): 0 Receiver need not transmit reply as soon as possible (¬ RASAP). 1 Receiver must transmit reply as soon as possible (RASAP).
	6	Retry indicator (RETRYI): 0 Sender will retransmit this packet (RETRY). (only value defined)
	7	Reserved
9	0	Last-message indicator (LMI): 0 not last message on this connection (¬ LM) 1 last message on this connection (LM)
	1–2	Reserved
	3–4	Connection qualifier field indicator (CQFI): 00 none present (NOCQF) 01 originator (ORIGIN)
	5	Optional segments present indicator (OSI): 0 No optional segments are present (¬ OS). 1 One or more optional segments are present (OS).
	6–7	Reserved
10–11		Data offset/4: the position of the Data field of the NLP (byte m+1 – n) relative to the beginning of the THDR; this position is always constrained to be a multiple of 4 bytes; the Data Offset/4 field carries the Data field offset value divided by 4.
12–15		Data length field (DLF): the exact number of bytes carried in the Data field; in the algebra of the NLP description, this is n minus m
16–19		Byte sequence number (BSN): sequence number of the first byte of the Data field Each byte in the Data field is (conceptually) assigned a sequence number. The BSN field carries the sequence number of the first byte of the Data field. (When the Data field is empty, this is the sequence number that will be assigned to the first byte of the next non-empty Data field). Additionally, there is an *implicit* special end-of-message character. It is also assigned its own sequence number. Suppose a message is sent indicating end of message (EOM) and carrying N bytes of data. In this case, the byte sequence number is a value of x, which corresponds to the first data byte. The last data byte has a sequence number of x+N−1. The implicit EOM character has a sequence number of x+N. The next message sent will have a BSN of x+N+1. This counting of the end-of-message character allows for detection of errors such as a lost packet carrying an End of Message indicator with an empty Data field. See *HPR Architecture Reference* for more information.

RTP Transport Header (THDR)

Byte	Bit	Content
20 – k		Connection Qualifier/Source Identifier Field (CQF) (present only if CQFI=ORIGIN): The Network Address (X′05′) control vector of the RTP connection endpoint sending this message. The CQF is included in the THDR under the following conditions. • CQF is always present on the Connection Setup packet (the one that contains the Connection Setup optional segment that establishes the RTP connection) and all subsequent packets sent before the packet containing the Connection Identifier Exchange optional segment is received from the partner. • CQF is optionally present on packets that do not contain the Connection Setup segment but do contain the Switching Information (SI) segment. In this case, the SI segment indicates that a path switch is being done. The sender includes the CQF when the node′s CP name has changed. To ensure that the CQF is received by the partner, status is requested and the Status segment with a SYNC value is included (i.e., a status exchange is performed) in the THDR. When the Status segment with matching ECHO is received, the sender knows that the partner received the CQF. Since the NCE identifier may not change, an RTP endpoint receiving a subsequent CQF ignores it.
k + 1 – m		Optional Segment Field (OSF) (at least one segment is present if OSI=OS): Each segment begins on a word boundary and has the following format: Byte Content 0 Segment length/4: length (in 4-byte multiples) of segment type + segment data 1 Segment type 2-t Segment data The sender includes the optional segments in the following order to maximize RTP performance (but the receiver does not check to see that they are in order). X′0E′ Status segment X′0D′ Connection Setup segment X′10′ Connection Identifier Exchange segment X′14′ Switching Information segment X′22′ Adaptive Rate-Based segment X′12′ Connection Fault segment X′0F′ Client Out-of-Band Bits segment The byte following the last byte of the THDR (i.e., the first byte of the Data field) is always on a word boundary, so up to 3 bytes of padding may be present at the end of the THDR.

RTP Optional Segments

Connection Setup (X′0D′) Segment

The Connection Setup segment is carried in any packet with the Setup Packet bit set to 1 to establish an RTP transport connection.

Connection Setup (X′0D′) Segment

Byte	Bit	Content
0		Length/4 of the segment, including this Length field. See “Alignment and padding rules” on page 2-11.
1		Key=X′0D′
2 – k		Segment Data
2– 3		Version of RTP used to conduct the protocol for this connection. X′0101′ Version 1.1 (only value defined)
4	0	Target resource identifier present indicator: 1 present (only value defined)
	1– 2	Reserved
	3	Adaptive rate-based (ARB) flow/congestion control used: 1 ARB flow/congestion control will be used for this connection.
	4	Connection reliability indicator: 1 Connection is reliable. (only value defined)
	5	Dedicated RTP connection (i.e., one session per RTP connection) indicator: 0 dedicated RTP connection not requested 1 dedicated RTP connection requested
	6– 7	Reserved
5– 7		Reserved
8 – k		Control vectors: *Note:* Each imbedded control vector begins on a word boundary.
8 – j		X′28′ Topic Identifier control vector
j+1 – k		Target resource identifier: This field is used to check that the first packet arrives at its intended target. It contains the following control vectors in the order listed: X′03′ Network Identifier control vector X′00′ Node Identifier control vector X′39′ NCE Instance Identifier control vector

Status (X′0E′) Segment

The Status segment is used to convey status information from one end of the connection to the other. This segment is sent as part of a reply to a packet with the Status Request bit set to 1. It may also be sent as an unsolicited request to report lost message bytes and to acknowledge received message bytes.

Status (X′0E′) Segment

Byte	Bit	Content
0		Length/4 of the segment, including this Length field. See “Alignment and padding rules” on page 2-11.
1		Key=X′0E′
2 – n		Segment Data
2		Status bits:
	0	Gap detected by the receiver (GAPDETR): 0 No gaps in the reliable message byte stream have been found. The value in the NABSP field is 0. 1 One or more gaps in the reliable message byte stream have been found. The missing reliable message bytes should be retransmitted as soon as possible. The value in the NABSP field is greater than 0.
	1	Idle: 0 This RTP connection has not been idle. 1 No packets have been received on this RTP connection for a while. The Connection Inactivity timer at this partner has expired. This packet is a “heart beat” (still alive) message.
	2– 7	Reserved
3		Number of acknowledged byte span pairs (NABSP): This is the number of Acknowledged Byte Span Pairs (ABSP) that appear at the end of the Status segment.
4– 5		Status report number (SYNC): Each Status segment is numbered by the sender. This numbering is used to distinguish the current from an old Status segment.
6– 7		Status acknowledgment number (ECHO): This is the most recent SYNC number that was received by the partner that is sending this Status segment. Together, SYNC and ECHO can be used to determine when an exchange of state information has been effected.
8– 1 1		Received sequence number (RSEQ): In no retry mode (i.e., when user messages are sent unreliably), this is 1 plus the byte sequence number of the most recently received user message (or end-of-message) byte. In retry mode (i.e., when user messages are sent reliably), this is 1 plus the byte sequence number of the most recently received user message (or end-of-message) byte without an earlier gap in the reliable data stream. Thus, in retry mode, RSEQ acknowledges all reliable user message (and end-of-message) bytes preceding the byte sequence number RSEQ.
12– 1 9		Reserved

Status (X′0E′) Segment

Byte	Bit	Content
20 – n		Acknowledged byte span pairs (ABSP): The remainder of the Status segment consists of the acknowledged byte span pairs. Each acknowledged byte span pair represents a sequence of reliable user message (and end-of-message) bytes held in this partner′s buffers pending arrival of the gaps. All implementations must support selective retransmissions and must be able to support at least 2 gaps (that is NABS ≥ 2). The number of ABSPs actually reported may depend on the buffer status of the receiver at the time the gap is detected. When the receiver′s buffers run low and it has already buffered one ABSP, the receiver may elect to discard subsequent data that does not reduce the size of the existing gap to avoid potential deadlock. Deadlock can occur when the receiver has used up all its available buffers to hold data in the ABSPs and none is left to receive data retransmitted to fill up the gap. The sender must retransmit the data within the gaps. On connections over a path with large bandwidth capacity and large propagation delay, keeping track of even one span beyond RSEQ can greatly improve the “link” efficiency.
(20+8(i−1)) – (23+8(i−1))		Acknowledged byte span pair begins (ABSPBEG): The byte sequence number of the first reliable user message (or end-of-message) byte in the *ith* acknowledged byte span.
(24+8(i−1)) – (27+8(i−1))		Acknowledged byte span pair ends (ABSPEND): This is 1 plus the byte sequence number of the last reliable user message (or end-of-message) byte in the *ith* acknowledged byte span. The receiver of this Status segment is no longer obligated to buffer the reliable user message (and end-of-message) bytes that are in this acknowledged byte span, from ABSPBEG to ABSPEND-1. Similar to the case of RSEQ described above, ABSPEND is the byte sequence number of the first byte of data that, upon arrival, will enlarge the *ith* acknowledged byte span. For example, if the current RSEQ is 10 and a message arrives with a BSN=21, DLF=10, SOMI, and EOMI, the receiver will report that a gap has been detected with NABS=1 and the acknowledged byte span pair contains ABSPBEG=21, ABSPEND=21+10+1=32. The sender is no longer obligated to buffer the bytes that are in an acknowledged byte span, from ABSPBEG to ABSPEND-1. Based on the above example, the sender will retransmit the message with BSN=10, DLF=10, SOMI, EOMI, and it can free the buffer that is used to hold the message with BSN=21, and DLF=10. A receiver can detect a gap upon receiving an empty message (i.e., a message with DLF=0, ¬ SOM, and ¬ EOMI) by checking the BSN field (the BSN field contains the byte sequence number of the first byte of a non-empty message to be sent by the sender), that is ,if the BSN field contains a sequence number that is greater than the RSEQ (taking into account the wrap-around) kept at the receiver. This empty message can be sent by the sender as a result doing a status exchange (i.e., a condition caused by the SHORT_REQ timer being timed out as a result of a loss). When this condition occurs, the receiver will report as usual that a gap has been detected with NABS=1, but the acknowledged byte span pair contains ABSPBEG=BSN and ABSPEND=BSN, since it is an empty message. This will allow the sender to transmit only the data within the gap instead of retransmitting everything starting from RSEQ (although the sender can continue to send data even after a time-out has been detected, and this data can arrive safely at the receiver). This is because without ABSP(s), the sender has no information as to what needs to be retransmitted and must resolve to a go-back N scheme, that is, retransmitting everything that has not been acknowledged.

Client Out-of-Band Bits (X′0F′) Segment

The Client Out-of-Band Bits segment is used to pass client signaling information.

Client Out-of-Band Bits (X′0F′) Segment

Byte	Bit	Content
0		Length/4 of the segment, including this Length field. See “Alignment and padding rules” on page 2-11.
1		Key=X′0F′
2 – n		Segment Data
2– 3 (= n)		Client bits: X′0001′ Request Deactivation X′8000′ Reply—OK X′8004′ Reply—Reject

Connection Identifier Exchange (X′10′) Segment

The Connection Identifier Exchange segment is sent to the partner to provide a TCID for it to use in all messages it sends on this RTP connection.

Connection Identifier Exchange (X′10′) Segment

Byte	Bit	Content
0		Length/4 of the segment, including this Length field. See “Alignment and padding rules” on page 2-11.
1		Key=X′10′
2 – n		Segment Data
2– 3		Reserved
4– 1 1 (= n)		Transport connection identifier (TCID) field:
	0	TCID assignor: always set to 1, indicating that the sender has chosen the TCID
	1	Reserved
	2– 6 3	TCID, identifying an RTP transport connection.

Connection Fault (X′12′) Segment

The Connection Fault segment is used to signal to the partner RTP end point that a connection fault has occurred.

Connection Fault (X′12′) Segment

Byte	Bit	Content
0		Length/4 of the segment, including this Length field. See “Alignment and padding rules” on page 2-11.
1		Key=X′12′
2 – n		Segment Data
2– 3		Reserved
4 – n		Sense data, indicating the reason for the fault

Switching Information (X′14′) Segment

The Switching Information segment is used in conjunction with the Connection Setup segment or to convey new path information to the partner on a path switch.

Switching Information (X′14′) Segment

Byte	Bit	Content
0		Length/4 of the segment, including this Length field. See “Alignment and padding rules” on page 2-11.
1		Key=X′14′
2 – n		Segment Data
2– 3		Reserved
4 – n		Switching information control vectors, included in LT form and in the order listed: X′83′ Switching Information control vector X′85′ Return Route TG Descriptor control vector

Switching Information (X′83′) Control Vector

The Switching Information control vector contains information about the path used by the RTP connection.

Switching Information (X′83′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′83′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – n		Vector Data
2	0	Resequencing (REFIFO) indicator: indicates whether or not the transport component (RTP) will, as part of normal operation (i.e. with no errors occurring), receive data traffic that arrives out of order. For example, traffic may arrive out of order because of a multilink TG along the path. If RTP expects traffic to be arriving out of order, it will allow sufficient time to receive missing (i.e., delayed) packets before asking the sender to resend them. 0 No—do not allow for normal operation resequencing. 1 Yes—allow for normal operation resequencing.
	1	Mobility indicator: Indicates whether or not the origin is mobile. This field is used in determining the time to allow for a path switch on this RTP connection. 0 The origin is not mobile (i.e., the origin is stationary). 1 The origin is mobile.

Switching Information (X′83′) Control Vector

Byte	Bit	Content
	2	Directory search required on path-switch indicator: This field is used when doing a path switch in order to determine whether a directory search is required. The origin node sets this bit to 1 if it is an EN or if the corresponding bit received in the Route Setup reply was 1; otherwise, this bit is set to 0. 0 directory search not required 1 directory search required
	3	Limited-resource link along the path indicator: indicates whether one or more limited-resource links exist along the path. 0 No limited resource links are along the path. 1 One or more limited resource links are along the path.
	4	NCE scope indicator: indicates whether the NCE associated with the LU (or BF link) is used for all LUs (or all BF links) in the origin node. The NCE is identified in the Connection Qualifier and/or Source Identifier Field (CQF) of the THDR when the Connection Setup segment is present. If the NCE is used for all LUs (or BF links) in the origin node, the destination node remembers the NCE identifier so that when establishing subsequent RTP connections to other LUs (or BF links) in the origin node, a Route Setup may not be required to get the NCE identifier. 0 The NCE is not used for all LUs (or BFs) in the origin node. 1 The NCE is used for all LUs (or BFs) in the origin node.
	5	MNPS RSCV retention indicator: When both users at the endpoints of an RTP connection are MNPS LUs, this bit indicates that the partner endpoint should treat the new RSCV and ANR data (for the new path) as the real route information, but to retain and use the old RSCV for putting new sessions on this RTP connection. 0 Use the new RSCV 1 Use the old RSCV
	6–7	Reserved
3		Reserved
4–7		Maximum packet size on the return path (in bytes): On RTP connections for CP-CP sessions and Route Setup flows, the value of this field is the minimum of the adjacent partner node′s maximum send packet size obtained on XID3 (in subfield X′81′ in control vector X′61′) and this node′s maximum receive packet size. On RTP connections for LU-LU sessions, the value of this field is obtained from the Reverse Route Information Maximum Packet Size field in the Route Setup reply GDS variable.
8–11		Path switch time: indicates the maximum time (in milliseconds) that the origin requires for a path switch. This time is used in conjunction with the destination′s path-switch time to determine the maximum allowed time for doing a path switch on this RTP connection. If a new path is not found within this time, the connection is terminated.
12–15		RTP ALIVE timer value (in seconds): For description of how this field is set and used see *HPR Architecture Reference*.
16 – n		ANR Path (X′67′) control vector for the reverse path: The NCE identifier for the reverse path is obtained from the Connection Qualifier field in the THDR.

Return Route TG Descriptor (X′85′) Control Vector

The Return Route TG Descriptor control vector contains the description of the reverse route in terms of TG numbers and CP names (as in an RSCV). It consists of a series of CV X′46′s (and all the associated X′8n′ subfields) that describe the route. The route described is one of the following:

- (Case 1) If the CV X′85′ is being sent along with a Connection Setup segment and the node sending the CV X′85′ does not contain the LU (i.e., the session continues over a boundary function link at the CV X′85′ sender node), the route described is from the node receiving the CV X′85′ to the node sending the CV X′85′ plus one additional entry for the BF link (the link between the HPR and APPN node) at the CV X′85′ sender node.
- In all cases other than Case 1, the route described is from the node receiving the CV X′85′ to the node sending the CV X′85′. No BF link is included. A BF link is never included when doing a path switch (a Connection Setup is not present when doing a path switch).

The route from the CV X′85′ receiver to the CV X′85′ sender is obtained from the RSCV returned on the Route Setup reply GDS variable. The series of CV X′46′s is copied directly from the Route Setup reply RSCV into the CV (X′85′) described below. If a BF link at the CV X′85′ sender node is required (see Case 1 above), a CV X′46′ entry for it is appended. The CV X′46′ entry for the BF link need not contain any subfields other than X′80′.

The CV X′85′ is used by the CV X′85′ receiver RTP end point to determine when this RTP connection may be used to carry additional (new) LU-LU sessions.

NOTE: The CV X′46′s do not follow the normal THDR CV alignment rules. They are not word aligned (except for the first one) and there are no padding bytes between them.

Return Route TG Descriptor (X′85′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′85′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – n		Vector Data
2	0	BF entry indicator: indicating whether a BF link for the CV X′85′ sender is present as the last CV X′46′ entry in bytes 4 – n: 0 BF entry not present 1 BF entry present
	1– 7	Reserved
3		Number of TG entries (CV X′46′s) contained in bytes 4 – n.
4 – n		A series of CV X′46′s, in LT format

Adaptive Rate-Based (X′22′) Segment

The Adaptive Rate-Based (ARB) segment is used to pass adaptive rate-based information between sending and receiving partner. The term ″forward″ is used to indicate the path in the direction that the ARB segment is being sent (i.e., from sender to receiver). The term ″reverse″ is used to indicate the path in the opposite direction (i.e., from receiver to sender).

Adaptive Rate-Based (X′22′) Segment

Byte	Bit	Content
0		Length/4 of the segment, including this Length field. See “Alignment and padding rules” on page 2-11.
1		Key=X′22′
2 – n		Segment Data

Adaptive Rate-Based (X′22′) Segment

Byte	Bit	Content
2	0–1	Message type: 00 Setup: When the ARB mode is Responsive (byte 2, bits 6-7 are B′01′) fields 1-4 are defined as follows. Field 1: Minimum receiver threshold (time in microseconds) Field 2: Maximum receiver threshold (time in microseconds) Field 3: Link capacity (Kbps) of slowest link along the path. If the ARB setup segment is sent along with a Switching Information segment, Field 3 is for the path in the reverse direction; otherwise, it′s for the forward direction. Field 4: Contains the total time, in microseconds, to transmit 1200 bits over the entire path. If the ARB setup segment is sent along with a Switching Information segment, Field 4 is for the path in the reverse direction; otherwise, it′s for the forward direction. When the ARB mode is Base (byte 2, bits 6-7 are B′00′) fields 1-4 are defined as follows. Field 1: Contains the time, in microseconds, to transmit a packet (NLP) containing 1000 bytes of data over the slowest link along the path in the forward direction. This value represents the beginning of the ARB operating range. Field 2: Contains the time, in microseconds, to transmit a packet containing either 10000, 15000, or 20000 bytes of data over the slowest link along the path in the forward direction. This value represents the end of the ARB operating range. (The receiver need not know whether it′s 10000, 15000, or 20000. It just uses the received value for the end of the ARB operating range.) Field 3: Contains the link capacity, in Kbps, of the slowest link along the path in the reverse direction. Field 4: Contains the total time, in microseconds, to transmit 1200 bits over the entire path in the reverse direction. This field is currently not used by the receiver in its execution of the ARB algorithm. It is included here for informational purposes such as for network management. 01 Rate Reply: A reply (sent by the receiver of an ARB(Rate Request) segment) that is used by the sender of the ARB(Rate Request) segment to adjust its send rate; fields 1-2 below contain the following information: Field 1: Reserved. Field 2: When the Base mode is being used, this field optionally contains the receiver′s receive rate, in Kbps, when the rate adjustment action (byte 2, bits 2–4) indicates either Slowdown 1 or Slowdown 2. Otherwise, this field is reserved. 10 Rate Request: The message type used by the sender of the ARB(Rate Request) segment to request ARB information from the receiver of the ARB(Rate Request) segment that will enable it to adjust its send rate; fields 1-2 below contain the following information: Field 1: Contains the sender′s measurement interval in microseconds. This interval is the time that has elapsed since the last rate request was sent. Field 2: Reserved. 11 Rate Request/Rate Reply: Both a rate request and a rate reply are contained in this segment; fields 1-2 below contain the following information: Field 1: See description of Field 1 under Rate Request. Field 2: See description of Field 2 under Rate Reply.

Adaptive Rate-Based (X′22′) Segment

Byte	Bit	Content
	2-4	Rate Adjustment Action: one of the following actions is returned in a Rate Reply message (i.e., when the Message Type field indicates Rate Reply or Rate Request/Rate Reply). 000 Normal: The sender may increase its send rate. 001 Restraint: The sender maintains its current send rate. 010 Slowdown 1: The sender reduces its send rate by 12.5%. 011 Slowdown 2: The sender reduces its send rate by 25%. 100 Critical: The sender reduces its send rate by 50%.
	5	Parity indicator - used in conjunction with the correlator fields (see byte 3) to determine if rate replies were successfully received by the rate request sender. This field is only meaningful when message type indicates rate request (i.e., byte 2, bits 0-1 are either B′10′ or B′11′). This field is not used in Base mode, only in Responsive mode.
	6– 7	ARB mode - indicates the mode of ARB supported by the ARB Setup segment sender. This field is defined as follows when the message type field indicates Setup (i.e., byte 2, bits 0-1 are set to B′00′); otherwise, it′s reserved. 00 Base mode ARB 01 Responsive mode ARB When the node sending the ARB Setup segment supports Responsive mode, it sets the ARB mode to Base if the partner supports Base (i.e., Base was previously received from the partner on either a Route Setup reply or an XID); otherwise, the ARB mode is set to Responsive. Note that the ARB Setup segment may be sent when the connection is established, a path switch is being done, or the link rate has changed. When the node receiving the ARB Setup segment supports Responsive mode, it uses the Base mode on the RTP connection if the partner supports Base (i.e., Base is received from the partner on the ARB Setup segment); otherwise, it uses the Responsive mode. Note that this logic is executed regardless of the condition (connection establishment, path switch, or link speed change) that caused the ARB Setup segment to be sent. It is possible that the ARB mode may change during the life of the connection. For example, a connection with a Multi-Node-Persistent-Session (MNPS) partner starts out running Responsive mode and the MNPS partner switches to a node that only supports Base (e.g., different nodes within a sysplex support different levels of VTAM). When this happens, a switch is made from Responsive to Base mode.
3	0– 3	Rate request correlator - used to uniquely identify a rate request. The first sent rate request carries a correlator of 1. The correlator is incremented by 1 for each subsequently sent rate request. When the correlator has a value of 15 (X′F′) and is incremented, it wraps backs to 0. This field is meaningful only when Responsive mode is being used; otherwise, it′s reserved.
	4– 7	Rate reply correlator - contains the value of the correlator received in the rate request being replied to. This field is meaningful only when Responsive mode is being used; otherwise, it′s reserved.
4-7		Field 1: See the Message Type field description above for this field′s contents.
8-11		Field 2: See the Message Type field description above for this field′s contents.
12-15		Field 3: See the Message Type field description above for this field′s contents.
16-19		Field 4: See the Message Type field description above for this field′s contents.

Note: Field 1 and Field 2 are always present, Field 3 and Field 4 are present only when the Message Type field indicates Setup (00).

End of Chapter 2

Chapter 3. Exchange Identification (XID) Information Fields

Introduction

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′ variable format (for T4|5 to T4|5 node exchanges): bytes 0 – p are included X′3′ variable format (for T2.1 to T2.1|4|5 node exchanges): bytes 0 – p are included X′8′ – X′F′ defined for external standards organizations
	4–7	Type of the XID-sending node: X′1′ T1 node X′2′ T2.0 or T2.1 node X′3′ reserved X′4′ T4 or T5 node
1		Length, in binary, of variable-format XID I-field (bytes 0 – p); reserved for fixed-format XID I-field
2–5|7		Node Identification
2–5	0–11	Block number: an IBM product-specific number; see the individual product specifications for the specific values used *Note:* The values of all 0′s and all 1′s indicate that bytes 2–5 do not contain a unique node identifier.
	12–31	ID number: a binary value that, together with the block number, identifies a specific station uniquely within a customer network installation; the ID number can be assigned in various ways, depending on the product; see the individual product specifications for details *Note 1:* When the Block Number field does not contain all 0′s or all 1′s, a value of all 0′s in the ID number indicates that no ID number has been assigned. *Note 2:* For XIDs flowing between T4|5 nodes, if the block number (bits 0–11) is all 0′s or all 1′s, the ID number (bits 12–31) may contain implementation-specific information. *Note 3:* For XID format 3, the contents of bytes 2–5 of the Node Identification field are used in some instances as a role-negotiation-value to resolve contention in protocol roles of nodes, e.g., primary/secondary DLC roles or the ODAI value to be appended to the (OAF′, DAF′) values assigned at a node. When a role-negotiation value is needed and the node does not supply a unique node identification value, it supplies a random value in the ID Number field.
End of Format 0		
6 – p		Format 1 Continuation

DLC XID Information Field

Byte	Bit	Content
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 mode status indicator (reserved if byte 9, bit 7 is set to 0): 0 sender not already engaged in a logical connection using short-hold mode 1 sender already engaged in a logical connection using short-hold mode
	7	Short-hold mode support indicator: 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	UA
SNRME	UA
Disconnect	DM
-	RD
-	Frame Reject
Reject	Reject

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

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

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

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

13	0– 1	Reserved
	2	SDLC initialization mode options: 0 SIM and RIM not supported 1 SIM and RIM supported
	3– 7	Reserved
14– 1 5		Reserved
16	0	Reserved
	1– 7	Maximum number of I-frames that can be received by the XID sender before an acknowledgment is sent, with an implied modulus for the send and receive sequence counts—less than 8 implies a modulus of 8; 8 or greater implies a modulus of 128
17		Reserved
For byte 9, bit 7 = 0 (short-hold mode not supported)		
18 – p		SDLC Address Assignment Field
18		Length (p minus 18), in binary, of the SDLC address to be assigned

DLC XID Information Field

Byte	Bit	Content
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 2 Continuation
6		Length of the XID exclusive of control vectors (n+1).
7		Miscellaneous flags:
	0	If byte 8, bit 1 is 1, “TG reorder not required” support indicator; otherwise, reserved: 0 This multiple-link TG does not support receipt of “TG reorder not required” PIUs. 1 This multiple-link TG does support receipt of “TG reorder not required” PIUs.
	1	PIU checksum support indicator: 0 PIU checksum is not supported on this link. 1 PIU checksum is supported on this link. *Note:* When both partners support PIU checksum, a two-byte checksum is generated for each PIU and carried in byte 8 and byte 12 of the FID4 or FIDF TH.
	2– 7	Reserved

DLC XID Information Field

Byte	Bit	Content
8	0	TG status: 0 TG inactive 1 TG active
	1	Multiple-link TG support: 0 multiple-link TG not supported 1 multiple-link TG supported
	2–3	Segment assembly capability of the path control element of the node: 00 segments are ignored and passed through 01 segments are assembled on a link station basis 10 segments are assembled on a session basis 11 segments are not allowed
	4	Multiple PIU frame capability: 0 multiple PIU frame not supported 1 multiple PIU frame support
	5	FID4 TGSF support: 0 segmentation not supported on this TG 1 segmentation supported on this TG
	6–7	Reserved
9		FID types supported:
	0	0 FID0 not supported 1 FID0 supported
	1	0 FID1 not supported 1 FID1 supported *Note:* Neither bit 0 nor bit 1 is set to 1 when XID Format 2 is exchanged, but can be set by PU.SVC_MGR when the contents of XID Format 2 is carried in the CONTACTED RU.
	2–3	Reserved
	4	0 FID4 not supported 1 FID4 supported
	5–7	Reserved
10	0	Upper-layer protocol use: 0 subarea SNA protocols used on the link 1 non-SNA protocols used on the link
	1–7	Reserved
11–12		Length, in binary, of maximum PIU that the XID sender can receive *Note:* The secondary link station sets bytes 11–12 to the value defined by implementation- and installation-specific parameters. If the value of bytes 11–12 in a received XID is less than the value being sent, the secondary sets byte 18, bits 1–4 to X′8′ (incompatible parameters).
13		Transmission group number (TGN)
14–17		Subarea address of the XID sender (right-justified with leading 0′s)

DLC XID Information Field

Byte	Bit	Content
18	0	Reserved
	1– 4	Error status (set in reply to a previously received XID): 0000 no error 1000 Exchanged parameters in the XIDs are not compatible. (*Note:* This value must be set when bit 5, Switched Subarea Support, is set to 1.) 1001 Incompatible parameters in the XID were received for addition of the link station to a currently active multiple-link TG (e.g., maximum PIU length). 1010 TG is not defined (i.e., no routing found). 1100 The specified TG between the subarea nodes exchanging XIDs is already active on one or more other links, but multiple-link TG support (byte 8, bit 1) is not available or the DLC type (byte 30) is incompatible with the already active TG.
	5	Switched subarea support: 0 call security verification successful, if applicable; if not, reserved 1 call security verification failed *Note:* When bit 5 is set to 1, bits 1– 4 must indicate exchanged parameters in the XIDs are not compatible.
	6– 7	Reserved
19		CONTACT or load status of XID sender: X′00′ CONTACT has been received by an XID command sender. X′07′ XID response sender is already loaded. X′09′ Load is required.
20– 27		IPL load module name: an 8-character EBCDIC symbolic name of the IPL load module of the XID sender *Note:* X′40...40′ = no information conveyed.
28		Extended Subarea Address support:
	0	0 Extended Subarea Address not supported 1 Extended Subarea Address supported
	1– 3	Reserved
	4– 7	Extended Subarea Address address limit: 0000 Subarea address limit = 255 0001 Subarea address limit = 511 0010 Subarea address limit = 1023 0011 Subarea address limit = 2047 0100 Subarea address limit = 4095 0101 Subarea address limit = 8191 0110 Subarea address limit = 16383 0111 Subarea address limit = 32767 1000 Subarea address limit = 65535
29		Reserved
30		DLC type: X′01′ SDLC X′02′ System/390 channel (System/390 to communication controller) X′03′ System/390 channel (System/390 to System/390) X′04′ Multipath channel to channel, write path from sender (System/390 to System/390) X′05′ Multipath channel to channel, read path from sender (System/390 to System/390)
31 – n		DLC-dependent parameters

For SDLC

DLC XID Information Field

Byte	Bit	Content
31	0	Reserved
	1–3	Link-station role of XID sender: 1 0 XID sender cannot be an ABM combined station 1 XID sender can be an ABM combined station 2 0 XID sender cannot be secondary 1 XID sender can be secondary 3 0 XID sender cannot be primary 1 XID sender can be primary *Note:* A combination of 000 in bits 1–3 is reserved.
	4–5	Reserved
	6–7	Link station transmit-receive capability: 00 two-way alternating 01 two-way simultaneous
32–33		Maximum I-field length, in binary, that the XID sender can receive
34	0–3	Reserved
	4–7	SDLC command/response profile: X′0′ SNA link profile (only value defined) *Note:* These profiles refer to the mandatory command/response support on an SDLC link, as follows: • For an SDLC link in normal response mode (NRM/NRME), having a point-to-point or multipoint configuration (determined from system definition), the support required is: **Commands** / **Responses** I-frames / I-frames RR / RR RNR / RNR Test / Test XID / XID SNRM / UA SNRME / UA Disconnect / DM - / RD - / Frame Reject Reject / Reject *Note 1:* The RD response is sent by the secondary station if and only if CS has decided to deactivate the link. *Note 2:* Reject is required only if both sender and receiver have two-way simultaneous transmit-receive capability.

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 CS has decided to deactivate the link.

- 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.

35 0 Net ID processing capability on a nonswitched link connection:
0 sending node not capable of processing net ID on a nonswitched link connection
1 sending node capable of processing net ID on a nonswitched link connection

DLC XID Information Field

Byte	Bit	Content
	1	Reserved
	2–3	SDLC initialization mode options:
	2	0 XID sender cannot send SIM nor receive RIM. 1 XID sender can send SIM and receive RIM.
	3	0 XID sender cannot receive SIM nor send RIM. 1 XID sender can receive SIM and send RIM.
	4	0 echo defeat not supported 1 echo defeat supported
	5	Short-hold mode status (reserved if byte 35, bit 6 is set to 0): 0 sender not already engaged in a logical connection using short-hold mode 1 sender already engaged in a logical connection using short-hold mode
	6	Short-hold mode capability of the XID sender: 0 short-hold mode not supported 1 short-hold mode supported
	7	Prenegotiation status: 0 This is not a prenegotiation XID. 1 This is a prenegotiation XID. *Note:* This is typically used for pre-CONTACT identification exchanges on switched link connections.
36	0	Net ID qualifier: 0 The appended net ID cannot be changed. 1 The appended net ID is a default and can be changed.
	1–7	Reserved
37		Reserved
38	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
39–43 (=n)		Reserved
n+1 – p		One or more control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT. X′0E′ Network Name control vector: type X′F1′, PU name (present when the type 4 and 5 nodes have APPN capability). X′0E′ Network Name control vector: type X′F4′, network-qualified CP name (present when the type 4 and 5 nodes have APPN capability; the network identifier is always used; i.e., valid lengths of the network-qualified CP name are 3 to 17 bytes with an imbedded period separating the network identifier and CP name parts of the field) X′0E′ Network Name control vector: type X′F7′, local name of the ALS at the XID sender (present when the type 4 and 5 nodes have APPN capability). X′10′ Product Set ID control vector (always included) X′12′ Network Identifier control vector X′32′ Short-Hold Mode control vector (conditionally present) X′56′ Call Security Verification control vector (conditionally present)
For System/390 channel (System/390 to communication controller)		*Note:* The System/390 node always contains the primary link station for the System/390 channel (System/390 to communication controller) DLC.

DLC XID Information Field

Byte	Bit	Content
31		Number of buffers suggested by the primary link station for the secondary link station to obtain each time the secondary must obtain buffers for receiving data from the primary (primary sets and secondary echoes unless X′00′) *Note:* X′00′ = no suggestion made. If byte 31 = X′00′ in the XID received, secondary uses a value defined by optional implementation- and installation-specific parameters and sends it to the primary *Note:* The size of these buffers is not carried in the XID.
32–33		Number of Read commands a primary link station will provide to a secondary link station within Read Start channel programs; reading of each PIU begins with a new Read command; the number of unacknowledged PIUs will be less than or equal to the number of Read commands, depending on whether a PIU requires more than one Read command for transfer; this number defines the maximum number of Read commands needed by the link stations to recover from an error *Note:* See Note after bytes 34–35.
34–35		Number of bytes allocated by a primary link station per Read command (see bytes 32–33) *Note:* The secondary link station sets bytes 11–12 to the value defined by implementation- and installation-specific parameters. If the value of bytes 11–12 in a received XID is less than the value being sent, the secondary sets byte 18, bits 1–4 to X′8′ (incompatible parameters). If byte 18, bits 1–4 are set to a nonzero value, bytes 32–36 are set to the secondary′s implementation- and installation-specific values. If byte 18, bits 1–4 are set to X′0′ by the secondary, the secondary uses and echoes the values of bytes 32–33, 34–35, and 36 received from the primary in XID.
36		Number of pad bytes a secondary transmits immediately preceding each PIU sent to the primary *Note:* The first pad byte is an eight-bit binary count of pad bytes; the remaining pad values are unpredictable. See also the Note following bytes 34–35.
37	0	Reserved for primary; for secondary: 0 secondary does not use the status modifier option for data transfer to primary 1 secondary uses the status modifier option for data transfer to primary
	1	Reserved
	2	Contact option. Reserved for secondary; set by primary (with secondary requested to take the following action): 0 if the TG specified in this XID is in contacted state on another System/390 channel, the secondary is to send an XID response with X′C′ in the Error Status field of byte 18 1 if the TG specified in this XID is in contacted state on another System/390 channel, this latter channel is discontacted and the XID is accepted on the System/390 channel now being contacted
	3–7	Reserved
38–39		Reserved for primary; for secondary: the maximum interval (in tenths of a second) that the secondary delays between the time it has a PIU for the primary and the time it presents an Attention signal to the primary
40–41 (=n)		Reserved for primary; for secondary: the maximum interval (in tenths of a second) that the secondary awaits a response to an Attention signal that has been sent to the primary before initiating inoperative link processing

DLC XID Information Field

Byte	Bit	Content
n + 1 – p		One or more control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT. X'0E' Network Name control vector: type X'F1', PU name (present when the type 4 and 5 nodes have APPN capability). X'0E' Network Name control vector: type X'F4', network-qualified CP name (present when the type 4 and 5 nodes have APPN capability; the network identifier is always used; i.e., valid lengths of the network-qualified CP name are 3 to 17 bytes with an imbedded period separating the network identifier and CP name parts of the field) X'0E' Network Name control vector: type X'F7', local name of the ALS at the XID sender (present when the type 4 and 5 nodes have APPN capability). X'10' Product Set ID control vector (always included) X'12' Network Identifier control vector X'32' Short-Hold Mode control vector (conditionally present) X'56' Call Security Verification control vector (conditionally present)
For System/390 channel (System/390 to System/390)		
31–32		Number of buffers the XID sender will normally allocate to receive data
33–34		Maximum number of buffers the XID sender will allocate to receive data when signaled by the data sender that the maximum is required (This value times the value in bytes 35–36 equals the value in bytes 11–12: the maximum size PIU that the XID sender can receive.)
35–36 (=n)		Number of bytes in each buffer (see bytes 31–34) allocated by the XID sender to receive data on the channel
n + 1 – p		One or more control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT. X'0E' Network Name control vector: type X'F1', PU name (present when the type 4 and 5 nodes have APPN capability). X'0E' Network Name control vector: type X'F4', network-qualified CP name (present when the type 4 and 5 nodes have APPN capability; the network identifier is always used; i.e., valid lengths of the network-qualified CP name are 3 to 17 bytes with an imbedded period separating the network identifier and CP name parts of the field) X'0E' Network Name control vector: type X'F7', local name of the ALS at the XID sender (present when the type 4 and 5 nodes have APPN capability). X'10' Product Set ID control vector (always included) X'12' Network Identifier control vector X'32' Short-Hold Mode control vector (conditionally present) X'56' Call Security Verification control vector (conditionally present)
End of Format 2		
6 – p		Format 3 Continuation **Note:** For HPR, some of the following fields pertaining to RUs are relevant to the FID2 PIU flows on the link, but not to the NLP flows. This is because either the RU in question does not flow in NLPs, or the necessary support for their control is provided by RTP protocols for the NLP flows. These fields, which apply solely to the FID2 PIU flows, are flagged by an asterisk [*****] in parentheses after the field name. All fields not marked in this fashion are relevant to both flows.)
6–7		Reserved

DLC XID Information Field

Byte	Bit	Content
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 (or character-coded logon) cannot be sent to the XID sender.
		Note: For bits 0–1, the value 11 is reserved.
	1	Stand-alone BIND support (*):
		0 BIND may be sent to the XID sender without a prior INITIATE sequence (i.e., XID sender supports independent-PLU session partners).
		1 BIND may not be sent to the XID sender (i.e., the XID sender does not support independent-PLU session partners).
		Note: For bits 0–1, the value 11 is reserved.
	2	Whole-BIND-PIUs generated indicator (*):
		0 This node can generate BIND PIU segments.
		1 This node does not generate BIND PIU segments.
	3	Whole-BIND-PIUs required indicator (*):
		0 This node can receive BIND PIU segments.
		1 This node cannot receive BIND PIU segments.
		Note: The value 10 for bits 2–3 is reserved.
	4–7	Retired
	8	ACTPU suppression indicator (*):
		0 ACTPU for an SSCP-PU session requested
		1 ACTPU for an SSCP-PU session not requested
	9	Networking capabilities indicator:
		0 The sender is not a network node.
		1 The sender is a network node.
	10	Control point services (reserved when bytes 8–9, bit 11 is 0):
		0 CP services not requested or provided
		1 CP services requested or provided: when network services are not provided on this TG by the XID sender (bit 9 = 0), CP services are requested; when network services are provided on this TG by the XID sender (bit 9 = 1), CP services are provided.
	11	CP-CP session support:
		0 CP-CP sessions not supported on this TG
		1 CP-CP sessions supported on this TG
		Note: The following combinations of bits 9, 10, and 11 are not valid: 010, 101, 110.
	12–13	XID exchange state indicators:
		00 exchange state indicators not supported (set only by implementations not at the current level of SNA)
		01 negotiation-proceeding exchange
		10 prenegotiation exchange — the only other fields considered meaningful in this XID are:
		• Node Identification field • All control vectors • Networking Capabilities indicator, unless the APPN Peripheral Border Node indicator or the control vector X′46′ Extended Border Node indicator indicates the sender is a border node on this TG • Control Point Services and CP-CP Session Support, if the Branch indicators are set to 01

DLC XID Information Field

Byte	Bit	Content
		11 nonactivation exchange — the XID fields that can change during a nonactivation exchange are: • Transmission Group Number • Network Name (X′0E′, CP name) control vector • Quiesce TG Request • Control Point Services • CP-CP Sessions Support • CP Name Change Support
	14	Nonactivation exchange secondary-initiated capability: 0 nonactivation exchange initiated by secondary station not supported 1 nonactivation exchange initiated by secondary station supported (only value used by current-level APPN nodes)
	15	CP name change support indicator: 0 The sender will fulfill nonactivation XID exchange protocols but, except for the Exchange State indicators, is not able to process fields in the received XID3 that differ from those sent in during the previous XID3 exchange. 1 The sender can process nonactivation XID3s that contain a CP name or TG number that differs from that received by the sending node during the last XID exchange.
10	0–1	BIND pacing support over the TG (**Note**: See the Qualifier for Adaptive BIND Pacing Support field [byte 10, bits 6–7] for this field′s relevance to dependent and independent LUs.)
	0	Adaptive BIND pacing support as a BIND sender (*****): 0 adaptive BIND pacing as a BIND sender not supported 1 adaptive BIND pacing as a BIND sender supported
	1	Adaptive BIND pacing support as a BIND receiver (*****): 0 adaptive BIND pacing as a BIND receiver not supported 1 adaptive BIND pacing as a BIND receiver supported *Note:* The combinations of bits 0 and 1 have the following meanings: 00 means adaptive BIND pacing is not supported; 01 means one-way adaptive BIND pacing is supported; 10 is invalid; 11 means adaptive BIND pacing is fully supported.
	2	Quiesce TG request indicator: 0 The sender requests that the receiving node generate a topology update stating that the TG from the receiver to the sender is operative. 1 The sender is requesting that the receiving node generate a topology update stating that the TG from the receiver to the sender is quiesced. *Note:* The requested topology update is generated only if the value sent in this field differs from that sent in the previous XID exchange.
	3	PU capabilities support (*****): 0 does not support receipt of ACTPU containing a PU Capabilities (X′80′) control vector 1 supports receipt of ACTPU containing a PU Capabilities (X′80′) control vector
	4	APPN peripheral border node (PBN) indicator: 0 Sending node is not a PBN. 1 Sending node is a PBN.
	5	Reserved
	6–7	Qualifier for Adaptive BIND pacing support (*****): 00 Adaptive BIND pacing support applies to BINDs for BOTH independent and dependent LUs, and is nonnegotiable. 01 Adaptive BIND pacing support applies to BINDs for BOTH independent and dependent LUs, unless overridden by the partner node. 10 reserved 11 (Retired) Adaptive BIND pacing support applies to BINDs only for independent LUs. (**Note**: Nodes using this setting cannot be connected to those using the 00 setting.)
11	0	Reserved

DLC XID Information Field

Byte	Bit	Content
	1	Defined TG sharing prohibited indicator: 0 New connection network traffic may share this defined TG (and the virtual connection over which it was established). 1 New connection network traffic may not share this defined TG.
	2	Dedicated SVC indicator: 0 This SVC may be shared by multiple RTP connections. 1 This SVC may be used by only one RTP connection.
	3– 7	Reserved
12	0	Negotiation complete supported indicator - indicates whether or not the negotiation complete indicator (byte 12, bit 1) is supported. This field is meaningful when the XID exchange state is negotiation proceeding; otherwise, it is reserved. 0 the negotiation complete indicator is not supported 1 the negotiation complete indicator is supported
	1	Negotiation complete indicator - this field is meaningful only when the XID exchange state is negotiation proceeding, the XID is being sent from the secondary to the primary, and the negotiation complete supported indicator (byte 12, bit 0) is set to 1; otherwise, this field is reserved. This field is set to 1 and sent by a secondary that supports this function when it considers XID negotiation to be complete (i.e., it is ready to receive a ″setmode″ command from the primary). When this field is set to 1 and is received by a primary that supports this function, the primary will know that it can send the ″setmode″ command (SNRM, SABME, etc.) if it also considers XID negotiation to be complete. 0 XID negotiation (e.g., role and TG number) is not complete 1 XID negotiation is complete
	2-7	Reserved
13– 1 4		Reserved
15	0	Parallel TG support indicator: 0 parallel TGs not supported, only a single TG between the sender and the receiver is permitted 1 parallel TGs are supported, more than one TG between the sender and the receiver may be activated
	1	Dependent LU requester (DLUR) ACTPU indicator (reserved if the ACTPU Suppression indicator is set to 1, or if the sender is not a DLUR node): 0 DLUR XID sender has no preference on whether ACTPU is received on the TG dependent flow or encapsulated over the CP-SVR pipe. 1 DLUR XID sender prefers receiving ACTPU over the CP-SVR pipe.
	2	DLUS-served LU registration indicator (reserved if the sender is not an NN): 0 DLUS-served LU registration not supported 1 DLUS-served LU registration supported
	3	Extended HPR border node indicator: 0 This node is not an extended HPR border node. 1 This node is an extended HPR border node.
	4	Generalized ODAI Usage option set indicator: 0 Generalized ODAI Usage option set is not supported. 1 Generalized ODAI Usage option set is supported.

DLC XID Information Field

Byte	Bit	Content
	5–6	Branch indicators: 00 The XID sender does not support option set 1121 (Branch Extender); the TG is neither a branch downlink nor a branch uplink. 01 The XID sender defines this TG as a branch uplink. 10 The XID sender defines this TG as a branch downlink. 11 The XID sender supports option set 1121 but defines this TG as neither an uplink nor a downlink.
	7	End Node Resource Registration with different owning CP name NNS(BrNN) support (option set 1123) indicator: 0 This node does not support option set 1123. 1 This node supports option set 1123.
16		Transmission group number: a binary value in the range 0 to 255
17		DLC type: X′01′ non-channel DLC (e.g., SDLC, token-ring, Ethernet, frame-relay) X′02′ System/390 channel to controller DLC X′06′ APPN host-to-host channel
18 – n		DLC Dependent Section
18		Length, in binary, of the DLC Dependent Section field
For Non-channel DLC		
19		Link Station and Connection Protocol flags
19	0	Reserved
	1	ABM support indicator: 0 XID sender is not using ABM on this link. 1 XID sender is using ABM on this link.
	2–3	Link-station role of XID sender: 00 secondary link station (nonnegotiable) 01 primary link station (nonnegotiable) 10 reserved 11 negotiable link station (primary or secondary capability) *Note:* For ABM stations, the value of bits 2–3 is used only for the purposes of OAF′-DAF′ assignment and deciding which node sends the Set Mode command.
	4	Short-hold mode status indicator (reserved if byte 19, bit 5 is set to 0): 0 sender not already engaged in a logical connection using short-hold mode 1 sender already engaged in a logical connection using short-hold mode
	5	Short-hold mode indicator: 0 short-hold mode not supported 1 short-hold mode supported
	6–7	Link-station transmit-receive capability: 00 two-way alternating 01 two-way simultaneous
20	0	ABM nonactivation XID exchange initiator indicator: 0 XID sender is not the initiator of a nonactivation XID exchange on an ABM TG 1 XID sender is the initiator of a nonactivation XID exchange on an ABM TG XID command *Note:* Support for the ABM Nonactivation XID Initiator indicator is required for all ABM link stations that also support secondary-initiated nonactivation XID exchanges.
	1–7	Reserved
21–22		Maximum BTU length that the XID sender can receive:
	0	Format flag: 0 bits 1–15 contain the maximum BTU length (only value defined)

DLC XID Information Field

Byte	Bit	Content
	1–15	Maximum BTU length, in binary: when HPR Capabilities (X′61′) control vector is present, this value must be 768 or greater — if not, the link activation is rejected with control vector X′22′ (carrying sense data X′10160022′)
23	0–3	Reserved
	4–7	Retired (set to 0′s)
24	0–1	Reserved
	2	DLC 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 (i.e., the receive window), with an implied modulus on NRM connections for the send and receive sequence counts — less than 8 implies a modulus of 8; 8 or greater implies a modulus of 128. **(This field is reserved if I-frames are not used on this TG.)** The value received in this field is the maximum number of I-frames that the XID receiver may send and have unacknowledged at any given time, i.e., the maximum size of the send window. (This value is not the system parameter, N3 [as described in the *Token-Ring Architecture Reference*, SC30-3374], that IBM has implemented for IEEE 802.2 Type 2 operation; typically, N3 is half the size of this receive window.) *Note:* ABM connections are associated with IEEE 802.2 LLC Type 2 operation, which may be over a local-area network or a frame-relay connection. The settings of certain system parameters depend upon the underlying transmission medium.
28(=n)		Reserved
For Channel DLC (System/390 Channel to Controller)		
		Note: The System/390 node always contains the primary link station for channel data link control (CDLC); the controller always contains the secondary station.
19–20		Indicators:
	0	Change CDLC parameters; may be set by the primary on a nonactivation XID; echoed by the secondary; reserved for both primary and secondary for other XID exchange types: 0 do not change CDLC parameters 1 change CDLC parameters to the values in this XID; the parameters that may be changed are buffer prefetch, number of read commands, buffer size, blocking delay, attention timeout, and time units
	1	Attention timeout support; set by the secondary; reserved for the primary: 0 not supported 1 supported
	2	Data streaming support indicator: 0 not supported 1 supported
	3	Change CDLC parameters support; specifies whether the XID sender supports changing CDLC parameters by means of a nonactivation XID exchange (see bytes 19–20, bit 0): 0 not supported 1 supported
	4–15	Reserved
21–22		Length, in binary, of the maximum link PIU (LPIU) that the XID sender can receive
23		Buffer prefetch: number of buffers suggested for the secondary to preallocate each time the secondary reads LPIUs from the primary

DLC XID Information Field

Byte	Bit	Content
24–25		Number of Read commands: number of Read CCWs the primary must include in every read channel program used to read LPIUs
26–27		Buffer size: for the primary, the size of the input area associated with each Read CCW in channel programs used to read LPIUs; for the secondary, the approximate number of bytes available for LPIU storage in each buffer used for accepting LPIUs from the primary
28–29		Blocking delay: maximum interval that the secondary delays between the time it has an LPIU to send to the primary and the time it presents an Attention to the primary
30–31		Attention timeout: maximum interval that a secondary awaits a read channel program after presenting an Attention to the primary; if the timeout expires, a secondary-detected inoperative station condition is declared. This timeout value is also used for idle detection (1/2 Attention timeout [ATO] is used), second-chance Attention (1/2 ATO is used), and primary-detected inoperative station (3/2 ATO is used)
32–33		Previous number of Read commands: set by the secondary in an XID sent in reply to a change-CDLC-parameters nonactivation XID; otherwise, reserved. The field contains the value of the number-of-Read-commands parameter that was active prior to the change.
34–35		Previous primary buffer size: set by the secondary in an XID sent in reply to a change-CDLC-parameters nonactivation XID; otherwise, reserved. The field contains the value of the primary-buffer-size parameter that was active prior to the change.
36(=n)		Time units: specifies the time units used for Attention timeout and blocking delay X′00′ 100-millisecond time units X′01′ 1-millisecond time units
For APPN Host-to-Host Channel		
19		Connection protocol flags:
	0–1	Reserved
	2–3	Role of XID sender: 00 Sender is secondary (nonnegotiable). 01 Sender is primary (nonnegotiable). 10 reserved 11 Sender role is negotiable (primary or secondary capability).
	4–7	Reserved
20		Reserved
21–22 (=n)		Maximum BTU size, in binary, the XID sender can receive
n+1 – p		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′0E′ Network Name control vector: type X′F1′, PU name (present only from a T4|5 node XID sender) X′0E′ Network Name control vector: type X′F4′, network-qualified CP name (always present; the network identifier is always used; i.e., valid lengths of the network-qualified CP name are 3 to 17 bytes with an imbedded period separating the network identifier and CP name parts of the field) X′0E′ Network Name control vector: type X′F7′, local name of the ALS at the XID sender (present when the sending node provides a nonnegotiated representation of the link in addition to the negotiated TG number)

DLC XID Information Field

Byte	Bit	Content	
		X′10′	Product Set ID control vector (always present when the Exchange State indicators are supported) *Note:* When included in XID, the Product Set ID control vector is limited to a maximum of 60 bytes.
		X′22′	XID Negotiation Error control vector (present when an error during XID negotiation is detected; more than one may be present)
		X′32′	Short-Hold Mode control vector (always present when the XID pertains to short-hold mode and the value of the Exchange State indicators is not “prenegotiation exchange”)
		X′46′	TG Descriptor control vector (always present when the sending node is a branch network node, the sending node is activating a defined TG over a switched virtual circuit [e.g., an ATM SVC], the sending node is activating a defined TG over an IP network, the sending node is activating a TG through a virtual routing node, or a border node is activating an intersubnetwork TG; optionally present to identify the TG number when the sending node is activating a predefined TG or is reactivating a TG with the TG number used in the previous activation)
		X′57′	DLC Connection Data control vector (present only from a T4|5 node XID sender for channel-attached IP hosts)
		X′61′	HPR Capabilities control vector (present on a negotiation-proceeding or nonactivation XID when the sender wants to use HPR protocols over the link)

End of Chapter 3

Chapter 4. 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. The following TH formats, or FID types, are defined: FID0, FID1, FID2, FID3, FID4, FID5, and FIDF; they correspond to hexadecimal values 0–5, and F, respectively, in the FID field. All undefined FID values are reserved.[1]

The different THs, according to FID type, are defined below.

FID Types: Field Layouts and Descriptions

FID0 and FID1 Layout

Byte		
0	FID0|1—Format Identification MPF—Mapping Field Reserved Bit EFI—Expedited Flow Ind.	Reserved Byte
2	DAF—Destination Address Field	
4	OAF—Origin Address Field	
6	SNF—Sequence Number Field	
8	DCF—Data Count Field	

Figure 4-1. Transmission Header for FID Types 0 and 1

FID0 and FID1 Field Descriptions

These formats are used between adjacent subarea nodes when either or both nodes do not support ER and VR protocols.

FID0 is used for non-SNA device traffic, and FID1 is used for SNA traffic. Except for the FID field value, the TH fields for FID0 and FID1 are identical.

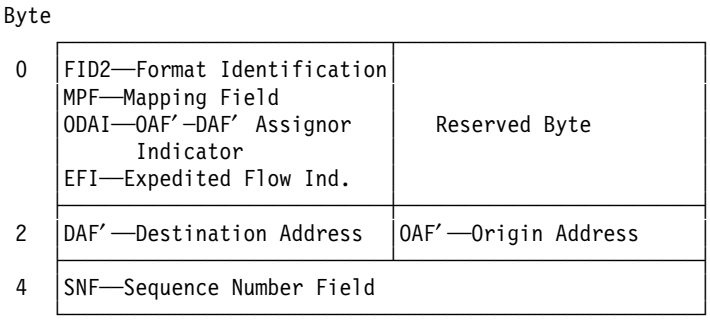
Nodes that support ER and VR protocols provide conversion between FID4 THs and FID0 and FID1 THs. FID4 THs to be sent to nodes not supporting ER and VR protocols are converted to either FID0 or FID1 THs, as determined by the FID4 TH SNA indicator. FID0 and FID1 THs received from nodes not supporting ER and VR protocols are converted to FID4 THs, with the FID4 SNA indicator set to ¬ SNA or SNA, respectively.

[1] 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.

FID0 and FID1 Field Descriptions

Byte	Bit	Content
0	0– 3	FID0|1—Format Identification: 0000 FID0 0001 FID1
	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)
	6	Reserved
	7	EFI—Expedited Flow Indicator. It has the following meaning: 0 normal flow 1 expedited flow *Note:* 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.
1		Reserved
2– 3		DAF—Destination Address Field, a two-byte network address denoting the BIU′s destination network addressable unit (NAU). The DAF provides the principal routing information needed by PC. In a network address the subarea address 0 is reserved; the element address 0 always denotes the PU T4|5 generating the associated subarea.
4– 5		OAF—Origin Address Field, a two-byte network address denoting the originating NAU. The OAF allows multiple active half-sessions per NAU by distinguishing the origins of all PIUs received by the NAU.
6– 7		SNF—Sequence Number Field, a two-byte numerical identifier for the associated BIU.
8– 9		DCF—Data Count Field, a binary count of the number of bytes in the BIU or BIU segment associated with the transmission header; the count does not include any of the bytes in the transmission header. The DCFs are required in PIUs that are to be blocked, as they convey the PIU length information necessary for proper deblocking.

FID2 Layout



Note: FID2 PIUs cannot be blocked because there is no DCF in the TH format for deblocking.

Figure 4-2. Transmission Header for FID Type 2

FID2 Field Descriptions

The FID2 format is used between a T4 or T5 subarea node and an adjacent T2 (i.e., T2.0 or T2.1) peripheral node, or between adjacent APPN or LEN 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.
	6	ODAI—OAF′-DAF′ Assignor Indicator (used for LEN or APPN flows; otherwise, reserved). The ODAI indicates which node assigned (at session-activation time) the OAF′-DAF′ values carried in the TH (see *SNA APPN Architecture Reference* for details). Together with the DAF′ and OAF′ values, the ODAI value forms a 17-bit local-form session identifier (LFSID); the DAF′ and OAF′ values used in the TH in one direction are reversed in the other direction. *Note:* See Chapter 5, “Request/Response Headers (RHs)” for the discussion of the adaptive BIND pacing IPM′s use of these fields. See also Chapter 6, “Request/Response Units (RUs)” for the discussion of HPR′s ROUTE SETUP PIU′s usage of these fields.
	7	EFI—Expedited Flow Indicator. The EFI designates whether the PIU belongs to the normal or expedited flow. Normal-flow PIUs are kept in order on a session basis by PC; so are expedited-flow PIUs. Expedited-flow PIUs can pass normal-flow PIUs flowing in the same direction at queuing points in TC within half-sessions and boundary function and APPN session connectors. It has the following meaning: 0 normal flow 1 expedited flow
1		Reserved

FID2 Field Descriptions

Byte	Bit	Content
2		DAF′—Destination Address Field. See discussion above for ODAI.
3		OAF′—Origin Address Field. See discussion above for ODAI. *Note:* For T2.0 peripheral nodes, 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 APPN or LEN nodes, an OAF′ or DAF′ can also be set to 0 for independent LU-LU sessions (see *SNA APPN Architecture Reference* for details).
4– 5		SNF—Sequence Number Field. The Sequence Number Field contains a numerical identifier for the associated BIU; path control, when segmenting, puts the same SNF value in each segment derived from the same BIU. The numerical identifier used depends on a number of factors. If the TS profile indicates sequence numbers are not used, the SNF value is a 16-bit identifier that distinguishes a request being sent or responded to from any other outstanding request on the same flow. If the TS profile indicates sequence numbers are used, the flow is a factor. Expedited-flow requests (other than SIG for LU 6.2) carry 16-bit identifiers; expedited-flow responses echo the SNF values of their corresponding requests. Normal-flow requests, other than between LU 6.2s, carry 16-bit numerical values ranging in value from 1–65,535 (incremented by 1 for each request) and wrapping through 0 thereafter; the corresponding responses echo their SNF values. The table below defines the SIG and normal-flow SNF usage between LU 6.2s.

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

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

B: Low-order 15 bits of the SNF in the request that carried the last successful BB; the high-order bit identifies the half-session that started the bracket (0 = secondary, 1 = primary); in the case of the first bracket of a session, where the BB is implied, not sent, the low-order 15 bits are 0 and the high-order bit is 1.

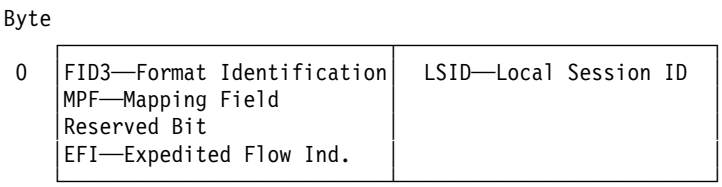
C: Low-order 15 bits of the SNF in the BB request being responded to; the high-order bit identifies the sender of the BB request (0 = secondary, 1 = primary).

D: The half-session does not respond to BIS.

E: Same value as the corresponding request.

Note: For additional details of LU 6.2 processing, see *SNA LU 6.2 Reference: Peer Protocols*.

FID3 Layout (Retired)



Note: FID3 PIUs cannot be blocked because there is no DCF in the TH format for deblocking.

Figure 4-3. Transmission Header for FID Type 3

FID3 Field Descriptions

(Retired header) The FID3 format is used between a T4 node and an adjacent T1 node or between a T5 node and an adjacent T1 node.

FID3 Field Descriptions

Byte	Bit	Content
0	0–3	FID3—Format Identification: 0011
	4–5	MPF—Mapping Field. Described earlier.
	6	Reserved
	7	EFI—Expedited Flow Indicator. Described earlier.
1		LSID—Local Session Identification. In FID3, the DAF and OAF are replaced by a single byte, the LSID, which provides a limited DAF/OAF capability. The LSID consists of three parts: an LU/SSCP indicator (bit 0), an LU/PU indicator (bit 1), and a local address (bits 2–7). Each T1 node can support up to 64 secondary LUs; each LU is known, local to its PU T1, by its six-bit local address. The PU T1 can have an active session only with an SSCP, and each LU can have active sessions only with an SSCP and one other LU. The BF adjacent to each T1 node translates between the (LINK, STA, LSID) combination and the equivalent (DAF, OAF) network address pair. The (LINK, STA, LSID) combination implicitly determines the network address of the secondary LU. The relationship between the (LINK, STA, LSID) combination and the equivalent (DAF, OAF) network address pair is established in the boundary function. For LU-LU sessions, since each secondary LU can have an active session with only one primary LU at a time, the network address of the secondary LU suffices to identify the session to the adjacent T4 or T5 boundary function, which can then derive the network address of the primary LU. The LU-SSCP and LU-PU bit settings for sessions supported by FID3 flows are: 00 SSCP-PU session 01 SSCP-LU session 10 reserved 11 LU-LU session *Note:* For the SSCP-PU session, the local address is always 0.

FID4 Layout

Byte		
0	FID4—Format Identification TG_SWEEP—TG Sweep Indicator ER_VR_SUPP_IND—ER and VR Support Indicator VR_PAC_CNT_IND—VR Pacing Count Indicator NTWK_PRTY—Network Priority	TGSF—TG Segmenting Field Reserved Bits CHK—PIU Checksum Indicator HFT—HPR FID4 Type PIUBF—PIU Blocking Field
2	IERN—Initial Explicit Route Number --or-- NLPOI—NLP Offset Indicator NLP_C/P—NLP Count/Padding --- ERN—Explicit Route Number	VRN—Virtual Route Number Reserved Bits TPF—Transmission Priority Field
4	VR_CWI—Virtual Route Change Window Indicator TG_NONFIFO_IND—TG nonFIFO Indicator VR_SQTI—Virtual Route Sequencing and Type Indicator TG_SNF—Transmission–Group Sequence Number Field	
6	VRPRQ—Virtual Route Pacing Request VRPRS—Virtual Route Pacing Response VR_CWRI—Virtual Route Change Window Reply Indicator VR_RWI—Virtual Route Reset Window Indicator VR_SNF_SEND—Virtual Route Send Sequence Number Field	
8	DSAF—Destination Subarea Address Field --or-- HCHK—High-Order Byte of PIU Checksum DSAF—Destination Subarea Address Field	
12	OSAF—Origin Subarea Address Field --or-- LCHK—Low-Order Byte of PIU Checksum OSAF—Origin Subarea Address Field	
16	Reserved SNAI—SNA Indicator MPF—Mapping Field Reserved EFI—Expedited Flow Indicator	Reserved Byte
18	DEF—Destination Element Field	
20	OEF—Origin Element Field	
22	SNF—Sequence Number Field	
24	DCF—Data Count Field	

Figure 4-4. Transmission Header for FID Type 4

FID4 Field Descriptions

The FID4 format is used between adjacent subarea nodes, provided that both support ER and VR protocols. (FID0|1 is used if either node does not support ER and VR protocols.)

FID4 Field Descriptions

Byte	Bit	Content
0	0– 3	FID4—Format Identification: 0100
	4	TG_SWEEP—TG Sweep. 0 This PIU may overtake any PIU ahead of it in the transmission group. 1 This PIU does not overtake any PIU ahead of it in the transmission group. *Note:* The TG Sweep indicator, when set to 1 in the TH of a PIU, prevents that PIU from getting ahead of other PIUs flowing on the transmission group. Thus, various RUs, such as NC_ER_OP and NC_ER_INOP, can be processed in the order they originate. This is performed in the transmission group control components of PC.
	5	ER_VR_SUPP_IND—ER and VR Support Indicator. This bit is set to the appropriate value when the FID4 TH is originated (and/or when a FID4 TH replaces a FID0|1 TH) to indicate whether some subarea node on the route specified by this FID4 TH does not support ER and VR protocols. The transformation between FID4 and FID1 (or FID0 for non†SNA traffic) takes place in nodes adjacent to the subarea node that does not support ER and VR protocols. The VRN, IERN, and ERN fields must be set to 0 when this bit is set to 1. If this bit is *on* and the SNAI indicator is *on*, then FID4 is changed to FID1. Receipt of the first PIU, with this bit set to 1, on an ER results in activation of the ER (ERN = 0) and VR (VRN = 0, TPF = 0). 0 Each node on the explicit route traversed by this PIU supports ER and VR protocols. 1 The explicit route traversed by this PIU includes at least one node that does not support ER and VR protocols.
	6	VR_PAC_CNT_IND—Virtual Route Pacing Count Indicator. This bit is used to initiate implementation specific action to hasten the flow of isolated VRPRSs to the VR_PAC_CNT sender. It indicates that the VR_PAC_CNT sender cannot send any more PIUs, because its pacing count has reached 0. 0 Pacing count, on the VR specified in VRID, has not reached a value of 0. 1 Pacing count, on the VR specified in VRID, has reached a value of 0.
	7	NTWK_PRTY—Network Priority. This bit provides a transmission priority higher than those specified by TPF (see VRID, byte 3). It is used to transmit PIUs that must flow ahead of others—for example, to prevent network congestion. Currently, it is used only for isolated VRPRSs. 0 PIU flows at a lower priority, as specified in TPF. 1 PIU flows at network priority, which is the highest transmission priority.

FID4 Field Descriptions

Byte	Bit	Content
1	0–1	TGSF—Transmission Group Segmenting Field. The TGSF specifies whether the information field associated with the TH is a complete or partial PIU, and, if a partial PIU, whether it is the first, a middle, or the last segment. 00 not segmented 01 last segment 10 first segment 11 middle segment
	2–3	Reserved
	4	CHK—PIU Checksum Indicator: 0 A PIU checksum is not included in this TH. 1 A PIU checksum is included in bytes 8 and 12 of this TH.
	5	HFT—HPR FID4 Type: 0 not an HPR FID4 1 HPR FID4
	6–7	PIUBF—PIU Blocking Field. The PIUBF specifies whether this frame contains a single PIU or multiple PIUs. 00 Single PIU frame 01 Last PIU of a multiple PIU frame 10 First PIU of a multiple PIU frame 11 Middle PIU of a multiple PIU frame
2	0–3	IERN—Initial Explicit Route Number if HFT = 0 (see byte 1, bit 5), in which case this field has the same value as VRN (byte 3); **OR** NLPOI — NLP Offset Indicator (bit 0) — and NLP_C/P — NLP Count or Padding (bits 1–3) if HFT = 1, as described below.
	0	NLPOI — NLP Offset Indicator: 0 NLP starts within this FID4 TH and continues following RH byte 0. 1 NLP byte 0 starts after RH byte 0 following NLP_C/P pad (X′00′) bytes
	1–3	NLP_C/P — NLP Count or Padding: • When NLPOI=1, the number of pad bytes inserted between RH byte 0 and NLP byte 0, or • When NLPOI=0, the number of NLP bytes carried in this FID4 TH before the rest of the NLP continues following RH byte 0. A full count of 7 in this field means NLP bytes 0–6 would appear, respectively, in TH bytes 20, 21, 22, 23, 18, 19, and 7; any lesser count removes bytes from the left in the listed series (i.e., a count of 6 would start with byte 21, 5 with 22, 4 with 23, 3 with 18, 2 with 19, while a count of 1 would mean NLP byte 0 would appear in TH byte 7). ***Note:*** Irrespective of whether NLPOI is 0 or 1, a value of 000 in this NLP_C/P field means NLP byte 0 appears immediately after RH byte 0.
	4–7	ERN—Explicit Route Number. The ERN in a TH identifies an explicit route direction of flow (i.e., in the direction the TH is flowing). Two ERNs—one ERN for each direction of flow—together with the two subarea addresses (OSAF, DSAF), specify an explicit route.
3		VRID—Virtual Route Identifier. The VRID is made up of the VRN and TPF. This field, along with DSAF and OSAF, identifies a virtual route.
	0–3	VRN—Virtual Route Number
	4–5	Reserved
	6–7	TPF—Transmission Priority Field. TPF, if the NTWK_PRTY bit is set to 0, carries the PIU transmission priority to be used by transmission groups on the explicit route; otherwise, this field is ignored by the transmission group, and is simply used to identify the virtual route. 00 low priority 01 medium priority 10 high priority

FID4 Field Descriptions

Byte	Bit	Content
4– 5	0	VR_CWI—Virtual Route Change Window Indicator. This indicator is used to change the window size of the virtual route by 1, in the direction of flow of this PIU. Any transmission group on the virtual route can turn this bit *on* if it is congested; each subsequent transmission group on the virtual route leaves it *on*. The window size mechanism controls the PIU flow on the virtual route. Window size is the amount by which the current virtual route pacing count is incremented when a VRPRS bit set to VR_PAC_RSP is received. 0 Increment window size (only value used for HPR FID4 THs). 1 Decrement window size.
	1	TG_NONFIFO_IND—TG non-FIFO Indicator. This indicator identifies whether or not FIFO discipline is to be enforced in transmitting PIUs through the transmission groups to prevent the PIUs getting out of sequence during transmission over the TGs. (A transmission group may have more than one link, providing simultaneous transmission within the group.) When TG FIFO is not indicated, virtual route end-to-end resequencing is coordinated by UPMs. 0 TG FIFO is required. 1 TG FIFO is not required. *Note:* A 0 value must be used when performing TG segmenting.
	2– 3	VR_SQTI—VR Sequence and Type Indicator. These bits specify the PIU type. All network control (NC) RUs flowing between ER managers or VR managers are coded *nonsequenced, nonsupervisory*. An isolated VRPRS is coded *nonsequenced, supervisory*. All other PIUs are coded *singly sequenced*. 00 nonsequenced, nonsupervisory (only value used for HPR FID4 THs). 01 nonsequenced, supervisory 10 singly sequenced
	4– 1 5	TG_SNF—Transmission-Group Sequence Number Field. It is used by transmission group protocols to provide TG FIFO when TG_NONFIFO_IND is set to FIFO. This field is also used when performing TG segmenting; otherwise, it is reserved. For TG segmenting, the TG sequence number is incremented by 1 for each new segment of a given PIU.

FID4 Field Descriptions

Byte	Bit	Content
6–7	0	VRPRQ—Virtual Route Pacing Request. 0 No VR pacing response is requested. 1 VR pacing request is sent asking for a VR pacing response.
	1	VRPRS—Virtual Route Pacing Response. Virtual route pacing provides traffic flow control between the two ends of a VR. In contrast to session pacing, virtual route pacing operates on a group of sessions (on each VR) and extends only up to the VR endpoints (subarea nodes). The virtual route pacing uses a window size, say k. The sender (endpoint of a VR) can transmit k PIUs for every VRPRS set to VR_PAC_RSP received from the other VR endpoint. 0 No pacing response is sent. 1 VR pacing response is sent in response to a VRPRQ bit set to VR_PAC_RQ.
	2	VR_CWRI—Virtual Route Change Window Reply Indicator. This bit permits changing the window size by 1 for PIUs received by the sender of this bit. If VRPRS is set to VR_PAC_RSP, this bit is VR_CWRI; otherwise, it is reserved. 0 Increment window size by 1 without exceeding the maximum window size, as specified in NC_ACTVR. 1 Decrement window size by 1 without going under the minimum window size, as specified in NC_ACTVR.
	3	VR_RWI—Virtual Route Reset Window Indicator. This bit is set to indicate severe congestion in a node on the virtual route. When a VR endpoint receives this bit set to 1, it reduces the window size to the minimum window size. 0 Do not reset window size. 1 Reset window size to the minimum specified in NC_ACTVR.
	4–15	VR_SNF_SEND—Virtual Route Send Sequence Number Field. This field is reserved when VR_SQTI is set to *non-sequenced, non-supervisory*—except, see NLP_C/P field (byte 2, bits 1–3) for discussion of byte 7 usage for HPR FID4 THs. When VR_SQTI is set to *non-sequenced, supervisory*, i.e., the message is an isolated VRPRS, this field carries the VR_SNF_SEND value from the latest PIU received with the VRPRQ bit set to VR_PAC_RQ. When VR_SQTI is set to *singly sequenced*, this number is a sequence number initialized using a parameter carried in NC_ACTVR. The sender increments this count by 1 for every PIU sent. The VR receiver checks the sequenced arrival of PIUs by examining the VR_SNF_SEND values.
8–11		DSAF—4-byte Destination Subarea Address Field if CHK = 0; **OR** high-order byte of PIU Checksum (byte 8) and 3-byte Destination Subarea Address Field (bytes 9—11) if CHK = 1, as described below.
8		HCHK—High-Order Byte of the PIU Checksum. See **Note 1**.
9–11		DSAF—3-byte Destination Subarea Address Field. See **Note 2**.
12–15		OSAF—4-byte Origin Subarea Address Field if CHK = 0; **OR** low-order byte of PIU Checksum (byte 12) and 3-byte Origin Subarea Address Field (bytes 13—15) if CHK = 1, as described below.
12		LCHK—Low-Order Byte of the PIU Checksum. See **Note 1**.
13–15		OSAF—3-byte Origin Subarea Address Field. See **Note 2**.

FID4 Field Descriptions

Byte	Bit	Content
16	0–2	Reserved
	3	SNAI—SNA Indicator. This bit is used to identify whether the PIU originated or is destined for an SNA or non-SNA device. If this bit is *off*, the TH is converted to FID0 in the node supporting the non-SNA device. If this bit is *on* and the ER_VR_SUPP_IND is set to PRE_ER_VR, the TH is converted to FID1 in the node adjacent to the node not supporting ERs and VRs. 0 ¬ SNA 1 SNA (only value used for HPR FID4 THs)
	4–5	MPF—Mapping Field. As explained earlier (always 11 for HPR FID4 THs).
	6	Reserved
	7	EFI—Expedited Flow Indicator. As explained earlier (always 1 for HPR FID4 THs). *Note:* For NC RUs, the EFI is set to 1 and reserved.
17		Reserved
18–19		DEF—Destination Element Field (but see the discussion in byte 2, bits 1–3 for usage in HPR FID4 THs). A two-byte destination element address field. The complete network address results from the combination of DSAF and DEF. An element address of 0 denotes the PU T4|5 controlling the associated subarea.
20–21		OEF—Origin Element Field (but see the discussion in byte 2, bits 1–3 for usage in HPR FID4 THs). A two-byte origin element address field. The complete network address results from the combination of OSAF and OEF.
22–23		SNF—Sequence Number Field, as described earlier (but see the discussion in byte 2, bits 1–3 for usage in HPR FID4 THs).
24–25		DCF—Data Count Field, as described earlier.
		Note 1: Each halfword of the PIU starting with the FID4 TH is treated as an integer. The checksum is the sum of the first 17 halfwords of the PIU using modulo 65535 (i.e., X′FFFF′) arithmetic. When the PIU is less than 34 bytes, the PIU is padded to the right with zeroes for the checksum computation. When the checksum is the integer 0, it is represented as X′FFFF′. As an example, the checksum for the PIU, X′4008 7770 2352 0041 0000 0002 0000 0051 1C00 0155 0095 000F 0003 8380 00′, is X′7CDB′.
		Note 2: When both partners support PIU checksum as indicated in the PIU Checksum Support indicator of XID2, a two-byte checksum is generated for each PIU and carried in byte 8 and byte 12 of the FID4 TH. As a result, both subarea address fields are limited to 3 bytes (i.e., the maximum subarea address is 16,777,215) when checksums are used for PIUs transported across the link.

FID5 Layout

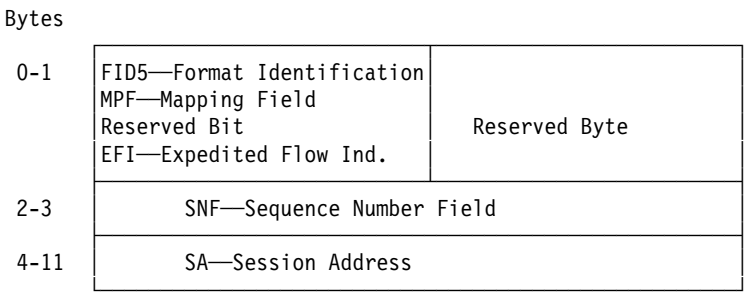


Figure 4-5. Transmission Header for FID Type 5

FID5 Field Descriptions

The FID5 format is used within network layer packets (NLPs) between HPR RTP endpoint nodes for LU-LU and CP-CP session traffic.

FID5 Field Descriptions

Byte	Bit	Content
0	0– 3	FID5—Format Identification: 0101
	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.
	6	Reserved.
	7	EFI—Expedited Flow Indicator. The EFI designates whether the PIU belongs to the normal or expedited flow. Normal-flow PIUs are kept in order on a session basis by PC; so are expedited-flow PIUs. Expedited-flow PIUs can pass normal-flow PIUs flowing in the same direction at queuing points in TC within half-sessions and boundary function and APPN session connectors. It has the following meaning: 0 normal flow 1 expedited flow
1		Reserved

FID5 Field Descriptions

Byte	Bit	Content
2–3		SNF—Sequence Number Field. The Sequence Number Field contains a numerical identifier for the associated BIU; path control, when segmenting, puts the same SNF value in each segment derived from the same BIU. The numerical identifier used depends on a number of factors. If the TS profile indicates sequence numbers are not used, the SNF value is a 16-bit identifier that distinguishes a request being sent or responded to from any other outstanding request on the same flow. If the TS profile indicates sequence numbers are used, the flow is a factor. Expedited-flow requests (other than SIG for LU 6.2) carry 16-bit identifiers; expedited-flow responses echo the SNF values of their corresponding requests. Normal-flow requests, other than between LU 6.2s, carry 16-bit numerical values ranging in value from 1–65,535 (incremented by 1 for each request) and wrapping through 0 thereafter; the corresponding responses echo their SNF values. The table below defines the SIG and normal-flow SNF usage between LU 6.2s.

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

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

B: Low-order 15 bits of the SNF in the request that carried the last successful BB; the high-order bit identifies the half-session that started the bracket (0 = secondary, 1 = primary); in the case of the first bracket of a session, where the BB is implied, not sent, the low-order 15 bits are 0 and the high-order bit is 1.

C: Low-order 15 bits of the SNF in the BB request being responded to; the high-order bit identifies the sender of the BB request (0 = secondary, 1 = primary).

D: The half-session does not respond to BIS.

E: Same value as the corresponding request.

Note: For additional details of LU 6.2 processing, see *SNA LU 6.2 Reference: Peer Protocols*.

4–11		SA—Session Address (replaces OAF′, DAF′, and ODAI fields used in FID2). Two addresses are associated with each session—one in each direction. The receiver node assigns the address used for session traffic being received.
	0	Session address assignor indicator—indicates which HPR RTP partner assigned this session address (set to 1 in a BIND PIU or when the sender does not yet have a partner-assigned SA it can use, and to 0 otherwise): 0 The receiver of this FID5 PIU assigned this address. 1 The sender of this FID5 PIU assigned this address.
	1	Reserved
	2–63	Session address: a binary value

FIDF Layout

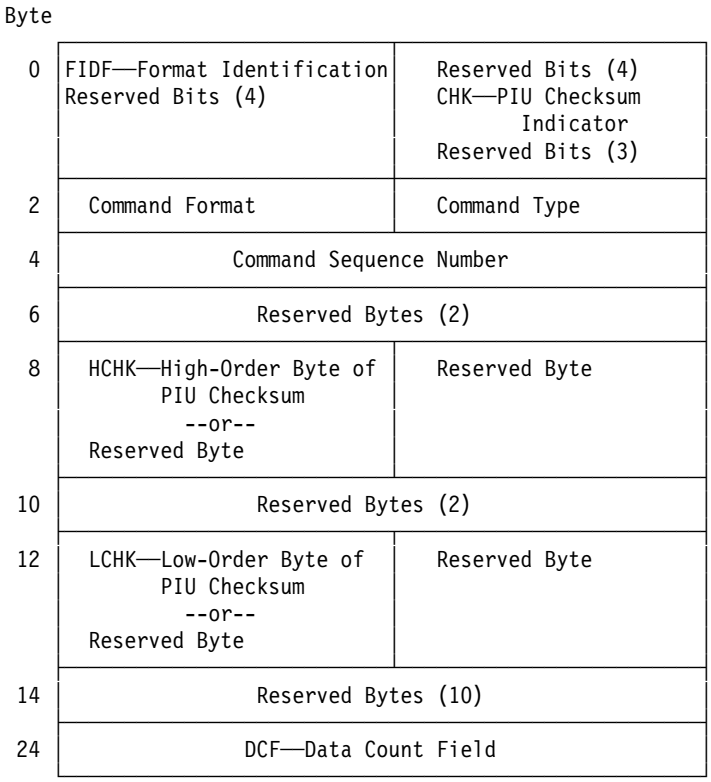


Figure 4-6. Transmission Header for FID Type F

FIDF Field Descriptions

The FIDF format is used between adjacent subarea nodes if both support ER and VR protocols.

FIDF Field Descriptions

Byte	Bit	Content
0	0– 3	FIDF—Format Identification: 1111
	4– 7	Reserved
1	0– 3	Reserved
	4	CHK—PIU Checksum Indicator: 0 A PIU checksum is not included in this TH. 1 A PIU checksum is included in bytes 8 and 12 of this TH.
	5– 7	Reserved
2		Command Format—X′01′ for currently defined format (only value defined).
3		Command Type—X′01′ for currently defined type, to indicate Transmission-Group Sequence-Number-Field Wrap Acknowledgment command (only value defined).

FIDF Field Descriptions

Byte	Bit	Content
4–5		Command Sequence Number—Identifier sequence number for Transmission-Group Sequence-Number-Field Wrap Acknowledgment command. This sequence number is distinct from the Transmission-Group Sequence Number field in the FID4 TH.
6–7		Reserved
8		HCHK—High-Order Byte of the PIU Checksum: (This field is reserved if CHK = 0.) See **Note**.
9–11		Reserved
12		LCHK—Low-Order Byte of the PIU Checksum: (This field is reserved if CHK = 0.) See **Note**.
13–23		Reserved
24–25		DCF—Same as described earlier
		Note: Each halfword of the PIU starting with the FIDF TH is treated as an integer. The checksum is the sum of the first 17 halfwords of the PIU using modulo 65535 (i.e., X′FFFF′) arithmetic. When the PIU is less than 34 bytes, the PIU is padded to the right with zeroes for the checksum computation. When the checksum is the integer 0, it is represented as X′FFFF′. As an example, the checksum for the PIU, X′F008 0101 0F20 0000 0000 0000 0000 0000 0000 0000 0000 0000′ is X′002A′.

End of Chapter 4

Chapter 5. Request/Response Headers (RHs)

Introduction

This chapter identifies the formats and meanings of the request and response headers (RH); Chapter 6, “Request/Response Units (RUs)” describes 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.

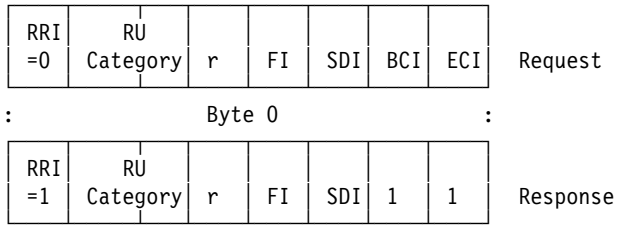
If bit 0 = 1: the RH is a response header and any associated RU is a response unit.

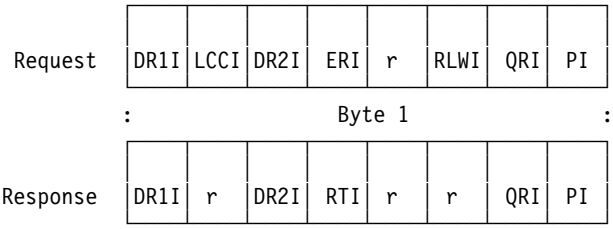
Figure 5-1 on page 5-4 provides a summary of the bytes and field names in the RH.

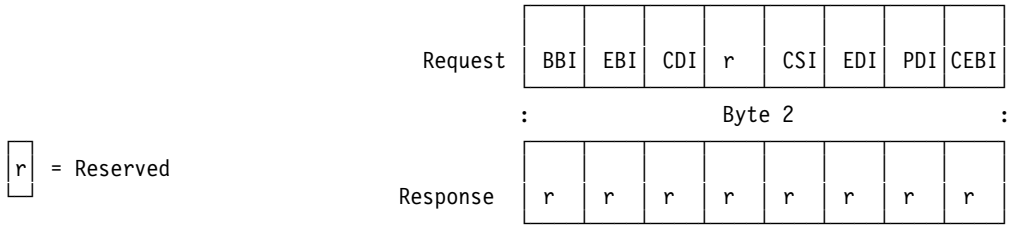
Length-checked compression (LCC) and the algorithms and additional formats supporting LCC are described in the section “Length-Checked Compression” on page 5-11. The lengths of request units are changed by LCC.

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

Request/Response Header







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 5-1 (Part 1 of 2). RH Formats

Field	Description	Explanation/Usage
ECI	End Chain indicator	0 = not last in chain (—EC); 1 = last in chain (EC)
DR1I	Definite Response 1 indicator	0 = —DR1; 1 = DR1
LCCI	Length-Checked Compression indicator	0 = RU is not compressed (—LCC); 1 = RU is compressed (LCC)
DR2I	Definite Response 2 indicator	0 = —DR2; 1 = DR2
ERI	Exception Response indicator	Used in conjunction with DR1I and DR2I to indicate, in a request, the form of response requested. Values and meanings of DRI1I, 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 5-1 (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 5-1 on page 5-4 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).

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

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

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

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

For LU-LU sessions that do not support FM headers, the meaning of this indicator on requests, positive responses, and negative responses is implementation dependent. (A BIND session parameter indicates whether FM headers are supported by the session. For further information, see Chapter 6, “Request/Response Units (RUs)” for details on BIND.)

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

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

Chaining Control: Indicates that a sequence of contiguous transmitted requests is being grouped in a chain. Two indicators, Begin Chain indicator (BCI) and End Chain indicator (ECI), together denote the relative position of the associated RU within a chain. The 1 values of these indicators (BCI = 1 and ECI = 1) are referred to as BC and EC, respectively.

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

Responses are always marked “only RU in chain.”

Length-Checked Compression Indicator (LCCI): Indicates that the request unit contains compressed data. Only normal-flow FMD request units can be compressed. When LCCI is set to LCC, the RU consists of a compression header (see “Length-Checked Compression” on page 5-11) followed by compressed data.

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

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

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

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

A request that asks for an exception response or a definite response has one or both of the DR1I and DR2I bits set to 1 (three combinations); a response to a request returns the same (DR1I, DR2I) bit combination (see Figure 5-2 on page 5-8).

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 5-2 on page 5-8 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 5-3 on page 5-9.

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 5-2. 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 5-3. Request/Response Combinations For TS Profile 4 Sync Points

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

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

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

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

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

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

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

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

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

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

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

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

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

Length-Checked Compression

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

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

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

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

Currently, two LCC algorithms exist: run-length encoding (RLE) and an adaptive dictionary-based algorithm called *LZ compression*, similar to the Lempel-Ziv algorithm. Their usage is specified in the compression header, which has the format shown below.

Compression Header

Byte	Bit	Content
0	0– 3	Compression algorithm: 0001 RLE, if uncompressed data type (indicated in bits 4–7) is 0001 0010 LZ, if uncompressed data type is 0001 0011 LZ compression after RLE, if uncompressed data type is 0001
	4– 7	Uncompressed data type and compression header size: 0001 8-bit text data; compression header size is 3 bytes
1– 2		Length, in binary, of uncompressed RU

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

Compression on XRF sessions is supported. If the BIND negotiation on an XRF session results in use of compression, the PLU-to-SLU traffic will be compressed with either small-table LZ or RLE, and the SLU-to-PLU traffic will be compressed with RLE.

Run-Length Encoding

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

SCB Format

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

SCBs cannot span RUs. In short:

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

- If the duplicate-character SCB is the last SCB in the RU, then that SCB is always the next-to-last byte in the RU, the last byte being the DC.

Adaptive Dictionary-Based Compression

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

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

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

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

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

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

LZ Control Sequence

The LZ control sequence is a two-byte format that may appear only directly after a compression header that also indicates the RU is LZ-compressed. The control sequence consists of the LZ control code and the LZ command. The control code is the 9-, 10-, or 12-bit encoding (depending on table size) of 256. (The

code 256 is reserved for this purpose; the compression tables do not use this code.) The format of the 2-byte control sequence is shown below.

LZ Control Sequence Format

Bit	Content
0 – i(=8|9|11)	A right-justified binary 256 control code indicating the following field carries an LZ command.
i + 1 – 15	A right-justified hex value representing an LZ command: X′1′ Reset the table to its initial state. X′2′ Freeze the table in its current state; do not update it for new strings. X′3′ Unfreeeze the table; update it for new strings.

BIND Negotiation for Compression

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

Two ways exist to negotiate for compression:

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

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

IPR, IPM, and EXR

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

ISOLATED PACING RESPONSE (IPR)

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

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

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

RH: An IPR is coded all 0′s except for the Request/Response indicator, the Pacing indicator, and the Chain indicators, which are set to 1′s; thus, the IPR RH

is coded X′830100′ by the sender; the receiver identifies an IPR by detecting that (RRI, DR1I, DR2I, PI) = (1, 0, 0, 1) and ignoring the remaining bits.

ISOLATED PACING MESSAGE (IPM)

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

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

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

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

The following fields are set for an IPM.

TH: Expedited flow is indicated except for a reset acknowledgment IPM, which is always sent normal-flow (because it delimits the current window). The sequence number is undefined (may be set to any value, and is not checked by the receiver). For an adaptive BIND pacing IPM, OAF′ and DAF′ are set according to the sender's normal setting of ODAI for BIND: a node that sets ODAI to 0 for BIND sets OAF′ to X′01′ and DAF′ to X′00′ for the BIND pacing IPM, while a node that sets ODAI to 1 for BIND sets OAF′ to X′00′ and DAF′ to X′01′ for the BIND pacing IPM. ODAI for the adaptive BIND pacing IPM is set to 0 unless both the local and adjacent nodes indicate support for APPN option set 1071 (Generalized ODAI Usage) in XID3. If both nodes indicate support for option set 1071, ODAI for the adaptive BIND pacing IPM is set to the same value as for BIND.

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, if present in the TH, 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: EXRs replacing too-long PIUs change only the Mapping field (to 1′s, i.e., BBIU and EBIU) and, if present in the TH, the Data Count field (to the appropriate value in this instance).

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′. If the PIU is a positive response, the RH is changed to indicate SD and negative response. If the PIU is a negative response, the RH is unchanged.

RU: Bytes 0–3 always contain the sense data, X′800Axxxx′ (replacing any sense data received in a too-long negative-response PIU). Bytes 4–6 contain the first three bytes of the original RU (or the three bytes received following the sense data in a too-long negative-response PIU). When xxxx=0002, the EXR is extended by the additional following fields in the order listed:

- A 2-byte binary value specifying the number of bytes in the PIU before its truncation
- A 2-byte binary value specifying the maximum size of a PIU allowed on the link in question
- Two Network Name (X′0E′) control vectors, the first (Type=X′F1′) identifying the PU name of the node detecting the error, and the second (Type=X′F7′) identifying the link station name for the link that could not accept the too-large PIU. These control vectors are parsed LT.

End of Chapter 5

Chapter 6. Request/Response Units (RUs)

Introduction to Request Units

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

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

Notes:

1. "RU Category" is abbreviated as follows:

DFC	data flow control
SC	session control
NC	network control
FMD NS(c)	function management data, network services, configuration services
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. All values for field-formatted requests that are not defined in this section are reserved. (The formats of character-coded FMD NS requests are implementation dependent.)

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

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

   - Reserved bits, or fields, are ones that currently are set to 0's (unless explicitly stated otherwise). Correct usage of reserved fields is enforced by the sender; no receive checks are made on these fields.
   - Reserved values are those that currently are invalid. Receive checks are applied to detect usage of reserved values.

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

6. User data, control vectors, and session keys referred to in the format descriptions are described in Chapter 8, "User Data Structured Subfields" and Chapter 9, "Common Fields."

7. The character sets referred to in the descriptions of names and other symbol strings in this chapter are defined in Appendix A, “SNA Character Sets and Symbol-String Types.”
8. A type 2.1 (T2.1) node contains a control point (CP) rather than a physical unit (PU). However, it can support SSCP-PU T2.0 flows, in which case the designations “SSCP ⟷ PU T2” or “SSCP ⟷ PU” in the RU descriptions should be assumed to apply to the T2.1 node as well.

Request Unit Summary Information

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

Summary of Request RUs by Category

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

NC Requests

†LSA	NC-ER-INOP	NC-IPL-FINAL
NC-ACTVR	NC-ER-OP	NC-IPL-INIT
NC-DACTVR	NC-ER-TEST	NC-IPL-TEXT
NC-ER-ACT	NC-ER-TEST-REPLY	ROUTE SETUP
NC-ER-ACT-REPLY	NC-IPL-ABORT	

SC Requests

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

DFC Requests

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

FMD NS(c) Requests

ABCONN
ABCONNOUT
ACTCONNIN
*ACTLINK
*ADDLINK
*ADDLINKSTA
†ANA
CONNOUT
CONTACT
CONTACTED
DACTCONNIN
DACTLINK
DELETENR
DISCONTACT
DUMPFINAL
*DUMPINIT
*DUMPTEXT
†ER-INOP
ESLOW
EXSLOW
FNA
INITPROC
INOP
IPLFINAL
IPLINIT
IPLTEXT
LCP
†LDREQD
NOTIFY (SSCP-PU)
†NS-IPL-ABORT
†NS-IPL-FINAL
†NS-IPL-INIT
†NS-IPL-TEXT
†NS-LSA
PROCSTAT
REQACTCDRM
REQACTLU
REQACTPU
REQCONT
REQDACTPU
REQDISCONT
REQFNA
*RNAA
ROUTE-INOP
RPO
*SETCV
†VR-INOP

FMD NS(ma) Requests

ACTTRACE
DACTTRACE
†DELIVER
DISPSTOR
†ECHOTEST
ER-TESTED
EXECTEST
†FORWARD
NMVT
RECFMS
†RECMS
RECSTOR
RECTD
RECTR
RECTRD
†REQECHO
REQMS
*ROUTE-TEST
SETCV
TESTMODE

FMD NS(s) Requests

BFCINIT
BFCLEANUP
BFINIT
BFSESSEND
BFSESSINFO
BFSESSST
BFTERM
BINDF
CDCINIT
*CDINIT
CDSESSEND
†CDSESSSF
CDSESSST
†CDSESSTF
CDTAKED
CDTAKEDC
*CDTERM
*CINIT
CLEANUP
CTERM
*DSRLST
INIT-OTHER
*INIT-OTHER-CD
INIT-SELF
NOTIFY (SSCP-SSCP|LU)
NSPE
SESSEND
SESSST
TERM-OTHER
TERM-SELF
UNBINDF

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). DSRLST, NOTIFY, and SETCV are exceptions: these three requests have request codes—X′27′, X′20′, and X′11′, respectively—that are unique across all the FMD NS categories.

FMD NS Headers (third byte is the request code)

X′010201′	CONTACT
X′010202′	DISCONTACT
X′010203′	IPLINIT
X′010204′	IPLTEXT
X′010205′	IPLFINAL
X′010206′	DUMPINIT
X′010207′	DUMPTEXT
X′010208′	DUMPFINAL
X′010209′	RPO
X′01020A′	ACTLINK
X′01020B′	DACTLINK
X′01020E′	CONNOUT
X′01020F′	ABCONN
X′010211′	SETCV (FMD NS(c))
X′010214′	ESLOW
X′010215′	EXSLOW
X′010216′	ACTCONNIN
X′010217′	DACTCONNIN
X′010218′	ABCONNOUT
X′010219′	ANA
X′01021A′	FNA
X′01021B′	REQDISCONT
X′010280′	CONTACTED
X′010281′	INOP
X′010284′	REQCONT
X′010285′	NS-LSA
X′010301′	EXECTEST
X′010302′	ACTTRACE
X′010303′	DACTTRACE
X′010311′	SETCV (FMD NS(ma))
X′010331′	DISPSTOR
X′010334′	RECSTOR
X′010381′	RECMS
X′010382′	RECTD
X′010383′	RECTRD
X′010604′	NSPE
X′010681′	INIT-SELF (Format 0)
X′010683′	TERM-SELF (Format 0)
X′410210′	RNAA
X′41021C′	DELETENR
X′41021D′	ER-INOP
X′41021E′	ADDLINK
X′410220′	NOTIFY (SSCP ⟷ PU)
X′410221′	ADDLINKSTA
X′410223′	VR-INOP
X′410235′	INITPROC
X′410236′	PROCSTAT
X′410237′	LDREQD
X′41023E′	REQACTPU
X′41023F′	REQDACTPU

X′410240′	REQACTLU
X′410243′	NS-IPL-INIT
X′410244′	NS-IPL-TEXT
X′410245′	NS-IPL-FINAL
X′410246′	NS-IPL-ABORT
X′410286′	REQFNA
X′410287′	LCP
X′410289′	ROUTE-INOP
X′41028A′	REQACTCDRM
X′410304′	REQMS
X′410305′	TESTMODE
X′410307′	ROUTE-TEST
X′410384′	RECFMS
X′410385′	RECTR
X′410386′	ER-TESTED
X′41038D′	NMVT
X′810387′	REQECHO
X′810389′	ECHOTEST
X′810601′	CINIT
X′810602′	CTERM
X′810620′	NOTIFY (SSCP ⟷ LU)
X′810629′	CLEANUP
X′810680′	INIT-OTHER
X′810681′	INIT-SELF (Format 1)
X′810682′	TERM-OTHER
X′810683′	TERM-SELF (Format 1)
X′810685′	BINDF
X′810686′	SESSST
X′810687′	UNBINDF
X′810688′	SESSEND
X′810810′	FORWARD (retired)
X′810812′	DELIVER (retired)
X′812601′	BFCINIT
X′812629′	BFCLEANUP
X′812681′	BFINIT
X′812683′	BFTERM
X′812686′	BFSESSST
X′812688′	BFSESSEND
X′81268C′	BFSESSINFO
X′818620′	NOTIFY (SSCP→SSCP)
X′818627′	DSRLST
X′818640′	INIT-OTHER-CD
X′818641′	CDINIT
X′818643′	CDTERM
X′818645′	CDSESSSF
X′818646′	CDSESSST
X′818647′	CDSESSTF
X′818648′	CDSESSEND
X′818649′	CDTAKED
X′81864A′	CDTAKEDC
X′81864B′	CDCINIT

DFC, NC, and SC Request Codes

X′02′	NC-IPL-FINAL (NC)
X′03′	NC-IPL-INIT (NC)
X′03′	EXPD (DFC)
X′04′	LUSTAT (DFC)
X′04′	NC-IPL-TEXT (NC)
X′05′	RTR (DFC)
X′05′	LSA (NC)
X′06′	NC-ER-INOP (NC)
X′09′	NC-ER-TEST (NC)
X′0A′	NC-ER-TEST-REPLY (NC)
X′0B′	NC-ER-ACT (NC)
X′0C′	NC-ER-ACT-REPLY (NC)
X′0D′	ACTLU (SC)
X′0D′	NC-ACTVR (NC)
X′0E′	DACTLU (SC)
X′0E′	NC-DACTVR (NC)
X′0F′	NC-ER-OP (NC)
X′10′	ROUTE SETUP (NC)
X′11′	ACTPU (SC)
X′12′	DACTPU (SC)
X′14′	ACTCDRM (SC)
X′15′	DACTCDRM (SC)
X′31′	BIND (SC)
X′32′	UNBIND (SC)
X′33′	SWITCH (SC)
X′46′	NC-IPL-ABORT (NC)
X′70′	BIS (DFC)
X′71′	SBI (DFC)
X′80′	QEC (DFC)
X′81′	QC (DFC)
X′82′	RELQ (DFC)
X′83′	CANCEL (DFC)
X′84′	CHASE (DFC)
X′A0′	SDT (SC)
X′A1′	CLEAR (SC)
X′A2′	STSN (SC)
X′A3′	RQR (SC)
X′C0′	SHUTD (DFC)
X′C0′	CRV (SC)
X′C1′	SHUTC (DFC)
X′C2′	RSHUTD (DFC)
X′C8′	BID (DFC)
X′C9′	SIG (DFC)

Descriptions of Request Units

ABCONN (ABANDON CONNECTION)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

ABCONN requests the PU to deactivate the link connection for the specified link.

ABCONN (ABANDON CONNECTION)

Byte	Bit	Content
0– 2		X′01020F′ NS header
3– 4		Element address of link, if ENA is supported; otherwise, its network address

ABCONNOUT (ABANDON CONNECT OUT)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

ABCONNOUT requests the PU to terminate a connect-out procedure on the designated link.

ABCONNOUT (ABANDON CONNECT OUT)

Byte	Bit	Content
0– 2		X′010218′ NS header
3– 4		Element address of link, if ENA is supported; otherwise, its network address

ACTCDRM (ACTIVATE CROSS-DOMAIN RESOURCE MANAGER)

SSCP→SSCP, Exp; SC

ACTCDRM is sent from one SSCP to another SSCP to activate a session between them and to exchange information about the SSCPs.

ACTCDRM (ACTIVATE CROSS-DOMAIN RESOURCE MANAGER)

Byte	Bit	Content
0		X′14′ request code
1	0– 3	Format: X′0′ (only value defined)
	4– 7	Type activation requested: X′1′ cold X′2′ ERP

ACTCDRM (ACTIVATE CROSS-DOMAIN RESOURCE MANAGER)

Byte	Bit	Content
2		FM profile (see Chapter 7, “Profiles”)
3		TS profile (see Chapter 7, “Profiles”)
4–11		Contents ID: 8-character EBCDIC symbolic name that represents implementation- and installation-dependent information about the SSCP issuing the ACTCDRM; eight space (X′40′) characters is the value used if no information is to be conveyed (This field could be used to provide a check for a functional and configurational match between the SSCPs.)
12–17		SSCP ID: a 6-byte field that includes the ID of the SSCP issuing the ACTCDRM; the first four bits specify the format for the remaining bits:
	0–3	Format 0000 (only value defined)
	4–7	Physical unit type of the node containing the SSCP
	8–47	Implementation and installation dependent binary identification
18		TS Usage
	0–1	Reserved
	2–7	Primary half-session receive window size (0 means no pacing of requests flowing to the primary)
19 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′06′ CDRM control vector X′09′ Activation Request/Response Sequence Identifier control vector X′13′ Gateway Support Capabilities control vector X′18′ SSCP Name control vector X′60′ Fully Qualified PCID control vector X′FE′ Control Vector Keys Not Recognized control vector

ACTCONNIN (ACTIVATE CONNECT IN)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

ACTCONNIN requests the PU to enable the specified link to accept incoming calls. It can also be used to solicit information about an existing connection on the link.

ACTCONNIN (ACTIVATE CONNECT IN)

Byte	Bit	Content
0–2		X′010216′ NS header
3–4		Element address of link, if ENA is supported; otherwise, its network address

ACTCONNIN (ACTIVATE CONNECT IN)

Byte	Bit	Content
5		Indicators for link connections:
	0	Incoming call indicator: 0 enable for incoming calls 1 disable for incoming calls
	1	Information request indicator: 0 information on link connection not requested 1 information on link connection requested
	2–7	Reserved
6 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′46′ (with subfield X′80′) TG Descriptor control vector with TG Identifier subfield *Note:* Control vector X′46′ is present when the connection is APPN and the APPN networking functions support bit (byte 5, bit 4) in the SSCP-PU Session Capabilities (X′0B′) control vector is set in the ACTPU and ACTPU RSP.

ACTLINK (ACTIVATE LINK)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

ACTLINK initiates a procedure at the PU to activate the protocol boundary between a link station in the node (as specified by the link network address parameter in the request) and the link connection attached to it.

ACTLINK (ACTIVATE LINK)

Byte	Bit	Content
0–2		X′01020A′ NS header
3–4		Element address of link, if ENA is supported; otherwise, its network address
5		Indicators:
	0	Switched subarea support indicator: 0 switched subarea not supported 1 switched subarea supported
	1	Subordinate logical link indicator: 0 not a subordinate logical link 1 logical link subordinate to a given port
	2–7	Reserved

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	Enhanced address management indicator: 0 Sender does not support enhanced address management. 1 Sender supports enhanced address management.
	1	Static/dynamic address indicator (reserved if byte 1, bit 0 = 0): 0 Sender considers the LU address to be static. 1 Sender considers the LU address to be dynamic.
	2–5	Reserved
	6–7	Type activation requested: 01 cold (retired) 10 ERP
2	0–3	FM profile: X′0′ FM profile 0 X′6′ FM profile 6
	4–7	TS profile: X′1′ TS profile 1 (only value defined)
3 – n		Control vector as described in the section “Control Vectors” in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′0E′ Network Name control vector (contains the LU name, type X′F3′; included if byte 1, bit 0 = 1)

ACTPU (ACTIVATE PHYSICAL UNIT)

SSCP|PUCP→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—may include one or more control vectors in bytes 9 – n X′1′ Format 1—same as Format 0, except that it always includes one or more control vectors in bytes 9 – n (sent only to type 2.1 nodes that use XID3 with byte 10, bit 3 set to 1) X′3′ Format 3—same as Format 0, except that it always includes one or more control vectors in bytes 9 – n (sent only to PU T4|5s that support ERs and VRs)
	4–7	Type activation requested: X′1′ cold (retired) X′2′ ERP

ACTPU (ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
2	0–3	FM profile: X′0′ FM profile 0 X′5′ FM profile 5
	4–7	TS profile: X′1′ TS profile 1 X′5′ TS profile 5
3–8		A 6-byte field that specifies the ID of the SSCP issuing ACTPU; the first four bits specify the format for the remaining bits:
	0–3	Format: 0000 (only value defined)
	4–7	Type of the node containing the SSCP
	8–47	Implementation and installation-dependent binary identification
9 – n		Control vectors as described following this RU or in the section “Control Vectors” in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′09′ Activation Request/Response Sequence Identifier control vector (present when sent to a PU T4|5) X′0B′ SSCP-PU Session Capabilities control vector (present when sent to a PU T4|5) X′0E′ Network Name control vector (contains the name of the PU, type X′F1′; present for Formats 0 and 1 when sent: (1) by an SSCP to an NCP only when that NCP is then to forward it to a PU T2.0|2.1; or (2) by a DLUS to a DLUR in all instances; or (3) by a DLUR to a PU T2.0|2.1 when asked to forward it by a DLUS) X′0E′ Network Name control vector (contains the name of the sending control point, type X′F4′; present when sent to a PU T4|5) X′18′ SSCP Name control vector (present when sent to a PU T4|5) X′80′ PU Capabilities control vector (always present on Format 1; present on Format 0 when sent: (1) between a DLUS and a DLUR that both support network name forwarding; or (2) by an SSCP requesting network name forwarding; or (3) by an NCP or DLUR forwarding a CV X′0E′)

PU Capabilities (X′80′) ACTPU Control Vector

PU Capabilities (X′80′) ACTPU Control Vector

Byte	Bit	Content
0–1		Vector header; Key= X′80′

PU Capabilities (X′80′) ACTPU Control Vector

Byte	Bit	Content
2		Vector Data
	0	Unsolicited NMVT support: 0 Sending node does not support unsolicited NMVTs for PSID. 1 Sending node supports unsolicited NMVTs for PSID.
	1	Network name forwarding request: 0 Sending node does not request the DLUR or NCP boundary function to include Network Name control vectors on ACTPUs and ACTLUs for this PU and its associated LUs. 1 Sending node requests the DLUR or NCP boundary function to include Network Name control vectors on ACTPUs and ACTLUs for this PU and its associated LUs. *Note:* This bit is reserved except when the sender is an SSCP whose CV X′0B′ exchange with the receiving NCP revealed that both support network name forwarding, or when the sender is a DLUS whose CV X′51′ exchange with the receiving DLUR revealed that both support network name forwarding.
	2	TCP/IP information forwarding authorization: 0 Receiving node may not include a TCP/IP Information (X′64′) control vector on RSP(ACTLU) or NOTIFY. 1 Receiving node may include a TCP/IP Information (X′64′) control vector on RSP(ACTLU) or NOTIFY.
	3–7	Reserved

ACTTRACE (ACTIVATE TRACE)

SSCP→PU T4|5, Norm; FMD NS(ma)

ACTTRACE requests the PU to activate a specified type of trace for a specified resource (or hierarchy of resources for a generalized PIU trace).

ACTTRACE (ACTIVATE TRACE)

Byte	Bit	Content
0–2		X′010302′ NS header
3–4		Element address of the resource associated with the trace, if ENA is supported; otherwise, its network address *Note:* For generalized PIU trace (byte 5, bit 1 set to 1), bytes 3–4 contain the address of the PU receiving ACTTRACE; the address of the specific resource identifying the resource hierarchy to be traced is contained in bytes 7–8.
5		Selected trace (a bit is set to 1 if the indicated trace option is selected):
	0	Transmission group trace
	1	Generalized PIU trace (GPT) *Note:* When this bit is set to 1, all other bits in this byte are set to 0.
	2–3	Reserved
	4	Scanner internal trace
	5	Reserved
	6	Trace all Frames
	7	Link trace
6 – n		Specific trace data

For byte 5, bit 1 = 0

ACTTRACE (ACTIVATE TRACE)

Byte	Bit	Content
6		Maximum time interval between RECTRD RUs (in units of .1 seconds)
7		Maximum number of bytes to be traced in each PIU, where X′FF′ means trace all data (no maximum)
8(=n)		Tracepoint or omitted: Implementation defined
For byte 5, bit 1 = 1		
6		Generalized PIU trace flags
	0–6	Reserved
	7	0 activate trace for all resources within the resource hierarchy of the PU receiving ACTTRACE (bytes 7–8 are reserved) 1 activate trace for resources within the resource hierarchy of the resource specified in bytes 7–8
7–8 (=n)		Element address of a specific resource identifying a resource hierarchy to be traced, if ENA is supported; otherwise, its network address (if byte 6, bit 7 = 1; otherwise, reserved)

ADDLINK (ADD LINK)

SSCP→PU T5, Norm; FMD NS(c)

ADDLINK is sent from the SSCP to the PU to obtain a link network address that will be mapped to the locally-used link identifier specified in the request.

ADDLINK (ADD LINK)

Byte	Bit	Content
0–2		X′41021E′ NS header
3–4		Element address of target PU, if ENA is supported; otherwise, its network address
5–6		Reserved
7		Length of local link identifier
8 – n		Local link identifier

ADDLINKSTA (ADD LINK STATION)

SSCP→PU T5, Norm; FMD NS(c)

ADDLINKSTA is sent from the SSCP to the PU to obtain an adjacent link station network address to be associated with the locally-used link station identifier specified in the request.

ADDLINKSTA (ADD LINK STATION)

Byte	Bit	Content
0– 2		X′410221′ NS header
3– 4		Element address of target PU or link, if ENA is supported; otherwise, its network address
5		FID types supported:
	0	1 FID0 support
	1	1 FID1 support
	2	1 FID2 support
	3	1 FID3 support
	4	1 FID4 support
	5– 7	Reserved
6		Reserved
7		Length of link station identifier *Note:* When assigning an address for a link station on a point to point link, this field can be 0, the link station identifier is omitted, and the target network address in bytes 3 and 4 indicates the link to which the link station belongs.
8 – n		Link station identifier

ANA (ASSIGN NETWORK ADDRESSES)

SSCP→PU T4|5, Norm; FMD NS(c)

(Retired RU) ANA has been retired from SNA. Consult product documentation for further information and support.

BFCINIT (BF CONTROL INITIATE)

SSCP→PU T4|5, Norm; FMD NS(s)

BFCINIT requests the BF(LU) to attempt to activate, via a BIND request, a session with the specified SLU.

BFCINIT (BF CONTROL INITIATE)

Byte	Bit	Content
0– 2		X′812601′ NS header
3– 4		Element address of the initiating LU
5	0– 3	Format: X′0′ format 0 (sent in reply to BFINIT format X′0′; bytes 3–4 contain the element address of the PLU) X′1′ format 1 (sent in reply to BFINIT format X′1′; bytes 3–4 contain the element address of the SLU)
	4– 7	Reserved

BFCINIT (BF CONTROL INITIATE)

Byte	Bit	Content
6	0	Reserved
	1	Substitution source (reserved if bits 4–5 do not = 11): 0 Use the names contained in the Network Name (X′0E′) control vectors; the Network Name (X′0E′) control vector for the PLU is not to be included in the BIND (reserved if bit 6 = 0). 1 Use the names contained in the Names Substitution (X′16′) control vector; if bit 6 = 1, the Network Name (X′0E′) control vector for the PLU is to be included in the BIND.
	2	Save RSCV for BIND response indicator: 0 No RSCV is to be saved, and the BF(PLU) is to strip the RSCV from the RSP(BIND) before forwarding it to the PLU. 1 The BF(PLU) is to save a copy of the first RSCV (control vector X′2B′) on the BFCINIT for use in sending RSP(BIND) (i.e., the BF(PLU) will ensure that the RSP(BIND) has an RSCV, if possible, before sending the RSP(BIND) to the PLU).
	3	Copy RSCV to BIND request indicator: 0 The BF(PLU) should not copy any RSCV from the BFCINIT to the BIND. 1 The BF(PLU) should copy the last RSCV on the BFCINIT to the BIND and send it to the BF(SLU).
	4–5	Names substitution in BIND PLU and SLU Name fields (reserved for format 1): 00 No name substitution is to be performed by the receiver. 01 No name substitution is to be performed by the receiver, but network identifiers are present and are to be removed from the BIND; if bit 6 = 1, the Network Name (X′0E′) control vector for the PLU is to be included in the BIND. 10 No name substitution is to be performed by the receiver, but the Network Name (X′0E′) control vector for the PLU is to be included in the BIND (this value reserved if bit 6 = 0). 11 Name substitution is to be performed by the receiver: the names from the source indicated by bit 1 are to be substituted into the PLU and SLU Name fields in BIND. *Note:* Control vector X′0E′ is used if the name to be substituted is a network-qualified real name. Control vector X′16′ is used if the name to be substituted is not a network-qualified real name.
	6	0 Extended BIND is not sent to the SLU. 1 Extended BIND is sent to the SLU.
	7	Reserved
7–11		Reserved
12–13		Length, in binary, of BIND image
14 – m		BIND image: bytes 1 – p of the BIND RU, i.e., through the URC field (see BIND format description) *Note:* The URC Length field is included, even if it is set to 0.
m + 1 – n		Secondary LU Name Field
m + 1		Type: X′F3′ logical unit
m + 2		Length, in binary, of secondary LU name (values 1 to 17 are valid)
m + 3 – n		Secondary LU name *Note:* If the SLU name is network-qualified (net ID included), the PLU name in bytes k + 2 – m of the BIND image contains the network-qualified PLU name.

BFCINIT (BF CONTROL INITIATE)

Byte	Bit	Content
n+1 – p		Session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* One of the following session keys is used; they are parsed according to subfield parsing rule KL: X′0A′ URC session key (present for format 0) *Note:* The URC in this session key differs from the URC that may be provided by the ILU and that appears in the BIND image; the control vector contains a URC generated by the T4|5 boundary function to identify the session until the network address pair is provided. X′15′ Network-Qualified Address Pair session key (present for format 1)
p+1 – q		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′0D′ Mode/Class-of-Service/Virtual-Route-Identifier List control vector (not included in format 1) X′0E′ Network Name control vector: the network-qualified name of the PLU (always present); followed by another Network Name control vector containing the SLU name X′0E′ Network Name control vector: the network-qualified name of the SLU (always present); preceded by another Network Name control vector containing the PLU name X′0E′ Network Name control vector: the network-qualified name of the CP(PLU); if present, preceded by two Network Name control vectors X′15′ Network-Qualified Address Pair control vector: PLU and SLU, respectively (present if the session key in bytes n+1 – p is not X′15′) X′16′ Names Substitution control vector: contains the names to be substituted into the PLU and SLU name fields in the BIND (present if the names to be substituted are not included in the X′0E′ control vectors, but not present in format 1) X′27′ XRF Session Activation control vector X′2B′ Route Selection control vector (present for format 1 when the control point can provide it — up to two RSCVs may be present, depending on the settings of byte 6, bits 2–3) X′2C′ COS/TPF control vector (always present) X′2D′ Mode control vector (conditionally present): contains the mode name as known in the network of the SLU X′52′ Primary Send Pacing Window Size control vector (present for format 0 when the resulting BIND is negotiable; not present for format 1) X′5F′ Extended Fully Qualified PCID control vector X′60′ Fully Qualified PCID control vector (always present) X′66′ Length-Checked Compression control vector (present when the default compression level is to be overridden)

BFCLEANUP (BF CLEANUP)

SSCP→PU T4|5, Norm; FMD NS(s)

BFCLEANUP is sent with definite response requested to request that the BF(PLU) or BF(SLU) attempt to deactivate the identified session.

BFCLEANUP (BF CLEANUP)

Byte	Bit	Content
0– 2		X′812629′ NS header
3– 4		Element address of the subject LU, if assigned; otherwise, element address of the associated adjacent link station, or the element address of the PU T4|5 if no link station is associated with the session
5	0– 3	Format: X′0′ format 0 (only value defined)
	4– 7	Reserved
6– 7		Reserved
8 – n		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* The following session keys (SKs) are used; either or both may be present. When both are present, SK X′0A′ will precede SK X′15′. They are parsed according to subfield parsing rule KL: X′0A′ URC (The length of the URC Session Key Data field is set to 4.) X′15′ network-qualified address pair: PLU and SLU, respectively
n + 1 – p		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′35′ Extended Sense Data control vector (always present) X′60′ Fully Qualified PCID control vector

BFINIT (BF INITIATE)

PU T4|5→SSCP, Norm; FMD NS(s)

BFINIT from the BF(LU) requests the initiation of a session between the two LUs named in the BIND image.

BFINIT (BF INITIATE)

Byte	Bit	Content
0– 2		X′812681′ NS header
3– 4		Element address of the ALS associated with the session request
5	0– 3	Format: X′0′ format 0: Requests initiation of a session between the two LUs named in the BIND image X′1′ format 1: Requests calculation of an RSCV, which will be appended to the BIND before the BIND is forwarded to the SLU
	4– 7	Reserved
6– 7		Length, in binary, of BIND image
8 – m		BIND image: bytes 1 – s of the BIND RU (see BIND format description), i.e., through the control vectors

BFINIT (BF INITIATE)

Byte	Bit	Content
m + 1 – n		Session keys, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* One of the following session keys may be included; it is parsed according to subfield parsing rule KL: X'0A' URC session key (present for format 0) X'15' Network-Qualified Address Pair (present for format 1)
n + 1 – p		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X'23' Local-Form Session Identifier control vector (present for format 0)

BFSESSEND (BF SESSION ENDED)

PU T4|5→SSCP, Norm; FMD NS(s)

BFSESSEND notifies the SSCP that the LU-LU session identified has been deactivated.

BFSESSEND (BF SESSION ENDED)

Byte	Bit	Content
0– 2		X'812688' NS header
3– 4		Element address of the PU T4|5
5	0– 3	Format: X'0' format 0 (only value defined)
	4	LU role: 0 the subject LU is the SLU for this session 1 the subject LU is the PLU for this session
	5– 7	Reserved
6		Cause: indicates the reason for the deactivation of the identified LU-LU session (see UNBIND for values)
7		Reserved
8 – n		Session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* One of the following session keys is included; they are parsed according to subfield parsing rule KL: X'15' Network-qualified Address Pair control vector: PLU and SLU respectively
n + 1 – p		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X'35' Extended Sense Data control vector X'60' Fully Qualified PCID control vector

BFSESSINFO (BF SESSION INFORMATION)

PU T4→SSCP, Norm; FMD NS(s)

BFSESSINFO provides the SSCP with information about sessions with SSCP-independent LUs in a peripheral node taken over by the receiving SSCP.

BFSESSINFO (BF SESSION INFORMATION)

Byte	Bit	Content
0– 2		X′81268C′ NS Header
3– 4		Element address of the adjacent link station (ALS) identified in the CONTACT RU that caused the takeover procedure
5	0– 3	Format: X′0′ format 0 (only value defined)
	4– 7	Reserved
6	0	ALS takeover status: 0 ALS takeover not complete: related BFSESSINFO RUs forthcoming 1 ALS takeover complete
	1	LU takeover status: 0 LU takeover not complete: related BFSESSINFO RUs forthcoming 1 LU takeover complete
	2	Authorized LU indicator: 0 the subject LU requires system definition to receive network services 1 the subject LU does not require system definition to receive network services; it is authorized for automatic receipt of network services
	3	Static/dynamic address indicator: 0 sender considers the LU to have a static secondary address 1 sender considers the LU to have a dynamic or unassigned secondary address
	4	Static LU address status (reserved except when byte 6, bit 0 = 1): 0 addresses for unreported static LUs may have changed from those in the original load module 1 addresses for unreported static LUs are unchanged from those in the original load module
	5– 7	Reserved
Note:		The following fields may be omitted when no session information is being reported.
7– 8		Reserved
9 – m		Subject LU name (the BF is reporting session information on behalf of the LU)
9		Length, in binary, of network-qualified LU name (values 0 to 17 are valid)
10 – m		Network-qualified LU name
m + 1 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′2A′ Session Information control vector (zero or more Session Information control vectors may be present, each describing a session being reported for the independent LU)

BFSESSST (BF SESSION STARTED)

PU T4|5→SSCP, Norm; FMD NS(s)

BFSESSST informs the SSCP that a new LU-LU session has been activated and provides information about the active session.

BFSESSST (BF SESSION STARTED)

Byte	Bit	Content
0– 2		X′812686′ NS Header
3– 4		Element address of the PU T4|5
5	0– 3	Format: X′0′ format 0 (only value defined)
	4	LU role: 0 The subject LU is the SLU for this session. 1 The subject LU is the PLU for this session.
	5– 7	Reserved
6 – m		Session key, as described the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* One of the following session keys is used; it is parsed according to subfield parsing rule KL: X′15′ Network-Qualified Address Pair session key: PLU and SLU, respectively
m + 1 – n		Control vectors, as described the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′1E′ VR-ER Mapping Data control vector (always present) X′23′ Local-Form Session Identifier control vector (always present) X′2B′ Route Selection control vector X′31′ BIND Image control vector X′60′ Fully Qualified PCID control vector

BFTERM (BF TERMINATE)

PU T4|5→SSCP, Norm; FMD NS(s)

BFTERM from the BF(LU) requests that the SSCP assist in the termination of the identified LU-LU session.

BFTERM (BF TERMINATE)

Byte	Bit	Content
0– 2		X′812683′ NS header
3– 4		Element address of the PU T4|5
5	0– 3	Format: X′0′ format 0 (only value defined)
	4– 7	Reserved

BFTERM (BF TERMINATE)

Byte	Bit	Content
6		Cause: X′00′ session-activation request rejected: The boundary node has received an UNBIND or a -RSP(BIND) from the direction of the SLU for a pending-active session. X′01′ VR activation failure: The boundary node was unable to activate a virtual route (VR) from the VRID List (X′0D′) control vector included in the BFCINIT. X′02′ session-activation request terminated: The boundary node has received an UNBIND from the direction of the PLU for a pending-active session. X′03′ VR activation failure: indicates the VR on which a BIND arrived failed before session activation could complete
7		Reserved
8 – n		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* One of the following session keys is included; they are parsed according to sub-field parsing rule KL: X′0A′ URC session key X′15′ Network-Qualified Address Pair session key
n + 1 – p		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′1B′ VRID List control vector (included only when byte 6 = X′01′ as the result of detecting a failure to activate a route specified in the VRID List (X′0D′) control vector.) X′35′ Extended Sense Data control vector (conditionally present) X′60′ Fully Qualified PCID control vector (conditionally present)

BID (BID)

LU→LU, Norm; DFC

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

BID (BID)

Byte	Bit	Content
0		X′C8′ request code

BIND (BIND SESSION)

PLU→SLU, Exp; SC

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

BIND (BIND SESSION)

Byte	Bit	Content
0		X′31′ request code
1	0–3	Format: 0000 (only value defined)
	4–7	Type: 0000 negotiable (only value defined for LU 6.2) 0001 nonnegotiable
2		FM profile: X′02′ FM profile 2 X′03′ FM profile 3 X′04′ FM profile 4 X′07′ FM profile 7 X′12′ FM profile 18 X′13′ FM profile 19 (only value defined for LU 6.2)
3		TS profile: X′02′ TS profile 2 X′03′ TS profile 3 X′04′ TS profile 4 X′07′ TS profile 7 (only value defined for LU 6.2)
		FM Usage—Primary LU Protocols for FM Data
4	0	Chaining use selection: 0 only single-RU chains allowed from primary LU half-session 1 multiple-RU chains allowed from primary LU half-session (only value defined for LU 6.2)
	1	Request control mode selection: 0 immediate request mode (only value defined for LU 6.2) 1 delayed request mode
	2–3	Chain response protocol used by primary LU half-session for FMD requests; chains from primary will ask for: 00 no response 01 exception response 10 definite response 11 definite or exception response (only value defined for LU 6.2)
	4	2-phase commit for sync point (reserved if any TS profile other than 4): 0 2-phase commit not supported 1 2-phase commit supported
	5	Reserved
	6	FMH-1 SCB compression indicator (reserved for LU 6.2): 0 FMH-1 SCB compression will not be used on requests from primary 1 FMH-1 SCB compression may be used
	7	Send End Bracket indicator: 0 primary will not send EB (only value defined for LU 6.2) 1 primary may send EB

BIND (BIND SESSION)

Byte	Bit	Content
		FM Usage—Secondary LU Protocols for FM Data
5	0	Chaining use selection: 0 only single-RU chains allowed from secondary LU half-session 1 multiple-RU chains allowed from secondary LU half-session (only value defined for LU 6.2)
	1	Request control mode selection: 0 immediate request mode (only value defined for LU 6.2) 1 delayed request mode
	2– 3	Chain response protocol used by secondary LU half-session for FMD requests; chains from secondary will ask for: 00 no response 01 exception response 10 definite response 11 definite or exception response (only value defined for LU 6.2)
	4	2-phase commit for sync point (reserved if any TS profile other than 4): 0 2-phase commit not supported 1 2-phase commit supported
	5	Reserved
	6	FMH-1 SCB compression indicator (reserved for LU 6.2): 0 FMH-1 SCB compression will not be used on requests from secondary 1 FMH-1 SCB compression may be used
	7	Send End Bracket indicator: 0 secondary will not send EB (only value defined for LU 6.2) 1 secondary may send EB
		FM Usage—Common LU Protocols
6	0	Whole-BIUs required indicator (reserved in nonextended, non-LU 6.2 BINDs, i.e., when control vector X′60′ is not present): 0 the sending node supports receipt of segments on this session 1 the sending node does not support receipt of segments on this session; the maximum send-RU size specified in bytes 10 and 11 of BIND and RSP(BIND) are negotiated so that BIUs on this session are not segmented when sent to a node requiring whole BIUs
	1	FM header usage: 0 FM headers not allowed 1 FM headers allowed (only value defined for LU 6.2)
	2	Brackets usage and reset state: 0 The value of this bit should be 0 if either condition (1) or condition (2) is true. 1. Brackets are not used if neither primary nor secondary will send EB (byte 4, bit 7 = 0 and byte 5, bit 7 = 0). 2. Brackets are used and the bracket state managers′ reset states are INB if: • either primary or secondary, or both, may send EB (byte 4, bit 7 = 1 or byte 5, bit 7 = 1). • FM profile 19 is specified (byte 2 = X′13′). (only value defined for LU 6.2) 1 brackets are used and bracket state managers′ reset states are BETB

BIND (BIND SESSION)

Byte	Bit	Content
	3	Bracket termination rule selection; byte 4, bit 7 = 0, and byte 5, bit 7 = 0; and if FM profile is not 19): 0 Rule 2 (unconditional termination) will be used during this session 1 Rule 1 (conditional termination) will be used during this session (only value defined for LU 6.2) *Note:* This bit is reserved if both of the following conditions are true. 1. Brackets are not used (byte 4, bit 7 = 0, byte 5, bit 7 = 0 , and byte 6, bit 2 = 0). 2. The FM profile is not 19 (byte 2 ≠ X′13′).
	4	Alternate code set allowed indicator: 0 alternate code set will not be used 1 alternate code set may be used
	5	Sequence number availability for sync point resynchronization (reserved if any TS profile other than 4 is used): 0 sequence numbers not available 1 sequence numbers available *Note:* Sequence numbers are transaction processing program sequence numbers from the previous activation of the session with the same session name; they are associated with the last acknowledged requests and any pending requests to commit a unit of work. If no previous activation existed, the numbers are 0, and this bit is set to 0.
	6	BIS sent (reserved for TS profiles other than 4): 0 BIS not sent 1 BIS sent
	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 11 reserved *Note:* Values 00 and 10 are the only values defined for LU 6.2.
	2	Recovery responsibility: 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) *Note:* This bit is reserved if the normal-flow send/receive mode is full-duplex and the FM profile is not 19 (byte 2 ≠ X′13′).
	3	Contention winner/loser: 0 secondary is contention winner and primary is contention loser 1 primary is contention winner and secondary is contention loser

BIND (BIND SESSION)

Byte	Bit	Content
		Note 1: Contention winner is also brackets first speaker. *Note 2:* This bit is reserved if either condition (1) or condition (2) holds. 1. The normal-flow send/receive mode is full-duplex and the FM profile is not 19 (byte 2 ≠ X′13′). 2. All of the following are true. • The normal flow send/receive mode is HDX-FF (byte 7, bits 0–1 = 10). • Brackets are not used (byte 4, bit 7 = 0, byte 5, bit 7 = 0 , and byte 6, bit 2 = 0). • The FM profile is not 19 (byte 2 ≠ X′13′). • Symmetric responsibility for recovery is used (byte 7, bit 2 = 1).
	4–5	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 either condition (1) or conditions (2) *and* (3) hold. 1. The FM profile is 19. 2. The normal-flow send/receive mode is half-duplex flip-flop (byte 7, bits 0–1 = 10). 3. Brackets are not used or the bracket state manager′s reset state is INB (byte 6, bit 2 = 0). TS Usage
8	0	Staging indicator for session-level pacing of the secondary-to-primary normal flow: 0 the secondary send window size (byte 8, bits 2–7) and the primary receive window size (byte 13, bits 2–7) are for one-stage (or APPN hop-by-hop) pacing (The secondary send window size is always equal to the primary receive window size.) 1 the secondary send window size (byte 8, bits 2–7) and the primary receive window size (byte 13, bits 2–7) are for two-stage pacing *Note:* The meanings of 0 and 1 are reversed from the corresponding staging indicator for the primary-to-secondary normal flow.
	1	Reserved
	2–7	Secondary send window size, in binary, for session-level pacing: a value of 0 indicates that there will be no pacing of requests flowing from the secondary. *Note:* If pacing on a session stage in a particular direction is not to be performed, the values for the window size on that stage are set to 0. For example, if there is to be no pacing in the secondary to primary direction, the primary receive and secondary send window sizes are both set to 0.

BIND (BIND SESSION)

Byte	Bit	Content
9	0	Adaptive session-level pacing support: 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 bits 2–7 of bytes 8 and 13 specifies no pacing in the secondary-to-primary direction; a 0 value in bits 2–7 of bytes 9 and 12 specifies the maximum window size is requested 1 adaptive pacing supported by the sending node: pacing window values in bits 2–7 of bytes 8, 9, 12, and 13 specify the *preferred minimum value* for each ISOLATED PACING MESSAGE; a 0 value specifies that the preferred minimum value is as large as possible; each adaptive pacing partner initializes its own send window size to 1 at session activation *Note:* If the PLU specifies adaptive pacing in BIND, and the SLU is able to support adaptive pacing, the SLU responds with this bit set to 1 in RSP(BIND). If the PLU indicates it does not support adaptive pacing, or if the SLU does not support adaptive pacing, this bit will be set to 0 in RSP(BIND). See Chapter 5, “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 (BF) 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 from the BF 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, $a \geq 8$ and therefore X′ab′ is a normalized floating point representation.) See Figure 6-1 on page 6-36 for all possible values.
11		Maximum RU size sent on the normal flow by the primary half-session: identical encoding as described for byte 10
12	0	Staging indicator for session-level pacing of the primary-to-secondary normal flow: 0 the primary send window size (byte 12, bits 2–7) and the secondary receive window size (byte 9, bits 2–7) are for two-stage pacing 1 the primary send window size (byte 12, bits 2–7) and the secondary receive window size (byte 9, bits 2–7) are for one-stage (or APPN hop-by-hop) pacing (The primary send window size is always equal to the secondary receive window size.) *Note:* The meanings of 0 and 1 are reversed from the corresponding staging indicator for the secondary-to-primary normal flow (byte 8, bit 0).
	1	Reserved
	2–7	Primary send window size, in binary, for session-level pacing: a value of 0 causes the value set by a system definition pacing parameter (if the system definition includes such a parameter) to be assumed for the session; if this is also 0, it means no pacing of requests flowing from the primary (For one-stage pacing in the primary-to-secondary direction, this field is redundant with, and will indicate the same value as, the secondary receive window size—see byte 9, bits 2–7, above.)
13	0–1	Reserved

BIND (BIND SESSION)

Byte	Bit	Content
	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)
16–21		Reserved
22	0	Extended security mechanism support: 0 Extended security mechanisms are not supported. 1 At least one extended security mechanism is supported. (And a structured data subfield X′14′ is present.)
	1	Security extended sense data support: 0 This RU sender does not support receipt of sense data values in the range X′080FFF00′ to X′080FFFFF′. 1 This RU sender supports receipt of sense data values in the range X′080FFF00′ to X′080FFFFF′ (and it does not automatically send UNBIND for the session if it does not understand a sense data value that it receives in this range).
	2–7	Reserved
23		Security Support Indicators
	0–2	Retired
	3	Conversation-level security support: 0 Access Security Information field will not be accepted on incoming FMH-5s. 1 Access Security Information field will be accepted on incoming FMH-5s.
	4	LU-LU verification protocol supported: 0 basic verification protocol 1 enhanced verification protocol
	5	Password substitution support: 0 not supported 1 supported
	6	Already-verified function support: 0 Already-Verified indicator will not be accepted on incoming FMH-5s. 1 Already-Verified indicator will be accepted on incoming FMH-5s.
	7	Persistent verification capability: 0 Persistent Verification indicator is not supported on incoming FMH-5s. 1 Persistent Verification indicator is supported on incoming FMH-5s.

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 session indicator: 0 not a limited-resource session and thus the contention-winner LU will not deactivate it when it is no longer busy 1 a limited-resource session and thus the contention-winner LU will deactivate it when it is no longer busy
	2	Indicates whether the NNS receiving this BIND should send its conwinner BIND over the same TG that this BIND was received on. This indicator is optionally set by ENs and is optionally supported by receiving NNSs. 0 No - it is not desired that the NNS send its BIND over the same TG 1 Yes - it is desired that the NNS send its BIND over the same TG
	3– 5	Reserved
	6– 7	Length-checked compression options: 00 no compression 01 compression supported by the PLU — compression to be determined by the SLU *Note:* This value indicates that the PLU supports compression but the PLU does not have any information that would indicate that this session would benefit from compression. 10 reserved 11 compression requested by the PLU *Note:* This value indicates that the PLU both supports and desires compression on this session. *Note 1:* On extended BINDs, the Length-Checked Compression (X′66′) control vector serves to define the requested compression options. *Note 2:* Bits 6-7, as defined, apply also to LU types 0, 1, 2, 3, and 6.1.
		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

BIND (BIND SESSION)

Byte	Bit	Content
	2–3	Session-level cryptography options: 00 no session-level cryptography supported 01 session-level selective cryptography supported; all cryptography key management is supported by the SSCP and LU; exchange (via +RSP(BIND)) and verification (via CRV) of the cryptography session-seed value is supported by the LUs for the session; all FMD requests carrying ED are enciphered/deciphered by the TCs 10 reserved 11 session-level mandatory cryptography supported; all cryptography key management is supported by the SSCP and LU; exchange (via +RSP(BIND)) and verification (via CRV) of the cryptography session-seed value is supported by the LUs for the session; all FMD requests are enciphered/deciphered by TC
	4–7	Session-level cryptography options field length: X′0′ no session-level cryptography specified; following additional cryptography options fields (bytes 27 – k) omitted X′9′ session-level cryptography specified; additional options follow in next nine bytes
27	0–1	Session cryptography key encipherment method: 00 session cryptography key enciphered under SLU master cryptography key using a seed value of 0 (only value defined)
	2–4	Reserved
	5–7	Cryptography cipher method: 000 block chaining with seed and cipher text feedback, using the Data Encryption Standard (DES) algorithm 001 block chaining with seed and cipher text feedback, using the Triple Data Encryption Standard (Triple DES) algorithm
28 – k		Session cryptography key enciphered under secondary LU master cryptography key; an 8-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FMD requests. When the cryptography cipher method is Triple DES (byte 27, bits 5–7 set to 001) this field contains the first 8 bytes of the Triple DES cryptography key enciphered under the secondary LU master cryptography key. The last 16 bytes of the Triple DES key are contained in the appended Triple DES Cryptography Key Continuation (X′71′) control vector.
k+1 – m		Network Services (NS) Primary LU Name Field (always present) This parameter is always network-qualified for implementations at the current level of SNA, unless the name is uninterpreted; uninterpreted names and back-level implementations may omit the network ID.
k+1		Length of primary LU name (values 1 to 17 are valid) *Note:* Value 0 is retired.
k+2 – m		Primary LU name or, if the secondary LU issued an INIT-SELF (or INIT-OTHER), the uninterpreted name as carried in that RU (and also in CDINIT for a cross-domain session)
m+1 – n		User Data Field
m+1		Length of user data *Note:* X′00′ = no User Data field present; if unstructured user data present, values 1 to 65 are valid.
m+2 – n		User Data
m+2		User data key: X′00′ Structured subfields follow (only value defined for LU 6.2). *Note:* Individual structured subfields may be omitted entirely. When present, they appear in ascending subfield-number order. ¬ X′00′ First byte of unstructured user data.

BIND (BIND SESSION)

Byte	Bit	Content
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 8, "User Data Structured Subfields.")
n + 1 – p		User Request Correlation Field (present only if carried in INIT from SLU, or if Secondary LU name field or control vectors are included)
n + 1		Length of user request correlation (URC) field (values 0 to 12 are valid) *Note:* X′00′ = no URC present.
n + 2 – p		URC: LU-defined identifier (present only if carried in INIT from SLU)
p + 1 – r		Network Services (NS) Secondary LU Name Field (always present for negotiable BINDs; optionally present for nonnegotiable BINDs that include control vectors; otherwise, omitted) This parameter is always network-qualified for implementations at the current level of SNA; back-level implementations may omit the network ID.
p + 1		Length of secondary LU name (values 1 to 17 are valid) *Note:* Value 0 is retired.
p + 2 – r		Secondary LU name

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

BIND (BIND SESSION)

Byte	Bit	Content
r+1 – s		Control vectors, as described in "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields"). X'0E' Network Name control vector: PLU network name, X'F3' (present in extended BINDs when bytes k+2 – m contain a name that is not network-qualified, such as an uninterpreted name) X'0E' Network Name control vector: network-qualified CP name, X'F4' (present in extended BINDs sent from nodes at the current level of SNA for PLU-inititated sessions when the Fully Qualified PCID (X'60') control vector does not contain the real CP(PLU) name and the real CP(PLU) name is known) X'27' XRF Session Activation control vector *Note:* Control vector X'27' specifies that an XRF-active or XRF-backup session is to be activated. X'2B' Route Selection control vector (present in extended BINDs when the BIND sender has the information available as the result of a Locate search and the session-route calculation) X'2C' COS/TPF control vector (present in extended BINDs when the BIND sender supports mode-to-COS mapping or when it received the control vector on a Locate search reply) X'2D' Mode control vector (present in non-LU6.2 extended BINDs unless the default mode name — eight X'40' characters — is intended) X'60' Fully Qualified PCID control vector (in which case the BIND is called an *extended BIND*) X'66' Length-Checked Compression control vector (present in extended BINDs when compression is supported) X'71' Triple DES Cryptography Key Continuation control vector. When concatenated with the first 8 bytes (contained in bytes 28-k, where k equals 35) the complete 24-byte enciphered (under the secondary LU master cryptography key) Triple DES session cryptography key is formed. Control vector 71 is included on BIND when the cryptography cipher method indicates Triple DES (i.e., byte 27, bits 5–7 are set to 001). *Note:* The receiving LU simply ignores unrecognized control vectors.
Note:		The length of the BIND RU cannot exceed 256 or 512 bytes. The length of the basic BIND RU is restricted to 256 bytes including the X'0E', X'2C', X'2D', and X'60' control vectors; any additional control vectors may cause the length to increase up to 512 bytes.
Note:		If the last byte of a Format 0 BIND request not having control vectors is a length field and that field is 0, that byte may be omitted from the BIND request.

Exponent (b)	Mantissa (a)							
	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 6-1. RU Sizes Corresponding to Values X′ab′ in BIND

BINDF (BIND FAILURE)

PLU→SSCP, Norm; FMD NS(s)

BINDF is sent, with no-response requested, by the PLU to notify the SSCP that the attempt to activate the session between the specified LUs has failed.

BINDF (BIND FAILURE)

Byte	Bit	Content
0–2		X′810685′ NS header

BINDF (BIND FAILURE)

Byte	Bit	Content
3– 6		Sense data
7		Reason (a bit is set to 1 if the indicated error occurs):
	0	Error at PLU
	1	BIND error in reaching SLU
	2	Setup reject at PLU
	3	Setup reject at SLU
	4– 7	Reserved
8 – m		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* One of the following session keys is used: X′07′ Network Address Pair (retired): PLU and SLU, respectively X′15′ Network-Qualified Address Pair: PLU and SLU, respectively
m + 1 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′35′ Extended Sense Data control vector X′60′ Fully Qualified PCID control vector

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

CDCINIT (CROSS-DOMAIN CONTROL INITIATE)

SSCP→SSCP, Norm; FMD NS(s)

CDCINIT passes information about the SLU from the SSCP(SLU) to the SSCP(PLU) and requests that the SSCP(PLU) send CINIT to the PLU.

CDCINIT (CROSS-DOMAIN CONTROL INITIATE)

Byte	Bit	Content
0– 2		X′81864B′ NS header
3	0– 3	Format X′0′ Format 0: session pair identified by network addresses X′1′ Format 1: session pair identified by a session key
	4– 7	Reserved
4	0– 6	Reserved
	7	0 BIND with XRF Session Activation control vector (X′27′) is not supported by the SLU 1 BIND with XRF Session Activation control vector (X′27′) is supported by the SLU
5– 12		PCID: a unique value used as a session identifier (retired when the Fully-Qualified PCID (X′60′) control vector is included)
13 – m		Session Pair Identifier
For format 0:		
13– 14		Network address of PLU
15– 16 (= m)		Network address of SLU
For format 1:		
13 – m		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* The following session key is used: X′15′ Network-Qualified Address Pair: PLU and SLU, respectively (only value defined)
m + 1 – m + 2		Length, in binary, of BIND image
m + 3 – n		BIND image: bytes 1 – p of the BIND RU (see BIND format description), i.e., through the URC field *Note:* For SLUs not in the sending SSCP′s node, the session cryptography key is enciphered under the SLU master cryptography key; for SLUs in the SSCP′s node, the sending SSCP enciphers the session cryptography key under a dummy SLU master cryptography key.
n + 1 – n + 2		Length, in binary, of LU or non-SNA device characteristics field and format – i.e., bytes n + 3 – p (X′00′ = no characteristics/format field.)

CDCINIT (CROSS-DOMAIN CONTROL INITIATE)

Byte	Bit	Content
n + 3		LU or non-SNA device characteristics format: X′01′ Format 1: access method unique device characteristics (only value defined)
n + 4 – p		LU or non-SNA device specifications (See CINIT for the format of this field.)
p + 1		Length, in binary, of session cryptography key *Note:* X′00′ = no Session Cryptography Key field is present.
p + 2 – q		Session cryptography key for primary: the session cryptography key, enciphered under the cross-domain cryptography key defined for the SSCP(SLU) to SSCP(PLU) direction (a different cross-domain cryptography key is defined for the opposite direction) and using a seed value of 0
q + 1 – r		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′3F′ SSCP(SLU) Capabilities control vector: present if the SSCP(SLU) supports suppressing notifying the SLU of specific session initiation errors. X′60′ Fully Qualified PCID control vector (always present) X′66′ Length-Checked Compression control vector (present when compression is supported) X′68′ XRF/Session Cryptography control vector (present when session cryptography is required on an XRF backup session)

CDINIT (CROSS-DOMAIN INITIATE)

SSCP→SSCP, Norm; FMD NS(s)

CDINIT requests another SSCP to assist in initiating an LU-LU session for the specified (OLU,DLU) pair.

CDINIT (CROSS-DOMAIN INITIATE)

Byte	Bit	Content
0– 2		X′818641′ NS header
3	0– 3	Format: X′0′ Format 0: retired X′1′ Format 1: used when Type = DQ and specifies a subset of the parameters; Format 1 includes bytes 0 through 18 X′2′ Format 2: specifies COS fields and an additional OLU status (byte 6, bit 5) in addition to the parameters in Format 0; Format 2 includes bytes 0 through (s+9) X′3′ Format 3: used when Type = I, I/Q, or Q; includes bytes 0 through (s+9) as in Format 2; control vectors are appended following byte (s+9) X′4′ Format 4: used when Type = DQ and specifies a subset of the parameters; Format 4 includes Format 1, plus X′15′ Network-Qualified Address Pair session key and various control vectors X′5′ Format 5: sent across a VR-based TG to obtain necessary FID4 routing information prior to sending a BIND over that VR-based TG; specifies a subset of Format 3 parameters as shown below
	4– 7	Reserved

CDINIT (CROSS-DOMAIN INITIATE)

Byte	Bit	Content
Formats 0, 2, and 3 continue (see the continuation of the other formats further below)		
4	0– 1	Type: 00 reserved 01 initiate only (I) 10 queue only (Q) *Note:* This setting is used when the SSCP(OLU) has already decided to enqueue the request (see byte 6). 11 initiate or queue (I/Q)
	2– 3	Retired
	4	Network-qualified names support indicator: 0 a BIND for this session sent or received in the domain of the sending SSCP will not contain network-qualified LU names in bytes k+2 – m and p+2 – r 1 a BIND for this session sent or received in the domain of the sending SSCP may contain network-qualified LU names in bytes k+2 – m and p+2 – r
	5	Retired
	6	0 DLU is PLU 1 OLU is PLU
	7	Reserved
5		Queuing Conditions For DLU (reserved when Type = I)
	0	0 do not queue if session limit exceeded 1 queue if session limit exceeded
	1	0 do not queue if DLU is not currently able to comply with the PLU/SLU specification (as given in byte 4, bit 6) 1 queue if DLU is not currently able to comply with the PLU/SLU specification
	2	Reserved
	3	Retired
	4	Reserved
	5– 6	Queuing position/service: 00 retired 01 enqueue this request FIFO; i.e., the request will be dequeued after the requests already in the queue. 10 enqueue this request LIFO; i.e., the request will be dequeued before the requests already in the queue. 11 reserved
	7	Reserved *Note:* Queuing is not done if the DLU is unknown, or if the domain of the DLU is in takedown status.
6		OLU status:
	0	0 OLU does not support extended BIND RU 1 OLU supports extended BIND RU
	1	0 LU is not available 1 LU is available
	2– 3	(used if LU is not available; otherwise, reserved) 00 LU session limit exceeded 01 reserved 10 LU is not currently able to comply with the PLU/SLU specification 11 reserved
	4	Retired
	5	(reserved in format 0) 0 UNBIND and SESSEND cannot be sent by the LU or by its boundary function (retired) 1 UNBIND and SESSEND may be sent by the LU or by its boundary function

CDINIT (CROSS-DOMAIN INITIATE)

Byte	Bit	Content
	6–7	01 OLU is PLU 10 OLU is SLU
7–14		PCID: a unique value used as a session identifier (retired when the Fully-Qualified PCID (X′60′) control vector is included)
15–16		Network address of OLU (retired for format 3)
17–18		Reserved
19	0	INITIATE origin: 0 ILU is OLU 1 ILU is not OLU
	1–2	Reserved
	3	Retired
	4–5	Reserved
	6	SLU support of XRF indicator: 0 SLU does not support XRF 1 SLU does support XRF
	7	XRF backup session request indicator: 0 backup session not requested 1 backup session requested
20		NOTIFY specification:
	0–1	NOTIFY (Resource Requested) condition: 00 Do not send NOTIFY to LUs in session with DLU. 01 reserved 10 Send NOTIFY to the LU in session with DLU only if the CDINIT request is queued for session limit. 11 reserved
	2–6	Reserved
	7	0 Do not send NOTIFY when DLU is available. 1 Send NOTIFY when DLU is available.
21–28		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 to be used by the SSCP(PLU) to build the CINIT request *Note:* For format 3 (cross-network), this mode name represents the mode name as known in the network of the OLU.
29 – m		Network Name of DLU
29		Type: X′F3′ logical unit
30		Length, in binary, of symbolic name
31 – m		Symbolic name, in EBCDIC characters
m + 1 – m + 2		Retired: set to X′0000′
m + 3 – q		User Field
m + 3		Length, in binary, of user data *Note:* X′00′ = no user data is present.
m + 4 – q		User data: user-specific data that is passed to the primary LU on the CINIT request

CDINIT (CROSS-DOMAIN INITIATE)

Byte	Bit	Content
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 – q		Remainder of unstructured user data
For structured user data		
m + 5 – q		Structured subfields (For detailed definitions, see Chapter 8, “User Data Structured Subfields.”)
q + 1 – r		Network Name of OLU
q + 1		Type: X′F3′ logical unit
q + 2		Length, in binary, of symbolic name
q + 3 – r		Symbolic name in EBCDIC characters
r + 1 – s		Uninterpreted Name of DLU
r + 1		Type: X′F3′ logical unit
r + 2		Length, in binary, of DLU name *Note:* X′00′ = no uninterpreted name is present.
r + 3 – s		EBCDIC character string; when present, this name is obtained from the preceding INIT-SELF or INIT-OTHER (when ILU=SLU)
End of Format 0; Formats 2 and 3 continue below		
s + 1		COS name initialization indicators:
	0	0 COS name not received from ILU (see bits 1–2) 1 COS name received from ILU
	1–2	(reserved if byte s+1, bit 0 = 1) 01 SSCP(DLU) is to initialize COS name (DLU is SLU) 10 SSCP(OLU) has initialized COS name (OLU is SLU)
	3–7	Reserved
s + 2 – s + 9		COS name (If byte s+1, bit 0 = 0 and bits 1–2 = 01, this field carries unpredictable values and is not used): symbolic name of class of service in EBCDIC characters *Note:* For format 3 (cross-network) this COS name represents the COS name as known in the network of the OLU.
End of Format 2; Format 3 continues below		
s + 10 – t		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. When present, the control vectors specified up through the third X′19′ control vector appear in the order specified below.

CDINIT (CROSS-DOMAIN INITIATE)

Byte	Bit	Content	
		X′1A′	NAU Address control vector: contains the OLU network address (always present) *Notes:* 1. Between gateways, the OLU address is an address recognized in the subnetwork on the DLU side of the sending gateway. Within a gateway, the OLU address is an address recognized in the network on the OLU side of the gateway node, until the SSCP with address alias responsibility is reached. This SSCP replaces the received address with an address recognized in the network on the DLU side of the gateway node. 2. The network ID is identified in the X′19′ control vector for the OLU.
		X′14′	Session Initiation control vector (always present)
		X′19′	Resource Identifier control vector for DLU (always present)
		X′19′	Resource Identifier control vector for OLU (always present)
		X′19′	Resource Identifier control vector for the ILU of the third-party initiated session (present only on third-party initiated sessions)
		X′2B′	Route selection control vector
		X′2C′	COS/TPF control vector
		X′2F′	LU Definition control vector (present if data is available to be sent and if the CDINIT flows from the SSCP(SLU))
		X′2F′	LU Definition control vector (present only when immediately preceded by the previous X′2F′ control vector and model name or associated LU data is available)
		X′31′	Bind Image control vector
		X′34′	LU Definition Override control vector (present if model terminal support override values are available)
		X′3E′	Directory Entry Characteristic control vector
		X′3F′	SSCP(SLU) Capabilities control vector
		X′59′	Installation-Defined CDINIT Data control vector (present when an SSCP exit is invoked)
		X′5A′	Session Services Extensions Support control vector
		X′5B′	Interchange Node Parameters control vector
		X′5C′	APPN Message Transport control vector
		X′5F′	Extended Fully Qualified PCID control vector (conditionally present)
		X′60′	Fully Qualified PCID control vector (always present)
		X′63′	Cryptography Key Distribution control vector (present only when key distribution information is to be transferred)
		X′64′	TCP/IP Information control vector (present to forward SLU TCP/IP information to the SSCP(PLU) if that information was provided by the SLU)
		X′66′	Length-Checked Compression control vector
		X′68′	XRF/Session Cryptography control vector
		X′69′	Switched Parameters control vector

End of Format 3; Formats 1 and 4 continue below

4		Type:	
	0–1	00	dequeue (DQ)
	2–3	00	leave on queue if session limit exceeded or the LU is unable to comply with the PLU/SLU specification in bit 6
		01	remove from queue if dequeue retry is unsuccessful (retired)
		10	do not retry—remove from queue (retired)
		11	reserved
	4	Reserved	
	5	Retired	
	6	0	LU2 is PLU
		1	LU2 is SLU
	7	Reserved	

CDINIT (CROSS-DOMAIN INITIATE)

Byte	Bit	Content
5		Queuing Status (For LU associated with SSCP sending CDINIT(DQ))
	0–4	Reserved
	5–6	00 retired 01 enqueued request FIFO 10 enqueued request LIFO 11 reserved
	7	Reserved
6		LU Status (For LU associated with SSCP sending CDINIT(DQ))
	0	Reserved
	1	Retired
	2–5	Reserved
	6–7	Retired
7–14		PCID: a unique value used as a session identifier (retired for format 4) *Note:* This PCID is the same as in the original CDINIT request.
15–16		Network address of LU1 (retired for format 4)
17–18		Network address of LU2 (retired for format 4)
End of Format 1; Format 4 continues below		
19 – n		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* One of the following session keys is used: X′15′ Network-Qualified Address Pair: LU1 and LU2, respectively (only value defined)
n+1 – p		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′2B′ Route selection control vector X′31′ Bind Image control vector X′3F′ SSCP(SLU) Capabilities control vector X′5C′ APPN Message Transport control vector X′60′ Fully Qualified PCID control vector (always present) *Note:* This PCID is the same as in the original CDINIT request. X′63′ Cryptography Key Distribution control vector (present only when key distribution information is to be transferred) X′64′ TCP/IP Information control vector (present to forward SLU TCP/IP information to the SSCP(PLU) if that information was provided by the SLU) X′65′ Device Characteristics control vector X′68′ XRF/Session Cryptography control vector
End of Formats 1 and 4; Format 5 continues below		
4		Type indicators:
	0–1	01 Initiate only. 10 Initiate or queue.
	2	1 Queue if session count exceeded.
	3	1 Queue if not enabled.
	4–5	Queuing discipline: 01 FIFO 10 LIFO
	6	1 PLU supports network-qualified names.
	7	1 PLU supports extended BIND.

CDINIT (CROSS-DOMAIN INITIATE)

Byte	Bit	Content
5		Reserved
6– 1 3		Mode name to be used for the session (left-justified and padded on the right with X′40′ characters if necessary) *Note:* This field represents the mode name known in the network of the PLU.
14 – n		Control vectors, as described in "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. The control vectors specified through the second X′19′ control vector appear in the order shown below. X′1A′ NAU Address control vector (always present): contains the PLU network address X′19′ Resource Identifier control vector for SLU (always present) X′19′ Resource Identifier control vector for PLU (always present) X′2B′ Route Selection control vector (always present): represents the session path for the BIND triggering this flow X′2C′ COS/TPF control vector (always present) X′60′ Fully Qualified PCID control vector (always present): obtained from the BIND triggering this flow

CDSESSEND (CROSS-DOMAIN SESSION ENDED)

SSCP(PLU) ←→ SSCP(SLU), Norm; FMD NS(s)

CDSESSEND notifies the SSCP that the LU-LU session identified by the session key has been successfully deactivated, or that knowledge of session deactivation has been lost due to session outage with one or more of the participating LUs, gateway nodes, or adjacent SSCPs. In the latter case, the session may still be active, but explicit notification of session deactivation is no longer possible.

CDSESSEND (CROSS-DOMAIN SESSION ENDED)

Byte	Bit	Content
0– 2		X′818648′ NS header
3– 1 0		PCID: a unique value used as a session identifier (retired when the Fully-Qualified PCID (X′60′) control vector is included) *Note:* PCID is used in CDSESSEND only to aid in PIU trace correlation.
11	0– 3 4– 7	Format: X′0′ (only value defined) Reserved
12 – n		Session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* One of the following session keys is used: X′05′ PCID X′06′ Network Name Pair session key (retired): PLU and SLU, respectively X′07′ Network Address Pair session key: PLU and SLU, respectively X′15′ Network-Qualified Address Pair session key: PLU and SLU, respectively

CDSESSEND (CROSS-DOMAIN SESSION ENDED)

Byte	Bit	Content
n + 1 – p		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *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 SSCP, because of loss of sessions to one or more participating LUs, gateway nodes, or adjacent SSCPs, can no longer learn directly that the LU-LU session referenced by the CDSESSEND has ended) X′60′ Fully Qualified PCID control vector (always present)

CDSESSSF (CROSS-DOMAIN SESSION SETUP FAILURE)

SSCP(PLU)→SSCP(SLU), Norm; FMD NS(s)

(Retired RU) CDSESSSF has been retired from SNA. Consult product documentation for further information and support.

CDSESSST (CROSS-DOMAIN SESSION STARTED)

SSCP(PLU)→SSCP(SLU), Norm; FMD NS(s)

CDSESSST notifies the SSCP(SLU) that the LU-LU session identified by the Session Key Content field and the specified PCID for the initiation procedure has been successfully activated.

CDSESSST (CROSS-DOMAIN SESSION STARTED)

Byte	Bit	Content
0– 2		X′818646′ NS header
3– 10		PCID: a unique value used as a session identifier (retired when the Fully-Qualified PCID (X′60′) control vector is included)
11		Reserved
12 – n		Session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* One of the following session keys is used: X′06′ Network Name Pair session key (retired): PLU and SLU, respectively X′07′ Network Address Pair session key: PLU and SLU, respectively X′15′ Network-Qualified Address Pair session key: PLU and SLU, respectively
n + 1 – p		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′3E′ Directory Entry Characteristic control vector X′5C′ APPN Message Transport control vector X′60′ Fully Qualified PCID control vector (always present)

CDSESSTF (CROSS-DOMAIN SESSION TAKEDOWN FAILURE)

SSCP(PLU)→SSCP(SLU), Norm; FMD NS(s)

(Retired RU) CDSESSTF has been retired from SNA. Consult product documentation for further information and support.

CDTAKED (CROSS-DOMAIN TAKEDOWN)

SSCP→SSCP, Norm; FMD NS(s)

CDTAKED initiates a procedure to cause the takedown of all cross-domain LU-LU sessions (active, pending-active, and queued) involving the domains of both the sending and receiving SSCP. It also prevents the initiation of new LU-LU sessions between these domains.

CDTAKED (CROSS-DOMAIN TAKEDOWN)

Byte	Bit	Content
0–2		X′818649′ NS header
3–10		PCID: a unique value used as a session identifier
11	0–1	Type LU-LU sessions: 00 active and pending-active sessions 01 active, pending-active, and queued sessions 10 queued only sessions 11 pending-active and queued sessions
	2–3	Takedown: 00 reserved 01 orderly 10 forced 11 cleanup (mutual procedure)
	4	SSCP-SSCP session termination: 0 SSCP-SSCP session will remain active 1 SSCP-SSCP session will be terminated when the sessions indicated by the Type LU-LU Sessions field (byte 11, bits 0–1) have been terminated
	5–7	Reserved
12		Reason:
	0	0 network user 1 network manager
	1	0 normal 1 abnormal
	2–7	Reserved

CDTAKEDC (CROSS-DOMAIN TAKEDOWN COMPLETE)

SSCP→SSCP, Norm; FMD NS(s)

Except when the Cleanup option was specified, the SSCP that received CDTAKED (and responded positively to it) sends CDTAKEDC upon completion of its domain takedown procedure. The other SSCP, after completing its domain takedown procedure and receiving a CDTAKEDC, also sends a CDTAKEDC.

CDTAKEDC (CROSS-DOMAIN TAKEDOWN COMPLETE)

Byte	Bit	Content
0–2		X′81864A′ NS header
3–10		PCID: a unique value used as a session identifier
11		Type: X′01′ summary (only value defined)
12		Status: X′01′ all sessions successfully taken down (only value defined)

CDTERM (CROSS-DOMAIN TERMINATE)

SSCP→SSCP, Norm; FMD NS(s)

CDTERM requests that the receiving SSCP assist in the termination of the cross-domain LU-LU session identified by the Session Key and the Type byte of the RU. Each SSCP executes that portion of termination processing that relates to the LU in its domain.

CDTERM (CROSS-DOMAIN TERMINATE)

Byte	Bit	Content
0–2		X′818643′ NS header
3	0–3	0000 Format 0 (only value defined)
	4–7	Reserved
4		Type:
	0–1	00 request applies to active and pending-active sessions 01 request applies to active, pending-active, and queued sessions 10 request applies to queued sessions only 11 request applies to pending and queued sessions only
	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 DLU; another session initiation request will be sent for DLU 1 send DACTLU to DLU when appropriate; no further session initiation request will be sent (from this sender) for DLU
	4	Reserved

CDTERM (CROSS-DOMAIN TERMINATE)

Byte	Bit	Content
	5–6	Retired
	7	0 orderly or forced (see byte 4, bit 2) 1 cleanup
5–12		PCID: a unique value used as a session identifier *Note:* This PCID is used in CDTERM only to aid in PIU trace correlation.
13		Reason:
	0	0 network user 1 network manager
	1	0 normal 1 abnormal
	2	Reason code required; i.e., the CDTERM RU reports a session setup or takedown failure detected by a different NAU from the one that originated the CDTERM: 0 reason code not required 1 reason code required *Note:* reserved if byte 13, bit 1 = 0 (normal)
	3	0 session setup failure 1 session takedown failure
	4–7	Reason code (copied from CDTERM or BINDF [if byte 13, bit 3 indicates session setup failure] or from CDTERM or UNBINDF [if byte 13, bit 3 indicates session takedown failure]):
	4	1 CINIT or CTERM error in reaching PLU
	5	1 BIND or UNBIND error in reaching SLU
	6	1 setup or takedown reject at PLU
	7	1 setup reject at SLU *Note:* Bits 3–7 are reserved if byte 13, bit 2 indicates that the reason code is not required.
14–15		Reserved
16 – n		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* One of the following session keys is used: X′05′ PCID session key: generated by the SSCP(ILU) X′06′ Network Name Pair session key: OLU and DLU, respectively X′07′ Network Address Pair session key: PLU and SLU, respectively X′15′ Network-Qualified Address Pair session key: PLU and SLU, respectively
n+1 – n+2		Retired
n+3 – p		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′19′ Resource Identifier control vector (present if generic name resolution had occurred) X′1C′ Network-Qualified Name Pair control vector X′35′ Extended Sense Data control vector (present if the CDTERM RU represents a session initiation or termination failure) X′5C′ APPN Message Transport control vector X′5E′ Related Request control vector (present if the CDTERM is serving as a negative reply to an earlier request; this control vector identifies the request). X′60′ Fully Qualified PCID control vector (conditionally present)

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

CINIT (CONTROL INITIATE)

SSCP→PLU, Norm; FMD NS(s)

CINIT requests the PLU to attempt to activate, via a BIND request, a session with the specified SLU.

CINIT (CONTROL INITIATE)

Byte	Bit	Content
0–2		X′810601′ NS header
3		Format
	0–3	0000 Format 0 (only value defined) *Note:* CINIT format 0 may carry control vectors at the end of the basic RU.
	4–7	Reserved

CINIT (CONTROL INITIATE)

Byte	Bit	Content
4	0	INITIATE origin: 0 ILU is OLU 1 ILU is not OLU
	1	Substitution source (reserved if bits 4–5 do not = 11): 0 use the names contained in the X′0E′ control vectors; the Network Name (X′0E′) control vector for the PLU is not to be included in the BIND (reserved if bit 6 = 0) 1 use the names contained in the control vector X′16′; if bit 6 = 1, the Network Name (X′0E′) control vector for the PLU is to be included in the BIND
	2	0 SLU is OLU 1 PLU is OLU
	3	Retired
	4–5	Names substitution in BIND PLU and SLU name fields (bytes k+2 – m and p+2 – r): 00 no name substitution is to be performed by the receiver 01 no name substitution is to be performed by the receiver, but network identifiers are present and are to be removed from the BIND; if bit 6 = 1, the Network Name (X′0E′) control vector for the PLU is to be included in the BIND 10 no name substitution is to be performed by the receiver, but the Network Name (X′0E′) control vector for the PLU is to be included in the BIND (reserved if bit 6 = 0) 11 name substitution is to be performed by the receiver: the names from the source indicated by bit 1 are to be substituted into the PLU and SLU name fields in BIND. *Note:* Control vector X′0E′s are used if the names to be substituted are network-qualified real names; control vector X′16′ is used if the names to be substituted are not network-qualified real names.
	6	0 extended BIND is not sent to the SLU 1 extended BIND is sent to the SLU
	7	0 BIND with XRF Session Activation (X′27′) control vector is not supported for the SLU 1 BIND with XRF Session Activation (X′27′) control vector is supported for the SLU
5–9		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* The following session key is used: X′07′ Network Address Pair session key (retired): PLU and SLU, respectively *Note:* If control vector X′15′ is supported by the LU (or BF), then bytes 5–9 are reserved; otherwise, these bytes contain session key X′07′ when sent from the SSCP to a subarea LU or BF.
10–11		Length, in binary, of BIND Image field
12 – m		BIND image: bytes 1-p of the BIND RU, i.e., through the URC field (see BIND format description) *Note:* The URC Length field is included, even if it is set to 0.
m+1 – n		Name of SLU
m+1		Type: X′F3′ logical unit
m+2		Length, in binary, of symbolic name
m+3 – n		Symbolic name, in EBCDIC characters
n+1 – n+2		Retired
n+3 – r		User Field (from INITIATE RU)
n+3		Length, in binary, of user data *Note:* X′00′ = no user data is present.
n+4 – r		User data: user-specific data

CINIT (CONTROL INITIATE)

Byte	Bit	Content
n + 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		
n + 5 – r		Remainder of unstructured user data
For structured user data		
n + 5 – r		Structured subfields (For detailed definitions, see Chapter 8, “User Data Structured Subfields.”)
r + 1 – s		LU or Non-SNA Device Specifications
r + 1 – r + 2		Length, in binary, of characteristics field, including both format and characteristics fields—i.e., bytes r+3 – s *Note:* X′0000′ = no Format and no Characteristics fields are present.
r + 3 – s		Characteristics Field
r + 3		Characteristics format: X′01′ device characteristics (only value defined)
r + 4 – s		LU or Non-SNA Device Characteristics
Format X′01′ (This format represents an access-method-unique LU/device characteristics definition. For more specific information, refer to access method implementation documentation.)		
r + 4		Scheduling information: X′80′ input device X′40′ output device X′20′ conversational mode X′10′ reserved X′08′ start print sensitive X′04′ reserved X′02′ additional information provided (always *on*) X′01′ specific poll=*on*, general poll=*off*
r + 5		Device type: X′00′ undefined device type X′04′ 2741 X′08′ WTTY X′10′ 115A X′20′ TWX (33–35) X′30′ 83B3 X′40′ 2740 X′80′ 1050 X′90′ 2780 X′19′ 3277 X′1A′ 3284 X′1B′ 3286/3288 X′1C′ 3275 X′91′ 3780 X′6D′ SNA logical unit
r + 6		Model information: X′00′ Model 1 X′01′ Model 2

CINIT (CONTROL INITIATE)

Byte	Bit	Content
r+7		Feature information:
	0–1	00 SDLC 01 start/stop 10 BSC 11 reserved
	2–7	X′20′ XMIT interrupt feature X′10′ SWITCHED LINE = ON; LEASED LINE = OFF X′08′ attention X′04′ checking X′02′ station control X′01′ selector pen
r+8		Physical device address
r+9		Miscellaneous flags: X′80′ SNA compatible application program interface (always *on*) X′40′ non-SNA application program interface (always *off*) X′20′ buffered X′10′ continue mode X′08′ contention mode X′04′ inhibit mode (text timeout) X′02′ end-to-end control X′01′ 3270 extended data stream requiring BSC transparency
r+10		Device data stream compatibility characteristics: (This field is used in conjunction with the Device Type field, r+5, when that field is set to X′6D′: SNA logical unit; otherwise, it is reserved.): X′00′ no data stream characteristics defined here X′04′ 2741 X′08′ WTTY X′10′ 115A X′20′ TWX (33–35) X′30′ 83B3 X′40′ 2740 X′80′ 1050 X′90′ 2780 X′19′ 3277 X′1A′ 3284 X′1B′ 3286/3288 X′1C′ 3275 X′91′ 3780 X′A0′ – X′FF′ available for installation-defined use
r+11		Device language support:
	0	Query indicator: 0 Identify language characteristics of the device from the code specified in bits 1–7 1 Send query command to the device to determine the single byte character set language and double byte character set capability. (If the language cannot be determined from the input received from the query, the code specified in bits 1–7 will be used as default.)
	1–7	Language supported: 0000001 US English 0010001 Katakana
r+12 – r+16		Screen size (see the PS Usage field in the BIND RU for format)
r+17 – s		Work Area (This field is optional—if not present, s = r+16.):

CINIT (CONTROL INITIATE)

Byte	Bit	Content
r+17		Work area format: X′00′ unformatted X′01′ TCAM format
r+18 – s		Work area excluding format
s+1		Length of Session Cryptography Key field *Note:* X′00′ = no Session Cryptography Key field present.
s+2 – t		Session Cryptography Key field: session cryptography key enciphered under PLU master cryptography key
Note:		End of base RU
t+1 – u		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′0D′ Mode/Class-of-Service/Virtual-Route-Identifier List (always present) X′0E′ Network Name control vector: the network-qualified name of the PLU (always present) (Followed by another Network Name control vector containing the SLU name) X′0E′ Network Name control vector: the network-qualified name of the SLU (always present) (Preceded by another Network Name control vector containing the PLU name) X′15′ Network-Qualified Address Pair control vector: PLU and SLU, respectively (always present) X′16′ Names Substitution control vector: contains the names to be substituted into the PLU and SLU name fields in the BIND (present if the names to be substituted are not included in the X′0E′ control vectors) X′2B′ Route Selection control vector X′2C′ COS/TPF control vector: conditionally present X′2D′ Mode control vector: conditionally present (contains the mode name as known in the network of the SLU) X′2F′ LU Definition control vector (present if data is available to be sent and if Unrecognized-Control-Vectors-on-CINIT Support indicator was set in LU-LU Session Services Capabilities (X′0C′) control vector on RSP(ACTLU)) X′3F′ SSCP(SLU) Capabilities control vector: present if the SSCP(SLU) supports suppressing notifying the SLU of specific session initiation errors. X′59′ Installation-Defined CDINIT control vector (present when an SSCP exit is invoked) X′5F′ Extended Fully Qualified PCID control vector: conditionally present X′60′ Fully Qualified PCID control vector: present when the SLU supports extended BINDs X′64′ TCP/IP Information control vector (present to forward SLU TCP/IP information to the PLU if that information was provided by the SLU) X′66′ Length-Checked Compression control vector (present when compression is supported) X′68′ XRF/Session Cryptography control vector

CLEANUP (CLEAN UP SESSION)

SSCP→PLU|SLU, Norm; FMD NS(s)

CLEANUP is sent by the SSCP to an LU (in a subarea node or BF for peripheral LU) requesting that the LU or BF attempt to deactivate the session for the specified (PLU,SLU) network address pair.

CLEANUP (CLEAN UP SESSION)

Byte	Bit	Content
0– 2		X′810629′ NS header
3	0– 3	Format:
		0000 Format 0 (only value defined)
	4– 7	Reserved
4		Reserved
5		Reason:
	0	0 network user
		1 network manager
	1	0 normal
		1 abnormal
	2– 7	Reserved
6 – n		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields”
		Note: One of the following session keys is used:
		X′06′ uninterpreted name pair: PLU and SLU, respectively
		X′07′ network address pair (retired): PLU and SLU, respectively
		X′15′ network-qualified address pair: PLU and SLU, respectively
		Note: Only session keys X′07′ and X′15′ are defined for LU 6.2.
n+1 – m		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields”
		Note: The following control vectors may be included; they are parsed according to subfield parsing rule KL.
		X′35′ Extended Sense Data control vector

CLEAR (CLEAR)

PLU→SLU, Exp; SC

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

CLEAR (CLEAR)

Byte	Bit	Content
0		X′A1′ request code

CONNOUT (CONNECT OUT)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

CONNOUT requests the PU to initiate a connect-out procedure on the specified link.

CONNOUT (CONNECT OUT)

Byte	Bit	Content
0– 2		X′01020E′ NS header
3– 4		Element address of link, if ENA is supported; otherwise, its network address. For connection type = TYPE 1, bytes 3-4 contain the element address of the link station.
5		Link station identifier
6	0	Connection Type: 0 Type 0 CONNOUT for switched links. 1 Type 1 CONNOUT for non-switched links.
	1– 2	Connect-out feature: *Note:* In a Type 1 CONNOUT, the connect-out feature is always set to B′10′. 00 automatic connect out (dial digits are provided) 01 reserved 10 manual connect out (no dial digits are provided); this bit setting does not apply to CCITT X.21 connections 11 CCITT X.21 direct connect out (no dial digits are provided)
	3– 4	Reserved
	5	Connection support: 0 XID3s for connection establishment from this node should indicate that the node is LEN 1 XID3s for connection establishment from this node should indicate that the node is APPN.
	6	Networking Capabilities indicator: *Note:* This field is only defined when connection support (byte 6, bit 5) is APPN. This field is copied to the networking capabilities field of the XID(np) sent from this node. 0 the sender is not a network node 1 the sender is a network node
	7	Static/dynamic address indicator: 0 sender considers the adjacent link station (ALS) address to be static 1 sender considers the ALS address to be dynamic *Note:* Bytes 7 – n are not included on manual connect calls (bits 1–2 = 10).
7		Retry limit: number of times the connect-out procedure is to be retried
8		Number (m–8), in binary, of dial digits including ending and control characters, if any (0 for X.21 direct connect out)
9 – m		Dial digits: EBCDIC characters representing decimal digits and control information, including feature selection control codes, if any, as appropriate to the link connection

CONNOUT (CONNECT OUT)

Byte	Bit	Content
m+1 – n		ID number:
	0– 3	Reserved
	4– 2 3	A binary value that may uniquely identify a specific link station; the number can be assigned in various ways, depending on the product; see the individual product specifications for details. *Note:* Bytes m+1 – n are not included for peripheral links; they are present only for subarea links.
n+1 – p		One or more control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT. X′0E′ Network Name control vector: type X′F7′, local name of ALS at the XID sender (conditionally present) X′12′ Network ID control vector X′46′ TG Descriptor control vector (Control vector X′46′ is present when the connection is APPN level and the Connection support bit (byte 6, bit 5) is set in this RU.)

CONTACT (CONTACT)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

CONTACT requests the initiation of a procedure at the PU to activate DLC-level contact with the adjacent link station specified in the request. The DLC-level contact must be activated before any PIUs can be exchanged with the adjacent node over the link.

CONTACT (CONTACT)

Byte	Bit	Content
0– 2		X′010201′ NS header
3– 4		Element address of adjacent link station of the node to be contacted, if ENA is supported; otherwise, its network address

CONTACT (CONTACT)

Byte	Bit	Content
5	0	Retired
	1	Enhanced address management indicator: 0 sender does not support enhanced address management 1 sender supports enhanced address management
	2	Static/dynamic address indicator (reserved if byte 5, bit 1 = 0): 0 sender considers the adjacent link station (ALS) address to be static 1 sender considers the ALS address to be dynamic
	3	Limited resource indicator: 0 sender considers the link connection to the adjacent link station not to be a limited resource 1 sender considers the link connection to the adjacent link station to be a limited resource
	4	CP-CP session support indicator: *Note:* This field is defined only when connection support is APPN and it is copied to the networking capabilities field of the XID(np) sent from this node. 0 the XID(np) used in the Contact sequence should have both the ′CP-CP Sessions Supported′ and ′CP-CP Sessions Requested′ reset. 1 the XID(np) used in the Contact sequence should have both the ′CP-CP Sessions Supported′ and ′CP-CP Sessions Requested′ set.
	5	Connection support indicator: 0 the XID3s for connection establishment from this node should indicate that the node is LEN 1 the XID3s for connection establishment from this node should indicate that the node is APPN *Note:* This bit may not change between the value indicated on CONNOUT and the value indicated in CONTACT.
	6	Networking Capabilities indicator: *Note:* This field is defined only when connection support is APPN and it is copied to the networking capabilities field of tha XID(np) sent from this node. 0 the sender is not a network node 1 the sender is a network node
	7	Nonnative network ID usage indicator: (reserved when nonnative network LU attachment is not supported by both the SSCP and PU partners of this session) 0 use the adjacent node′s nonnative network ID as the network ID for all LUs on the connection. 1 use the native network ID as the network ID for all LUs on the connection, despite the fact that the adjacent node′s network ID is not the same as the native network ID.
6		Transmission group number *Note:* Used by the receiving PU T4|5 in the XID format 3 sent to the adjacent node; for SSCPs that do not support route extension transmission groups, set to 0. *Note:* Used by the receiving PU T4|5 for subarea applications

CONTACT (CONTACT)

Byte	Bit	Content
7 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′0E′ Network Name control vector: contains the adjacent link station name, type X′F4′ (included if byte 5, bit 1 is set to 1) X′46′ TG Descriptor control vector with TG Identifier (X′80′) subfield (present when the connection is APPN level and the APPN Networking Functions Support bit [byte 5, bit 4] in the SSCP-PU Session Capabilities [X′0B′] control vector is set in the ACTPU and RSP(ACTPU)) X′61′ HPR Capabilities control vector (present when the sender and receiver support HPR and the sender requests the receiver to present an HPR-capable appearance to the specified adjacent node)

CONTACTED (CONTACTED)

PU T4|5→SSCP, PU→PUCP, Norm; FMD NS(c)

CONTACTED is issued by the PU to indicate to the SSCP the completion of the DLC contact procedure. A status parameter conveyed by this request informs SSCP configuration services whether or not the contact procedure was successful; if not successful, the status indicates whether an adjacent node load is required or whether an error occurred on the contact procedure.

CONTACTED (CONTACTED)

Byte	Bit	Content
0– 2		X′010280′ NS header
3– 4		Element address of adjacent link station in the node being contacted, if ENA is supported; otherwise, its network address
5		Status of adjacent link station or node associated with adjacent link station: X′01′ loaded (no field follows) X′02′ load required (no field follows) X′03′ error on CONTACT (no field follows) X′04′ loaded (additional field, bytes 6 – p, follows) X′05′ exchanged parameters in XID Format 2 I-field not compatible (additional field, bytes 6 – p, follows) X′07′ no routing capability to adjacent node (additional field, bytes 6 – p, follows) X′08′ incompatible parameters in XID Format 2 I-field for addition of link station to currently active TG (additional field, bytes 6 – p, follows) X′09′ loaded, in another subnetwork (bytes 19–26 are added after byte 18 of X′04′ status) X′0A′ contacted node, XID fields present X′0B′ contact error for node, XID fields present X′0C′ set aside for internal implementation-specific use, and will not be otherwise defined in SNA. *Note:* Status bytes X′0A′ and X′0B′ are used when both the SSCP and the sending PU support attachment of T2.1 nodes. Format 3 XID fields are present when the CONTACTED node is a T2.1.

CONTACTED (CONTACTED)

Byte	Bit	Content
Note:		End of RU for status bytes X′01′, X′02′, and X′03′; RU continues for status bytes X′04′, X′05′, X′07′, X′08′ X′09′, X′0A′, and X′0B′.
6 – p		Additional fields for status bytes X′04′, X′05′, X′07′, X′08′, X′09′, X′0A′, and X′0B′.
For status bytes X′04′ and X′09′		
6		Resolved TG number
7–10		Adjacent node subarea address (right-justified with leading 0s)
11–18 (=p)		IPL load module ID received from the adjacent node: an eight-character EBCDIC symbolic name of the IPL load module currently operating in the adjacent node *Note:* X′40...40′ = no information conveyed.
p+1 – q		One or more control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT. X′57′ DLC connection data: may be present for status byte X′04′ if DLC type is appropriate
Note:		End of RU for status byte X′04′; RU continues for status byte X′05′, X′07′, X′08′, X′09′, X′0A′, X′0B′
19–26 (=p)		Network ID of the subnetwork that contains the contacted station
p+1 – q		One or more control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT. X′57′ DLC connection data: may be present for status byte X′09′ if DLC type is appropriate
Note:		End of RU for status byte X′09′
For status bytes X′05′, X′07′, X′08′, X′0A′, and X′0B′		
6		Length, in binary, of XID Format 2 or 3 I-field received
7 – n		XID Format 2 or 3 I-field received (See Chapter 3, “Exchange Identification (XID) Information Fields” for format details.)
n+1		Length, in binary, of XID Format 2 or 3 I-field sent
n+2 – p		XID Format 2 or 3 I-field sent (See Chapter 3, “Exchange Identification (XID) Information Fields” for format details.)
Note:		End of RU for status bytes X′05′, X′07′, and X′08′; RU continues for status byte X′0A′ and X′0B′

CONTACTED (CONTACTED)

Byte	Bit	Content
p + 1	0	DLC activation sequence: 0 DLC activation sequence not executed 1 DLC activation sequence executed (e.g., for SDLC links, SNRM/SNRME or SABME sent)
	1	SSCP takeover of independent LUs: 0 no BFSESSINFO RUs follow 1 BFSESSINFO RUs follow: a takeover has occurred and the T4 BF will report all sessions with independent LUs associated with the contacted adjacent link station
	2	0 contacted node is not a T2.1 node 1 contacted node is a T2.1 node
	3	0 CONTACTED is solicited 1 CONTACTED is unsolicited
	4	0 DLC XID exchange executed 1 DLC XID exchange not executed
	5	0 No mismatch exists between the CP name and/or TGN requested by CONTACT and those in use on the connection. 1 A mismatch exists between the CP name and/or TGN requested by CONTACT and those in use on the connection; the SSCP should treat the connection as a LEN connection. The condition persists until the TG is reactivated or the adjacent node regains CP name change support.
	6	Nonnative network ID usage indicator: (reserved when nonnative network LU attachment is not supported by both the SSCP and PU partners of this session) 0 The adjacent node′s nonnative network ID will be used as the network ID for all LUs on the connection. 1 The native network ID will be used as the network ID for all LUs on the connection, despite the fact that the adjacent node′s network ID is not the same as the native network ID.
	7	Reserved
p + 2 – q		One or more control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT. X′57′ DLC connection data: may be present for status byte X′0A′ if DLC type is appropriate
Note:		End of RU for status byte X′0A′ and X′0B′

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

CRV (CRYPTOGRAPHY VERIFICATION)

Byte	Bit	Content
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.

CTERM (CONTROL TERMINATE)

SSCP→PLU, Norm; FMD NS(s)

CTERM requests that the PLU attempt to deactivate a session identified by the specified (PLU,SLU) network address pair.

CTERM (CONTROL TERMINATE)

Byte	Bit	Content
0– 2		X′810602′ NS header
3	0– 3	0000 Format 0 (only value defined)
	4– 7	Reserved
4		Type:
	0– 1	Reserved
	2– 3	00 reserved 01 orderly 10 forced 11 cleanup
	4– 7	Reserved
5		Reason:
	0	0 network user 1 network manager
	1	0 normal 1 abnormal
	2– 7	Reserved
6– 7		Reserved
8 – m		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* One of the following session keys is used: X′07′ Network Address Pair session key (retired): PLU and SLU, respectively X′15′ Network-Qualified Address Pair session key: PLU and SLU, respectively
m + 1 – m + 2		Retired

DACTCDRM (DEACTIVATE CROSS-DOMAIN RESOURCE MANAGER)

SSCP→SSCP, Exp; SC

DACTCDRM is sent to deactivate an SSCP-SSCP session.

DACTCDRM (DEACTIVATE CROSS-DOMAIN RESOURCE MANAGER)

Byte	Bit	Content
0		X′15′ request code
1	0–3	Format: X′0′ (only value defined)
	4–7	Type deactivation requested: X′1′ normal end of session X′2′ invalid activation parameter, sent by the primary half-session to deactivate the session and to indicate to the secondary that the response to ACTCDRM contained an invalid parameter X′3′ session-outage notification (SON) X′4′ normal deactivation of SSCP-SSCP session; active LU-LU sessions that used this SSCP-SSCP session for session establishment should not be terminated
End of Type 1; Type 2 Continues		
2–5		Reason code (included only if type deactivation requested is invalid activation parameter, i.e., byte 1, bits 4–7 = X′2′): sense data (see Chapter 10, “Sense Data” on page 10-1) corresponding to the error
End of Type 2; Type 3 Continues		
2		Cause of session outage notification: X′07′ virtual route inoperative: the virtual route being used by the SSCP-SSCP session has become inoperative, thus forcing the deactivation of the SSCP-SSCP session X′0B′ virtual route deactivated: the identified SSCP-SSCP session is being deactivated because of a forced deactivation of the virtual route being used by the session X′0C′ SSCP failure—unrecoverable: the identified SSCP-SSCP session had to be deactivated because of an abnormal termination of one of the SSCPs of the session; recovery from the failure was not possible X′0D′ session override: the subject session has to be deactivated because of a more recent session activation request for the same session over a different virtual route X′0E′ SSCP failure—recoverable: the identified SSCP-SSCP session had to be deactivated because of an abnormal termination of one of the SSCPs of the session; recovery from the failure may be possible X′0F′ cleanup: the SSCP is resetting its half-session before it receives the response from the partner SSCP receiving the DACTCDRM X′10′ SSCP contention: two SSCPs have sent each other an ACTCDRM request over different virtual routes; the SSCP receiving the ACTCDRM from the SSCP with the greater SSCP ID sends DACTCDRM, with this SON code, to the other SSCP over the same virtual route on which the contention-losing ACTCDRM was sent X′11′ gateway node cleanup: a gateway node is cleaning up the session because the gateway SSCP session partner has forced deactivation of the session (via NOTIFY) *Note:* In this case, the receiving SSCP does not send NOTIFY to the gateway node.
3		Reserved
End of Type 3; Type 4 Continues		

DACTCDRM (DEACTIVATE CROSS-DOMAIN RESOURCE MANAGER)

Byte	Bit	Content
2		Reserved

DACTCONNIN (DEACTIVATE CONNECT IN)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

DACTCONNIN requests the PU to disable the specified link from accepting incoming calls.

DACTCONNIN (DEACTIVATE CONNECT IN)

Byte	Bit	Content
0–2		X′010217′ NS header
3–4		Element address of link, if ENA is supported; otherwise, its network address

DACTLINK (DEACTIVATE LINK)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

DACTLINK initiates a procedure at the PU to deactivate the protocol boundary between a link station in the node (as specified by the link network address parameter in the request) and the link connection attached to it. The normal type is used after all adjacent link stations on the specified link have been discontacted. The unconditional type may be used at any time to reset immediately a link and its attached stations, regardless of their current state. If any other control points were actively sharing control of the link at the time of reset, they are notified via INOP with a unique code. The give-back type gives back link ownership without disrupting LU-LU sessions.

DACTLINK (DEACTIVATE LINK)

Byte	Bit	Content
0–2		X′01020B′ NS header
3–4		Element address of link, if ENA is supported; otherwise, its network address
5		DACTLINK type: X′00′ normal X′01′ unconditional reset X′02′ give-back

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)
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|PUCP→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)

DACTTRACE (DEACTIVATE TRACE)

SSCP→PU T4|5, Norm; FMD NS(ma)

DACTTRACE requests the PU to deactivate a specified type of trace for a specified resource (or hierarchy of resources for a generalized PIU trace).

DACTTRACE (DEACTIVATE TRACE)

Byte	Bit	Content
0–2		X′010303′ NS header
3–4		Element address of the resource associated with the trace, if ENA is supported; otherwise, its network address *Note:* For generalized PIU trace (byte 5, bit 1 set to 1), bytes 3–4 contain the address of the PU receiving DACTTRACE; the address of the specific resource identifying the resource hierarchy for trace deactivation is contained in bytes 7–8.

DACTTRACE (DEACTIVATE TRACE)

Byte	Bit	Content
5		Selected trace (a bit is set to 1 if the indicated trace option is selected):
	0	TG Trace
	1	Generalized PIU trace *Note:* When this bit is set to 1, all other bits in this byte are set to 0.
	2–3	Reserved
	4	Scanner internal trace
	5	Reserved
	6	Trace all frames
	7	Link trace, transmission group trace, or both (depending on the active traces)
6–8		Specific trace data
For byte 5, bit 1 = 0		
6–7		Reserved
8		Reserved or omitted
For byte 5, bit 1 = 1		
6		Generalized PIU trace flags:
	0–6	Reserved
	7	0 deactivate trace for all resources within the resource hierarchy of the PU receiving DACTTRACE (bytes 7–8 are reserved) 1 deactivate trace for resources within the resource hierarchy of the resource specified in bytes 7–8
7–8		Element address of a specific resource identifying a resource hierarchy for trace deactivation, if ENA is supported; otherwise, its network address (if byte 6, bit 7 = 1; otherwise, reserved)

DELETENR (DELETE NETWORK RESOURCE)

SSCP→PU T4|5, Norm; FMD NS(c)

DELETENR is sent to free a network address assigned to a link or adjacent link station.

DELETENR (DELETE NETWORK RESOURCE)

Byte	Bit	Content
0–2		X′41021C′ NS header
3–4		Element address of resource being deleted, if ENA is supported; otherwise, its network address

DELIVER (DELIVER)

SSCP→LU, Norm; FMD NS(ma)

(Retired RU) DELIVER has been retired from SNA. Consult product documentation for further information and support.

DISCONTACT (DISCONTACT)

SSCP→PU T4|5, PUCP→PU, Norm; FMD NS(c)

DISCONTACT requests the PU to deactivate DLC-level contact with the specified adjacent node. The discontact procedure is DLC-dependent; if applicable, polling is stopped. DISCONTACT may be used to terminate contact, IPL, or dump procedures before their completion. The PU responds negatively to DISCONTACT if an uninterruptible link-level procedure is in progress at the primary link station of the specified link.

DISCONTACT (DISCONTACT)

Byte	Bit	Content
0–2		X′010202′ NS header
3–4		Element address of adjacent link station to be discontacted, if ENA is supported; otherwise, its network address

DISPSTOR (DISPLAY STORAGE)

SSCP→PU T4, Norm; FMD NS(ma)

DISPSTOR requests the PU to send a RECSTOR RU containing a specified number of bytes of storage beginning at a specified location, or the names and related information for load modules and dump (if present) on the disk attached to the T4 node.

DISPSTOR (DISPLAY STORAGE)

Byte	Bit	Content
0–2		X′010331′ NS header
3–4		Element address of resource to be displayed, if ENA is supported; otherwise, its network address

DISPSTOR (DISPLAY STORAGE)

Byte	Bit	Content
5		Display target and type:
	0– 3	Target address space to be displayed *Note:* Refer to implementation documentation for a description of these values.
	4– 7	Display type: 0001 nonstatic storage display 0010 static snapshot display 0100 purge dump 1000 names and related information for load modules and dump (if present) on the disk attached to the T4 node
6		Reserved
7– 8		Number of bytes to be displayed *Note:* This field is ignored if the display type is purge dump.
9– 1 2		Beginning location of display

DSRLST (DIRECT SEARCH LIST)

SSCP→SSCP, Norm; FMD NS(s)

DSRLST specifies a list search argument to be used at the receiving SSCP to identify a control list type to be returned on RSP(DSRLST).

DSRLST (DIRECT SEARCH LIST)

Byte	Bit	Content
0– 2		X′818627′ NS header
3		Search argument type: X′01′ Retired X′02′ Resource Identifier control vector identifying an LU for which an LU Status control list (type X′01′) is to be returned *Note:* DSRLST with search argument X′02′ replaces DSRLST with search argument X′01′. X′03′ Retired
4 – m		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors are included; they are parsed according to subfield parsing rule KL. X′19′ Resource Identifier control vector (always present): Identifies the LU as known to the originating SSCP. The SSCP name, if known, is the name of the SSCP in whose domain the target LU is defined, *not* the name of the SSCP on the target side of this gateway X′2C′ COS/TPF control vector X′2D′ Mode control vector (conditionally present, contains the mode name as known in the network of the SLU) X′3E′ Directory Entry Characteristic control vector X′5A′ Session services extensions support control vector X′5B′ Interchange Node Parameters control vector

DSRLST (DIRECT SEARCH LIST)

Byte	Bit	Content
		X′5C′ APPN Message Transport control vector
		X′60′ Fully Qualified PCID control vector (always present)
		X′63′ Cryptography Key Distribution control vector (present only when key distribution information is to be transferred)

DUMPFINAL (DUMP FINAL)

SSCP→PU T4|5, Norm; FMD NS(c)

DUMPFINAL performs one of two functions depending on the address used in the request:

- When the DUMPFINAL request contains the link station address of an adjacent T4 node, DUMPFINAL terminates the dump sequence in progress (whether DUMPTEXT is used or not). A positive response by the T4|5 node to this form of DUMPFINAL indicates that the dump sequence is complete.
- When the DUMPFINAL request contains the network address of the receiving T4 node (not applicable to a T5 node) and a link station address of X′0000′, the DUMPFINAL request causes an ABEND at the T4 node. The T4 node then dumps to local disk. No response is returned to the requester for this form of DUMPFINAL.

DUMPFINAL (DUMP FINAL)

Byte	Bit	Content
0– 2		X′010208′ NS header
3– 4		One of the following addresses: • Element address of adjacent link station of the node to be dumped, if ENA is supported; otherwise, its network address. • Element address (X′0000′) of the receiving PU when the request is to force a dump of the T4 node to local disk.

DUMPINIT (DUMP INITIAL)

SSCP→PU T4|5, Norm; FMD NS(c)

DUMPINIT performs one of two functions, depending on the address used in the request:

- When the node to be dumped is identified by an adjacent link station address, DUMPINIT causes the receiving T4|5 node to initiate a DLC-level dump from the adjacent T4 node (identified in the DUMPINIT) to the receiving T4|5 node; this dump is sent to the SSCP on subsequent RSP(DUMPTEXT)s.
- When the DUMPINIT request contains the network address of the receiving T4 node (not applicable to a T5 node), a link station address of X′0000′, and a Dump Control byte equal to X′80′, the DUMPINIT interrogates the status of the receiving node′s system-defined local options (to react to a subsequent DUMPFINAL), and its capacity to store a dump of its own contents to local disk storage. A positive response to the request indicates that a DUMPFINAL request can be accepted (and the local dump be performed). A negative response indicates either that the system defined local options conflict with those of the requester, or insufficient disk capacity exists to at the DUMPINIT receiver hold the dump.

DUMPINIT (DUMP INITIAL)

Byte	Bit	Content
0–2		X′010206′ NS header
3–4		One of the following addresses: • Element address of adjacent link station of the node to be dumped, if ENA is supported; otherwise, its network address. • Element address (X′0000′) of the receiving PU when the request is to initiate the sequence to force a dump of the local disk
5		Dump type: bit 0, 1 Force dump to local disk storage (only value defined) bits 1–7 reserved

DUMPTEXT (DUMP TEXT)

SSCP→PU T4|5, Norm; FMD NS(c)

If further dump data is required, DUMPINIT may be followed by DUMPTEXT. DUMPTEXT causes the dump data specified by the starting-address parameter to be returned to the SSCP on the response. The T4|5 obtains the dump data from the T4 node, using a DLC-level interchange.

DUMPTEXT (DUMP TEXT)

Byte	Bit	Content
0–2		X′010207′ NS header
3–4		Element address of adjacent link station of the node to be dumped, if ENA is supported; otherwise, its network address
5–8		Starting address where dump data is to begin
9–10		Length of text: 2-byte binary count of the number of bytes of dump data to be returned

ECHOTEST (ECHO TEST)

SSCP→LU, Norm; FMD NS(ma)

(Retired RU) ECHOTEST has been retired from SNA. Consult product documentation for further information and support.

ER-INOP (EXPLICIT ROUTE INOPERATIVE)

PU T4|5→SSCP, PU T4→PUCP, Norm; FMD NS(c)

(Retired RU) ER-INOP has been retired from SNA. Consult product documentation for further information and support.

ER-TESTED (EXPLICIT ROUTE TESTED)

PU T4|5→SSCP, Norm; FMD NS(ma)

ER-TESTED is sent by a subarea node to one or more SSCPs to provide the status of an ER as determined by explicit route test procedures.

ER-TESTED (EXPLICIT ROUTE TESTED)

Byte	Bit	Content
0–2		X′410386′ NS header
3		Format: X′01′ Format 1 X′02′ Format 2; same as Format 1, except that it includes bytes 48 – n
4		Type: X′00′ the corresponding NC-ER-TEST reached its destination subarea X′02′ ER not reversible since there is no reverse ERN defined X′03′ Encountered a node that does not support the ER X′04′ ER length exceeded that specified in the NC-ER-TEST request X′05′ ER requires a TG that is not active X′06′ ER is not defined in the NC-ER-TEST-REPLY originating node X′07′ Retired

ER-TESTED (EXPLICIT ROUTE TESTED)

Byte	Bit	Content
5		Explicit route length, in terms of the number of transmission groups in the explicit route, as accumulated in NC-ER-TEST
6		Maximum ER length, as specified in the NC-ER-TEST request
7–10		Subarea address of the destination PU of the corresponding NC-ER-TEST
11		Reserved
12	0–3 4–7	Reserved ERN of the ER tested
13–16		Subarea address of the originating PU of the corresponding NC-ER-TEST
17–18		Reverse ERN mask: A bit is *on* if the corresponding ERN can be used to route from the NC-ER-TEST-REPLY originating subarea to the NC-ER-TEST originating subarea (Bit 0 corresponds to ERN 0, bit 1 to ERN 1, and so forth.)
19–20		Maximum PIU length allowed on the reverse ERN specified in bytes 17–18: X′00′ no restriction (only value defined)
21–22		Maximum PIU size accumulated by the corresponding NC-ER-TEST: X′00′ no restriction (only value defined)
23–26		NS_ER_TEST origin SSCP subarea number
27–28		NS_ER_TEST origin SSCP element number
29–38		Request Correlation field, as specified in the corresponding ROUTE-TEST
39–42		Subarea address of the PU that originated the corresponding NC-ER-TEST-REPLY
43–46		Subarea address depending on the Type field (byte 4) as follows: **Type** **Contents of this field** X′00′ reserved X′02′ subarea on the ER prior to that with no reverse ERN defined X′03′ subarea that does not support the ER X′04′ subarea on the ER preceding the subarea where the explicit route length (byte 5 of NC-ER-TEST) is incremented to a value one more than the maximum ER length limit (byte 6) X′05′ subarea on the other end of the TG that is not active X′06′ subarea on the ER from which the PU (that does not have the ER defined) received the corresponding NC-ER-TEST X′07′ reserved
47		TGN of the TG between the subareas specified in bytes 39–42 and 43–46; reserved if Type is X′00′

End of Format 1; Format 2 continues below

48–51		Subarea address of the adjacent node through which the tested explicit route flows from the node receiving ROUTE-TEST *Note:* Bytes 48–51 are reserved if this request is built by nodes other than the original receiver of ROUTE-TEST.
52		Transmission group number of the TG (to the node identified in bytes 48–51) over which the tested explicit route flows from the node receiving ROUTE-TEST *Note:* Byte 52 is reserved if this request is built by nodes other than the original receiver of ROUTE-TEST
53–60		Network ID of subnetwork containing the ER *Note:* This network ID defines the subnetwork in which the above addresses are valid.

ER-TESTED (EXPLICIT ROUTE TESTED)

Byte	Bit	Content
61–62		Bit mask of VRs that use the ER specified by bytes 7–16 (bit n corresponds to VRN n) *Note:* Bytes 61–62 are reserved if this request is built by nodes other than the original receiver of ROUTE-TEST.
63 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included in ER-TESTED format 2; they are parsed according to subfield parsing rule KL. X'1F' ER Configuration control vector (One X'1F' control vector may be included for each node along the route that supports this vector in NC-ER-TEST-REPLY.)

ESLOW (ENTERING SLOWDOWN)

PU T4→SSCP, Norm; FMD NS(c)

ESLOW informs the SSCP that the node of the sending PU has entered a slowdown state. This state is generally associated with buffer depletion, and requires traffic through the node to be selectively reduced or suspended.

ESLOW (ENTERING SLOWDOWN)

Byte	Bit	Content
0–2		X'010214' NS header
3–4		Element address of PU, if ENA is supported; otherwise, its network address

EXECTEST (EXECUTE TEST)

SSCP→PU T4|5, Norm; FMD NS(ma)

EXECTEST requests the PU to activate the specified test type related to the specified network address. The test code specifies the test type and defines the contents of the test data field. The test may be for the PU, or for the LUs or links supported by the PU.

EXECTEST (EXECUTE TEST)

Byte	Bit	Content
0–2		X'010301' NS header
3–4		Element address of resource to be tested, if ENA is supported; otherwise, its network address
5–8		Binary code selecting the test
9 – n		Data to support the selected test

EXPD (EXPEDITED DATA)

LU→LU, Exp; DFC

EXPD is an expedited-flow request that can be sent between half-sessions, regardless of the status of the normal flows, to carry TP-defined data. This RU is defined for LU 6.2 only.

EXPD (EXPEDITED DATA)

Byte	Bit	Content
0		X′03′ request code
1		Reserved
2– 3		Length, in binary, of Expedited Data field
		Note: The value of this field includes the Length field itself. The minimum value of this field is 3.
4 – m(≤ 89)		Expedited data: one or more TP-defined bytes

EXSLOW (EXITING SLOWDOWN)

PU T4→SSCP, Norm; FMD NS(c)

EXSLOW informs the SSCP that the node of the sending PU is no longer in the slowdown state and regular traffic can resume.

EXSLOW (EXITING SLOWDOWN)

Byte	Bit	Content
0– 2		X′010215′ NS header
3– 4		Element address of PU, if ENA is supported; otherwise, its network address

FNA (FREE NETWORK ADDRESSES)

SSCP→PU T4|5, Norm; FMD NS(c)

FNA is sent from an SSCP to request the PU T4|5 to free the identified element address(es) associated with the target resource. If ENA is not supported, the entire network address is in each Element Address field throughout this RU.

FNA (FREE NETWORK ADDRESSES)

Byte	Bit	Content
0– 2		X′01021A′ NS header
3– 4		Element address of target link, adjacent link station (ALS), LU, or subarea PU
5		Number of element addresses (of the type indicated in the note below) to be freed (X′00′ = all—and bytes 7 – n not present)
6	0	Retired, set to 1
	1	Enhanced address management indicator: 0 sender does not support enhanced address management 1 sender supports enhanced address management *Note:* When bit 1 = 1, this FNA may free only a single address.
	2	Static/dynamic address indicator (reserved if byte 6, bit 1 = 0): 0 sender considers the address to be static 1 sender considers the address to be dynamic
	3– 7	Reserved
7– 8		First element address to be freed
9 – n		Any additional element addresses (two-byte multiples) *Note:* All the element addresses specified in bytes 7 – n are associated with the same target link, ALS, LU or subarea PU. See the following table for the relation of target resources to resources to free.

Target resource	Resources to be freed
Link	adjacent link station address(es) associated with the target link
ALS	LU address(es) associated with the target adjacent link station (see Note)
LU	primary LU address(es) used for parallel sessions by the subarea LU identified in the target address field; the LU address in the target address field is the one used for the SSCP-LU session (see Note)
Subarea PU	subarea LU address associated with the SSCP-LU session (see Note)

Note: For dependent and independent LUs, all LU addresses to be freed are identified by the associated target ALS; for subarea LUs, PLU addresses are identified by the associated target SLU (SSCP-LU session) address and SLU addresses are identified by the associated subarea PU address.

FORWARD (FORWARD)

LU→SSCP, Norm; FMD NS(ma)

(Retired RU) FORWARD has been retired from SNA. Consult product documentation for further information and support.

INIT-OTHER (INITIATE-OTHER)

ILU→SSCP, Norm; FMD NS(s)

INIT-OTHER from the ILU requests the initiation of a session between the two LUs named in the RU. The requester may be a third-party LU or one of the two named LUs. This RU is not used by LU 6.2, although it can be used by a third-party LU for LU 6.2.

INIT-OTHER (INITIATE-OTHER)

Byte	Bit	Content
0–2		X′810680′ NS header
3	0–3	Format: 0001 Format 1 0010 Format 2
	4–7	Reserved
4		Type:
	0–1	00 retired 01 initiate only (I): do not enqueue 10 retired 11 initiate/enqueue (I/Q): enqueue the request if it cannot be satisfied immediately (See bytes 5–6 for further specification of queuing conditions.)
	2–3	Retired
	4–5	Reserved
	6	PLU/SLU specification: 0 LU1 is PLU. 1 LU2 is PLU.
	7	Reserved
5	0–1	Queuing Conditions for LU1
	0	0 Do not enqueue if session limit will be exceeded. 1 Enqueue if session limit will be exceeded.
	1	0 Do not enqueue if the LU is not currently able to comply with the PLU/SLU specification (as given in byte 4, bits 5–6). 1 Enqueue even though the LU might not be currently able to comply with the PLU/SLU specification.
	2	Reserved
	3	Retired
	4	Reserved
	5–6	Queuing position/service: 00 retired 01 Enqueue this request FIFO; i.e., the request will be dequeued after the requests already in the queue. 10 Enqueue this request LIFO; i.e., the request will be dequeued before the requests already in the queue. 11 reserved
	7	Reserved *Note:* Enqueueing is not performed if the DLU is unknown, or if the domain of either LU is in takedown status.

INIT-OTHER (INITIATE-OTHER)

Byte	Bit	Content
6	0–1	Queuing Conditions for LU2
	0	0 Do not enqueue if session limit will be exceeded. 1 Enqueue if session limit will be exceeded.
	1	0 Do not enqueue if the LU is not currently able to comply with the PLU/SLU specification (as given in byte 4, bits 5–6). 1 Enqueue even though the LU might not be currently able to comply with the PLU/SLU specification. *Note:* Enqueuing is not performed if the DLU is unknown, or if the domain of either LU is in takedown status.
	2	Reserved
	3	Retired
	4	Reserved
	5–6	Queuing position/service: 00 retired 01 Enqueue this request FIFO; i.e., the request will be dequeued after the requests already in the queue. 10 Enqueue this request LIFO; i.e., the request will be dequeued before the requests already in the queue. 11 reserved
	7	Reserved
7	0	Initiate type: 0 asynchronous initiate 1 synchronous initiate
	1	Retired
	2	Reserved
	3	Retired
	4–6	Reserved
	7	0 backup session not requested 1 backup session requested
8		NOTIFY specifications:
	0–3	NOTIFY(Resource Requested) conditions:
	0–1	00 Do not send NOTIFY to LUs in session with LU1. 01 reserved 10 Send NOTIFY to all LUs in session with LU1 only if the request is queued. 11 reserved
	2–3	00 Do not send NOTIFY to LUs in session with LU2. 01 reserved 10 Send NOTIFY to all LUs in session with LU2 only if the request is enqueued. 11 reserved
	4	Reserved
	5	NOTIFY(X′03′) conditions: 0 Do not send NOTIFY to the ILU when the requested session is set up. 1 Send NOTIFY to the ILU when the requested session is set up.
	6	Reserved
	7	Request for notification of resource availability, i.e., if a resource required for setup of the requested session is temporarily unavailable and subsequently becomes available, NOTIFY(Resource Available) is requested to notify the initiator of the resource′s availability: 0 Do not send NOTIFY to the initiator. 1 Send NOTIFY to the initiator.

INIT-OTHER (INITIATE-OTHER)

Byte	Bit	Content
9–16		Mode name: an 8-character type-1130 symbol string (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
17 – m		Uninterpreted Name of LU1
17		Type: X′F3′ logical unit
18		Length, in binary, of LU1 name
19 – m		EBCDIC character string
m + 1 – n		Uninterpreted Name of LU2
m + 1		Type: X′F3′ logical unit
m + 2		Length, in binary, of LU2 name
m + 3 – n		EBCDIC character string
n + 1 – n + 2		Retired
n + 3 – r		User Field
n + 3		Length, in binary, of user data *Note:* X′00′ = no user data is present.
n + 4 – r		User Data
n + 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		
n + 5 – r		Remainder of unstructured user data
For structured user data		
n + 5 – r		Structured subfields (For detailed definitions, see Chapter 8, “User Data Structured Subfields.”)
r + 1 – s		User Request Correlation (URC) Field
r + 1		Length, in binary, of URC *Note:* X′00′ = no URC.
r + 2 – s		URC: LU-defined identifier; this value can be returned by the SSCP in a subsequent NOTIFY to correlate a given session to the initiating request
Format 1 Only		

INIT-OTHER (INITIATE-OTHER)

Byte	Bit	Content
s+1 – t		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors are used; they are parsed according to subfield parsing rule KL: X′0E′ Network Name control vector (present if needed to specify the network-qualified PLU name) X′0E′ Network Name control vector (present if needed to specify the network-qualified SLU name) X′34′ LU Definition Override control vector (present if model terminal support override values are available) X′5F′ Extended Fully Qualified PCID control vector (conditionally present) X′69′ Switched Parameters control vector
Format 2 Only		
s+1 – s+8		COS name: type-1130 symbol string identifying the class of service (A value of eight space [X′40′] characters may be specified; in this case, the COS name is derived from the mode name table, using the mode name received in bytes 9–16.)
s+9 – t		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL: X′34′ LU Definition Override control vector (present if model terminal support override values are available) X′5F′ Extended Fully Qualified PCID control vector (conditionally present)

INIT-OTHER-CD (INITIATE-OTHER CROSS-DOMAIN)

SSCP→SSCP, Norm; FMD NS(s)

INIT-OTHER-CD from the SSCP(ILU) requests that a session be initiated between the two LUs named in the RU. The INIT-OTHER-CD request simply transports an INIT-OTHER from the SSCP(ILU—a third-party SSCP in this case to the SSCP(OLU).

INIT-OTHER-CD (INITIATE-OTHER CROSS-DOMAIN)

Byte	Bit	Content
0–2		X′818640′ NS header
3	0–3	Format: 0000 Format 0 0010 Format 2 (retired): specifies COS Name field in addition to the parameters in Format 0 0011 Format 3 (replaces Format 2): includes a Resource Identifier control vector for rerouting
	4–7	Reserved
4	0–1	Type: 00 retired 01 initiate only (I): do not enqueue

INIT-OTHER-CD (INITIATE-OTHER CROSS-DOMAIN)

Byte	Bit	Content
		10 retired 11 initiate/enqueue (I/Q): enqueue the request if it cannot be satisfied immediately (See bytes 5–6 for further specification of queuing conditions.)
	2–3	Retired
	4–5	Reserved
	6	PLU/SLU specification: 0 LU1 is PLU 1 LU2 is PLU
	7	Reserved
5		Queuing conditions for LU1:
	0	0 do not enqueue if session limit will be exceeded 1 enqueue if session limit will be exceeded
	1	0 do not enqueue if the LU is not currently able to comply with the PLU/SLU specification (as given in byte 4, bit 6) 1 enqueue if the LU is not currently able to comply with the PLU/SLU specification
	2	Reserved
	3	Retired
	4	Reserved
	5–6	00 retired 01 enqueue this request FIFO; i.e., the request will be dequeued after the requests already in the queue. 10 enqueue this request LIFO; i.e., the request will be dequeued before the requests already in the queue. 11 reserved
	7	Reserved *Note:* Enqueuing is not performed if the DLU is unknown, or if the domain of either LU is in takedown status.
6		Queuing conditions for LU2:
	0	0 do not enqueue if session limit will be exceeded 1 enqueue if session limit will be exceeded
	1	0 do not enqueue if the LU is not currently able to comply with the PLU/SLU specification (as given in byte 4, bit 6) 1 enqueue even though the LU might not be currently able to comply with the PLU/SLU specification
	2	Reserved
	3	Retired
	4	Reserved
	5–6	Queuing position/service: 00 retired 01 enqueue this request FIFO, i.e., the request will be dequeued after the requests already in the queue. 10 enqueue this request LIFO, i.e., the request will be dequeued before the requests already in the queue. 11 reserved
	7	Reserved *Note:* Enqueuing is not performed if the DLU is unknown, or if the domain of either LU is in takedown status.
7–14		PCID: a unique value used as a session identifier (retired when the Fully-Qualified PCID (X′60′) control vector is included)
15	0–2	Reserved
	3	Retired

INIT-OTHER-CD (INITIATE-OTHER CROSS-DOMAIN)

Byte	Bit	Content
	4–6	Reserved
	7	0 backup session is not requested 1 backup session is requested
16		NOTIFY specifications:
	0–1	retired
	2–3	Sending of RELREQ NOTIFY to LUs in session with LU2: 00 do not send RELREQ 01 reserved 10 send RELREQ if request is queued 11 reserved
	4	Reserved
	5	NOTIFY(X′03′) condition: 0 do not send NOTIFY to the SSCP(ILU) when the requested session is set up 1 send NOTIFY to the SSCP(ILU) when the requested session is set up
	6–7	Reserved
17–24		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
25 – m		Network Name of LU1
25		Type: X′F3′ logical unit
26		Length, in binary, of symbolic name
27 – m		Symbolic name, in EBCDIC characters
m + 1 – n		Network Name of LU2
m + 1		Type: X′F3′ logical unit
m + 2		Length, in binary, of symbolic name
m + 3 – n		Symbolic name, in EBCDIC characters
n + 1 – n + 2		Retired
n + 3 – r		User Field
n + 3		Length, in binary, of user data *Note:* X′00′ = no user data is present.
n + 4 – r		User data: user-specific data that is passed to the primary LU on the CINIT request
n + 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		
n + 5 – r		Remainder of unstructured user data
For structured user data		
n + 5 – r		Structured subfields (For detailed definitions, see Chapter 8, “User Data Structured Subfields.”) *Note:* With the exception of the NS header and PCID, all the fields in the INIT-OTHER-CD RU are derived from its corresponding INIT-OTHER RU.

INIT-OTHER-CD (INITIATE-OTHER CROSS-DOMAIN)

Byte	Bit	Content
End of Format 0; Formats 2 and 3 continue below.		
r+1	0	COS name initialization indicator: 0 ILU did not specify COS name 1 ILU did specify COS name
	1–7	Reserved
r+2 – r+9		COS name (reserved if byte r+1, bit 0 = 0): symbolic name of class of service in EBCDIC characters (A value of eight space [X′40′] characters may be specified; in this case, the COS name is derived from the mode name table using the mode name received in bytes 17–24.)
End of Format 2; Format 3 continues below.		
r+10 – r+17		Network ID of subnetwork in which the ILU is located and the names of LU1 (bytes 27 – m) and LU2 (bytes m+3 – n) are known
r+18 – r+25		Network ID of subnetwork in which the following COS name is defined
r+26 – r+33		COS name as known in the above network *Note:* Bytes r+26 – r+33 contain the class of service name that results from translation of the COS name in bytes r+2 – r+9 to the corresponding COS name in the network indicated in bytes r+18 – r+25.
r+34 – r+41		Mode name as known in the network of the target LU
r+42 – s		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′19′ Resource Identifier control vector for LU1 (always present) X′19′ Resource Identifier control vector for LU2 (always present) *Note:* One of the LUs is selected by the SSCP(ILU) as the target LU. The X′19′ vector that represents the selected LU has the Target Resource Indicator bit set to 1. X′19′ Resource Identifier control vector for the ILU of the 3rd party initiated session. (conditionally present) X′34′ LU Definition Override control vector: present if model terminal support override values are present X′5C′ APPN Message Transport control vector X′5D′ Subarea specific capability flags control vector (conditionally present for LU definition override) X′5F′ Extended Fully Qualified PCID control vector: conditionally present X′60′ Fully Qualified PCID control vector (always present)

INITPROC (INITIATE PROCEDURE)

SSCP→PU T4|5, Norm; FMD NS(c)

INITPROC is sent to the subarea PU adjacent to a PU T2 in order to initiate a PU T4|5-PU T2 load operation.

INITPROC (INITIATE PROCEDURE)

Byte	Bit	Content
0– 2		X′410235′ NS header
3– 6		Reserved
7– 8		Element address of PU T2 for which the procedure is to be initiated, if ENA is supported; otherwise, its network address
9		Procedure type: X′00′ load (only value defined)
For procedure type = load		
10– 17		IPL load module: an 8-character EBCDIC symbolic name of the IPL load module to be sent to the PU identified in bytes 7– 8

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
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 1 DLU is SLU
	7	0 initiate only (I): do not enqueue. 1 initiate/enqueue (I/Q): enqueue the request if it cannot be satisfied immediately
4– 11		Mode name: an 8-character symbolic name (implementation and installation dependent) that identifies the set of rules and protocols to be used for the session; used by the SSCP(SLU) to select the BIND image that will be used by the SSCP(PLU) to build the CINIT request
12 – m		Uninterpreted Name of DLU
12		Type: X′F3′ logical unit
13		Length, in binary, of DLU name
14 – m		EBCDIC character string
m + 1 – m + 2		Retired

INIT-SELF Format 0 (INITIATE-SELF)

Byte	Bit	Content
m + 3 – n		User Field
m + 3		Length, in binary, of user data *Note:* X′00′ = no user data is present.
m + 4 – n		User data: user-specific data that is passed to the primary LU on the CINIT request
m + 4		User data key: X′00′ structured subfields follow ¬ X′00′ first byte of unstructured user data *Note:* Individual structured subfields may be omitted entirely. When present, they appear in ascending field number order.
For unstructured user data		
m + 5 – n		Remainder of unstructured user data
For structured user data		
m + 5 – n		Structured subfields (For detailed definitions, see Chapter 8, “User Data Structured Subfields” on page 8-1.)
n + 1 – p		Control vectors, as described in the section “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′0A′ User Request Correlation (URC) control vector (included by the DLUR) X′34′ LU Definition Override control vector (present if terminal operator entered one or more of the model terminal support override parameters with an implementation logon request)
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

INIT-SELF Format 1 (INITIATE-SELF)

Byte	Bit	Content
3	0– 3	Format: 0001 Format 1: specifies queuing, initiate origin, and URC in addition to the parameters in Format 0
	4– 7	Reserved
4		Type:
	0– 1	01 initiate only (I): do not enqueue 11 initiate/enqueue (I/Q): enqueue the request if it cannot be satisfied immediately (See byte 5 for further specification of queuing conditions.)
	2– 5	Reserved
	6	PLU/SLU specification: 0 DLU is PLU 1 DLU is SLU
	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 (only value defined for LU 6.2) 10 enqueue this request LIFO, i.e., the request will be dequeued before the requests already in the queue. 11 reserved
	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		Reserved for LU 6.2; otherwise:
	0	Backup request: 0 Backup session is not requested. 1 Backup session is requested.
	1– 2	Reserved
	3	Retired
	4– 7	Reserved
7	0– 1	Retired
	2– 7	Reserved
8– 1 5		Mode name: an 8-character symbolic name (implementation- and installation-dependent) that identifies the set of rules and protocols to be used for the session; used by the SSCP(SLU) to select the BIND image that will be used by the SSCP(PLU) to build the CINIT request
16 – n		Uninterpreted Name of DLU
16		Type: X′F3′ logical unit
17		Length, in binary, of DLU name
18 – n		EBCDIC character string

INIT-SELF Format 1 (INITIATE-SELF)

Byte	Bit	Content
n+1 – n+2		Retired
n+3 – r		User Field (reserved for LU 6.2)
n+3		Length, in binary, of user data *Note:* X′00′ = no user data is present.
n+4 – r		User data: user-specific data that is passed to the primary LU on the CINIT request
n+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		
n+5 – r		Remainder of unstructured user data
For structured user data		
n+5 – r		Structured subfields (For detailed definitions, see Chapter 8, “User Data Structured Subfields” on page 8-1.)
r+1 – s		User Request Correlation (URC) Field
r+1		Length, in binary, of URC *Note:* X′00′ = no URC. (The length field is always present.)
r+2 – s		URC: LU-defined identifier; may be returned by the SSCP in a subsequent NOTIFY to correlate a given session to this initiating request
s+1 – t		Control vectors, as described in the section “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′0E′ Network Name control vector (present if needed to specify the network-qualified PLU name) X′34′ LU Definition Override control vector (present if terminal operator entered one or more of the model terminal support override parameters with an implementation logon request)

INOP (INOPERATIVE)

PU T4|5→SSCP, PU→PUCP, Norm; FMD NS(c)

INOP is sent to the SSCP by the PU to report a link-related connection or contact failure involving one or more nodes.

INOP (INOPERATIVE)

Byte	Bit	Content
0– 2		X′010281′ NS header

INOP (INOPERATIVE)

Byte	Bit	Content
3–4		Element address of an inoperative link or adjacent link station, if ENA is supported; otherwise, its network address
5	0–3	Format: X′0′ *Note:* The value X′F′ is set aside for implementation use and will not be further defined in SNA.
	4–7	Reason: X′1′ adjacent link station: loss of contact, unexpected loss of connection, or connection establishment failure. The SSCP should attempt recovery. X′2′ link: link failure. The SSCP should not attempt recovery. X′3′ adjacent link station: The secondary station has received a disconnect command from the primary station. In SDLC this is a Disconnect (DISC) command. In CDLC this is a Discontact channel command. The SSCP should attempt recovery. X′4′ adjacent link station: The primary station has received an asynchronous request for disconnect from the secondary station. In SDLC this is a Request Disconnect (RD) response. In CDLC this is asynchronous DE+UC status with sense containing Abort. The SSCP should attempt recovery. X′5′ adjacent link station: The primary station has received synchronous notification from the secondary station that the secondary has disconnected. In SDLC this is a Disconnected Mode (DM) response. In CDLC this is synchronous UC status (possibly with CE and/or DE) with sense containing Abort. The SSCP should attempt recovery. X′6′ adjacent link station: IPL or DUMP in progress. The SSCP should attempt recovery. X′7′ adjacent link station: RPO in progress. The SSCP should not attempt recovery. X′8′ link: link reset by DACTLINK type X′01′ (unconditional reset). The SSCP should not attempt recovery. X′A′ adjacent link station: CCITT X.21 outgoing call establishment X.21 call progress signal was deactivated; it should return to the ready state. X′B′ adjacent link station: CCITT X.21 call establishment failure; DCE Clear Indication or DCE Controlled Not Ready. (Either case may apply, with or without T6 timeout.) The link should not be deactivated; it should return to the ready state. X′C′ adjacent link station: CCITT X.21 outgoing call establishment failure; X.21 timeout T1, T2 or T3 expired. The link should not be deactivated; it should return to the ready state. X′D′ adjacent link station: unexpected loss of connection (DCE Clear Indication, with or without T6 timeout) during the CCITT X.21 data transfer phase. The link should not be deactivated; it should return to the ready state. X′E′ adjacent link station: CCITT X.21 call clearing failure; X.21 timeout T5 expired. The link should not be deactivated; it should return to the ready state. X′F′ adjacent link station: CCITT X.21 outgoing call establishment failure; X.21 call progress signals was received; the signal is included in bytes 6–7. The link should not be deactivated; it should return to the ready state. *Note 1:* The SSCP uses the INOP code to determine whether recovery needs to be attempted on the resource identified by the network address in the INOP. The type of resource, adjacent link station or link, is irrelevant to the SSCP. *Note 2:* INOP codes that are unknown to the SSCP are treated as indicating unrecoverable errors (no recovery action) whether the associated network address is for a link or adjacent link station. *Note 3:* The only adjacent link station INOP code that indicates an unrecoverable error is X′7′. *Note 4:* The SSCP considers the ALS to be active for codes X′A′ – X′F′.

INOP (INOPERATIVE)

Byte	Bit	Content
6–7		The CCITT X.21 call progress signal last received—included only if byte 5, bits 4–7 = X′F′; otherwise, these bytes are omitted (The codes and meanings of these X.21 call progress signals are as described in the CCITT recommendation X.21.)
End of Format 0; Format 1 continues below.		
5	0–3	Format: X′1′
	4–7	Reason: X′0′ adjacent link station: ISDN interface error. Reason code is included in byte 6. X′1′ link: ISDN interface error. Reason code is included in byte 6. X′2′ link: link failure. SSCP should not attempt recovery. This logical link outage was caused by the failure of a higher level resource (i.e. the physical link). *Note 1:* The SSCP uses the INOP code to determine whether recovery needs to be attempted on the resource identified by the network address in the INOP. The type of resource, adjacent link station or link, is irrelevant to the SSCP. *Note 2:* INOP codes that are unknown to the SSCP are treated as indicating unrecoverable errors (no recovery action) whether the associated network address is for a link or adjacent link station.
6		ISDN reason code included only if byte 5, bits 4–7 = X′0′ or X′1′; otherwise, omitted.

IPLFINAL (IPL FINAL)

SSCP→PU T4, Norm; FMD NS(c)

IPLFINAL completes an IPL sequence and supplies the load-module entry point to the T4 node. A positive response to IPLFINAL indicates that the T4 node is successfully loaded, or the load module has been successfully added to or replaced on the T4 node's local disk.

IPLFINAL (IPL FINAL)

Byte	Bit	Content
0–2		X′010205′ NS header
3–4		One of the following addresses: • Element address of adjacent link station associated with the node to be loaded, if ENA is supported; otherwise, its network address • Element address (X′0000′) of the receiving PU itself when a load module is to be added or replaced on its local disk
5–8		Entry point location within load module *Note:* This field is ignored when loading from disk.

IPLFINAL (IPL FINAL)

Byte	Bit	Content
9		IPL and Dump indicators (reserved when the address in bytes 3–4 specifies the element address of the receiving PU itself):
	0	IPL save load module indicator: 0 Do not save the IPL load module on local disk. 1 Save the IPL load module on local disk. *Note:* This bit is reserved when loading from local disk.
	1	Usage indicator for automatic dump/load switches: 0 IPL using the IPL/dump indicators in bits 2 and 3 of this byte. 1 IPL ignoring the IPL/dump indicators (bits 2–3); use original settings.
	2	Disk automatic IPL indicator: 0 Reset the automatic disk re-IPL switch. 1 Set the automatic disk re-IPL switch.
	3	Disk automatic dump indicator: 0 Reset the automatic disk dump switch. 1 Set the automatic disk dump switch.
	4–7	Reserved

IPLINIT (IPL INITIAL)

SSCP→PU T4, Norm; FMD NS(c)

IPLINIT either initiates a DLC-level load of an adjacent T4 node from the T4 node receiving the IPLINIT, when the node to be loaded is identified by the adjacent link station address contained in the request; or initiates the adding, replacing, or purging of the load module on the local disk of the T4 node receiving the request when the address in the request is the network address of the PU T4 receiving the request. In the case of purging, no IPLFINAL is sent; a positive response to IPLINIT indicates that the load module has been successfully purged.

IPLINIT (IPL INITIAL)

Byte	Bit	Content
0–2		X′010203′ NS header
3–4		One of the following addresses: • Element address of adjacent link station associated with the node to be loaded, if ENA is supported; otherwise, its network address • Element address (X′0000′) of the receiving PU itself when a load module is to be added, replaced, or purged on its local disk
5		IPL indicators:
	0	Disk IPL indicator: 0 IPL using load module being sent from host 1 IPL using load module from local disk *Note:* This bit is reserved when the address in bytes 3–4 specifies the element address of the receiving PU itself.
	1–7	Reserved.
6–13		Load module name:

IPLINIT (IPL INITIAL)

Byte	Bit	Content
14		Local disk indicators:
	0– 1	Load module indicators: 00 Add load module to local disk. 01 Replace load module on local disk. 10 Purge load module from local disk. 11 reserved
	2– 7	Reserved.

IPLTEXT (IPL TEXT)

SSCP→PU T4, Norm; FMD NS(c)

IPLTEXT transfers load module information to the PU T4, which passes it in a DLC-level load to the T4 adjacent node or adds or replaces the load module on its local disk. Following an IPLINIT, any number of IPLTEXT commands are valid, except that for purging and loading from local disk, IPLTEXT is not sent.

IPLTEXT (IPL TEXT)

Byte	Bit	Content
0– 2		X′010204′ NS header
3– 4		One of the following addresses: • Element address of adjacent link station associated with the node to be loaded, if ENA is supported; otherwise, its network address • Element address (X′0000′) of the receiving PU itself when a load module is to be added or replaced on its local disk
5 – n		Text: a variable-length byte-string in the form required by the node being loaded

LCP (LOST CONTROL POINT)

PU T4|5→SSCP, PU T4→PUCP, Norm; FMD NS(c)

LCP notifies the SSCP that a subarea PU's session with another SSCP has failed. The SSCP displays this information for the network operator.

LCP (LOST CONTROL POINT)

Byte	Bit	Content
0– 2		X′410287′ NS header

LCP (LOST CONTROL POINT)

Byte	Bit	Content
3		Reason code, specifying why LCP was generated: X′07′ virtual route inoperative: VR-INOP received for the virtual route used by the SSCP-PU session (where the SSCP is the lost control point identified later, and the PU is the originator of the LCP) X′0A′ forced deactivation of the SSCP-PU session (DACTPU(¬ SON) received by the PU) X′0B′ virtual route deactivated: NC-DACTVR(Forced) received for the virtual route used by the SSCP-PU session (where the SSCP is the lost control point identified later and the PU is the originator of the LCP) X′0C′ SSCP failure: the session between this PU and the identified SSCP was reset because of an abnormal termination of the SSCP (DACTPU(SON,Cause = X′0C′) was received by the PU)
4		Reserved
5–10		Network address of the lost control point (SSCP)

LDREQD (LOAD REQUIRED)

PU T2→SSCP, Norm; FMD NS(c)

(Retired RU) LDREQD has been retired from SNA. Consult product documentation for further information and support.

LSA (LOST SUBAREA)

PU T4|5→PU T4|5, Exp; NC

(Retired RU) LSA has been retired from SNA. Consult product documentation for further information and support.

LUSTAT (LOGICAL UNIT STATUS)

LU→LU|SSCP, 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

LUSTAT (LOGICAL UNIT STATUS)

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

NC-ACTVR (ACTIVATE VIRTUAL ROUTE)

PU T4|5→PU T4|5; NC

NC-ACTVR initializes the state and attributes of the VR at each of its end nodes.

NC-ACTVR (ACTIVATE VIRTUAL ROUTE)

Byte	Bit	Content
0		X′0D′ request code
1– 2		Reserved
3		Format: X′01′ (only value defined)
4		Reserved
5– 6		Receive ERN mask: a bit is *on* if that ERN can be used to send PIUs to NC-ACTVR originator; multiple bits may be set to 1 (bit 0 corresponds to reverse ERN 0, bit 1 to reverse ERN 1, and so forth).
7– 8		Send ERN mask: a bit is *on* if that ERN can be used to send PIUs from the NC-ACTVR originator: exactly one bit is set to 1 (bit 0 corresponds to ERN 0, bit 1 to ERN 1, and so forth).
9– 1 0	0– 3 4– 1 5	Reserved Initial VR send sequence number
11		Reserved
12		Maximum window size permitted on the VR
13		Reserved
14		Minimum window size permitted on the VR
15– 1 6		Maximum PIU size permitted to be sent by the NC-ACTVR originator: X′0000′ no restriction (only value defined)
17– 1 8		Maximum PIU length permitted to be received by the NC-ACTVR originator: X′0000′ no restriction (only value defined)

NC-DACTVR (DEACTIVATE VIRTUAL ROUTE)

PU T4|5→PU T4|5; NC

NC-DACTVR deactivates a virtual route.

NC-DACTVR (DEACTIVATE VIRTUAL ROUTE)

Byte	Bit	Content
0		X′0E′ request code
1– 2		Reserved
3		Format: X′01′
4		Type X′01′ orderly: receiver of NC-DACTVR to deactivate the VR if there are no sessions on the VR X′02′ forced: receiver of NC-DACTVR to deactivate the VR even if there are sessions on the VR; it also results in session-outage notification for sessions using the VR

NC-ER-ACT (EXPLICIT ROUTE ACTIVATE)

PU T4|5→PU T4|5; NC

NC-ER-ACT is sent by the ER manager in a subarea node in order to activate an explicit route.

NC-ER-ACT (EXPLICIT ROUTE ACTIVATE)

Byte	Bit	Content
0		X′0B′ request code
1–2		Reserved
3		Format: X′01′ (only value defined)
4		Reserved
5		Explicit route length: initially set to 0 at the originating PU, incremented by 1 at each receiver of the original or propagated NC-ER-ACT
6		Maximum ER length, as specified by the request originator
7–10		Subarea address of the destination PU corresponding to the ERN specified in byte 12, bits 4–7
11	0	Route definition capability of RU sender: 0 RU sender does not allow route usage except by explicit installation definition 1 RU sender allows route usage without requiring explicit installation definition
	1–7	Reserved
12	0–3	Reserved
	4–7	ERN of the explicit route being activated
13–16		Subarea address of the PU that originated the NC-ER-ACT request
17–18		Reverse ERN mask: A bit is *on* if the corresponding ERN can be used to route to the originating subarea (bit 0 corresponds to ERN 0, bit 1 to ERN 1 and so forth).
19–20		Maximum PIU length allowed on the ER in the direction of flow of this NC-ER-ACT (where X′0000′ = no restriction)
21–28		Reserved
29–36		Activation request sequence identifier: an 8-byte binary value, generated by the originator of NC-ER-ACT, and included by the destination node in NC-ER-ACT-REPLY to correlate an NC-ER-ACT with its corresponding NC-ER-ACT-REPLY (The 8-byte field has the following characteristic: If n1 was generated at time t1, and n2 was generated at time t2, then t1 < t2 implies n1 < n2.)

NC-ER-ACT-REPLY (EXPLICIT ROUTE ACTIVATE REPLY)

PU T4|5→PU T4|5; NC

NC-ER-ACT-REPLY is returned to signal the successful or unsuccessful completion of the NC-ER-ACT.

NC-ER-ACT-REPLY (EXPLICIT ROUTE ACTIVATE REPLY)

Byte	Bit	Content
0		X′0C′ request code
1–2		Reserved
3		Format: X′01′ (only value defined)
4		Type X′00′ explicit route activated X′01′ race condition resulting from NC-ER-ACT being sent by both nodes, each of which allows routing usage without requiring explicit installation definition; this condition is resolved in favor of the NC-ER-ACT from the PU having the greater subarea address (thus, this Type code is sent by the PU having the larger subarea address) X′02′ ER is not reversible since there is no reverse ERN defined X′03′ Encountered a node that does not support the ER X′04′ ER length exceeded the maximum specified in NC-ER-ACT X′05′ ER requires a TG that is not active X′06′ ER is not defined in the NC-ER-ACT-REPLY originating node X′07′ Retired
5		Explicit route length, in terms of the number of transmission groups in the explicit route as accumulated by NC-ER-ACT
6		Maximum ER length, as specified in NC-ER-ACT request
7–10		Subarea address of the destination PU of corresponding NC-ER-ACT
11		Reserved
12	0–3 4–7	Reserved ERN of the ER being activated
13–16		Subarea address of the PU originating the corresponding NC-ER-ACT
17–18		Reverse ERN mask: A bit is *on* if the corresponding ERN can be used to route to the NC-ER-ACT originating subarea (bit 0 corresponds to ERN 0, bit 1 to ERN 1, and so forth).
19–20		Maximum size of PIU allowed to flow on the reverse ERNs specified in bytes 17–18 (where X′0000′ = no restriction)
21–22		Maximum PIU length allowed on the ER in the direction of the flow of the NC-ER-ACT (where X′0000′ = no restriction)
23–28		Reserved
29–36		Activation request sequence identifier: same value as specified in the corresponding NC-ER-ACT
37–38		Reserved
39–42		Subarea address of the node that originated this NC-ER-ACT-REPLY

NC-ER-ACT-REPLY (EXPLICIT ROUTE ACTIVATE REPLY)

Byte	Bit	Content
43–46		Subarea address depending on the Type field (byte 4), as follows: **Type** **Contents of this field** X′00′ reserved X′01′ reserved X′02′ subarea on the ER prior to that with no reverse ERN defined X′03′ subarea that does not support the ER X′04′ subarea on the ER preceding the subarea where the explicit route length (byte 5 of NC-ER-ACT) is incremented to a value one more than the maximum ER length limit (byte 6) X′05′ subarea on the other end of the TG that is not active X′06′ subarea on the ER from which the PU (that does not have the ER defined) received the corresponding NC-ER-ACT X′07′ reserved
47		TGN of the TG between the subareas specified in bytes 39–42 and 43–46; reserved if Type is X′00′ or X′01′
48		Reserved

NC-ER-INOP (EXPLICIT ROUTE INOPERATIVE)

PU T4|5→PU T4|5; NC

NC-ER-INOP is initiated when the last remaining link of the transmission group has failed or is discontacted via a link-level procedure.

NC-ER-INOP (EXPLICIT ROUTE INOPERATIVE)

Byte	Bit	Content
0		X′06′ request code
1–2		Reserved
3		Format: X′01′ (only value defined)
4		Reason code: X′01′ unexpected routing interruption over a transmission group; e.g., the last active link in a TG has failed X′02′ controlled routing interruption, such as the result of DISCONTACT
5–8		Subarea address of the PU that originated the NC-ER-INOP
9–12		Subarea address on other end of the transmission group that had the routing interruption
13		TG number of the transmission group that had the routing interruption
14		Number of destination subareas that are on the ERs using the above TG
15–20		Inoperative ER Field
15–18		Subarea address of a destination that is routed to using an ER requiring the TG that had the routing interruption

NC-ER-INOP (EXPLICIT ROUTE INOPERATIVE)

Byte	Bit	Content
19–20		Inoperative explicit route mask: A bit is *on* if the ER of the corresponding ERN is inoperative (bit 0 corresponds to ERN 0, bit 1 corresponds to ERN 1, and so forth).
21 – n		Any additional 6-byte entries in the same format as bytes 15–20

NC-ER-OP (EXPLICIT ROUTE OPERATIVE)

PU T4|5→PU T4|5; NC

NC-ER-OP is generated when a link of an inoperative transmission group becomes operative.

NC-ER-OP (EXPLICIT ROUTE OPERATIVE)

Byte	Bit	Content
0		X′0F′ request code
1–2		Reserved
3		Format: X′01′ (Only value defined)
4		Reserved
5–8		Subarea address of the PU that originated the NC-ER-OP
9–12		Subarea address on the other end of the operational TG if the TG just became operational *Note:* This field is set to 0 in the case that the TG is already operational and has just been defined (e.g., by an operator command) to the node originating the NC-ER-OP for use with the specified ER′s.
13		TG number of the operational TG if the TG just became operational *Note:* This field is set to 0 in the case that the TG is already operational and has just been defined (e.g., by an operator command) to the node originating the NC-ER-OP for use with the specified ER′s.
14		Number of destination subareas that are routed to using the ERs requiring the above TG
15–20		Operative ER Field *Note:* This field is included if at least one operative ER exists for the subarea in bytes 15–18.
15–18		Subarea address of a destination that is routed to using an ER requiring the above TG
19–20		Operative explicit route mask: A bit is *on* if the ER for the corresponding ERN is operative (bit 0 corresponds to ERN 0, bit 1 to ERN 1, and so forth).
21 – n		Any additional 6-byte field entries in the same format as bytes 15–20

NC-ER-TEST (EXPLICIT ROUTE TEST)

PU T4|5→PU T4|5; NC

NC-ER-TEST is sent by a subarea node that requires testing of an explicit route to a specified destination subarea.

NC-ER-TEST (EXPLICIT ROUTE TEST)

Byte	Bit	Content
0		X′09′ request code
1–2		Reserved
3		Format: X′01′ (only value defined)
4		Reserved
5		Explicit route length: initially set to *0* by the PU that originated the NC-ER-TEST, incremented by 1 at each receiver of the original or propagated NC-ER-TEST
6		Maximum ER length (number of TGs comprising the ER), specified by the request originator
7–10		Subarea address of the destination of ER corresponding to the ERN specified in byte 12, bits 4–7
11		Reserved
12	0–3	Reserved
	4–7	ERN of the explicit route being tested
13–16		Subarea address of the PU that originated the NC-ER-TEST
17–18		Reverse ERN mask: A bit is *on* if the corresponding ERN can be used to route to the originating subarea (Bit 0 corresponds to ERN 0, bit 1 to ERN 1 and so forth.)
19–20		Maximum size of PIU allowed on the ERN specified in byte 12, bits 4–7 (where X′00′ = no restriction)
21–22		Reserved
23–28		Network address of the SSCP that originated the corresponding NS request
29–38		Request correlation value: an implementation-defined value returned in NC-ER-TEST-REPLY for correlation of reply to request

NC-ER-TEST-REPLY (EXPLICIT ROUTE TEST REPLY)

PU T4|5→PU T4|5; NC

NC-ER-TEST-REPLY is returned to signal the successful or unsuccessful completion of the NC-ER-TEST.

NC-ER-TEST-REPLY (EXPLICIT ROUTE TEST REPLY)

Byte	Bit	Content
0		X′0A′ request code
1–2		Reserved
3		Format: X′01′ (only value defined)
4		Type: X′00′ The corresponding NC-ER-TEST reached its destination subarea. X′02′ ER not reversible since there is no reverse ERN defined X′03′ Encountered a node that does not support the ER X′04′ ER length exceeded the limit specified in the NC-ER-TEST request. X′05′ ER requires a TG that is not active X′06′ ER is not defined in the NC-ER-TEST-REPLY originating node. X′07′ retired
5		Explicit route length, in terms of number of transmission groups in the explicit route as accumulated in NC-ER-TEST
6		Maximum ER length, as specified in the NC-ER-TEST request
7–10		Subarea address of the destination PU for corresponding NC-ER-TEST
11		Reserved
12	0–3 4–7	Reserved ERN of the ER being tested
13–16		Subarea address of the PU that originated the corresponding NC-ER-TEST
17–18		Reverse ERN mask: A bit is *on* if the corresponding ERN can be used to route to the originating subarea.
19–20		Maximum PIU size permitted on the reverse ERN specified in bytes 17–18 (where X′0000′ = no restriction)
21–22		Maximum PIU size accumulated by the NC-ER-TEST (where X′0000′ = no restriction)
23–28		Network address of the SSCP originating the corresponding NS test request
29–38		Request correlation field: same value as specified in the corresponding NC-ER-TEST
39–42		Subarea address of the PU that originated this NC-ER-TEST-REPLY
43–46		Subarea address depending on the type field (byte 4) as follows: **Type** **Contents of this field** X′00′ reserved X′02′ subarea on the ER prior to that with no reverse ERN defined X′03′ subarea that does not support the ER X′04′ subarea on the ER preceding the subarea where the explicit route length (byte 5 of NC-ER-TEST) is incremented to a value one more than the maximum ER length limit (byte 6) X′05′ subarea on the other end of the TG that is not active X′06′ subarea on the ER from which the PU (that does not have the ER defined) received the corresponding NC-ER-TEST
47		TGN of the TG between the subareas specified in bytes 39–42 and 43–46; reserved if Type is X′00′ *Note:* For nodes supporting generation of the X′1F′ Format 1 continues below.

NC-ER-TEST-REPLY (EXPLICIT ROUTE TEST REPLY)

Byte	Bit	Content
48 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′1F′ ER Configuration control vector (One X′1F′ control vector may be included by each node that supports the X′1F′ control vector along the route being tested.)

NC-IPL-ABORT (NC IPL ABORT)

PU T4|5→PU T2; NC

NC-IPL-ABORT contains sense data indicating the reason for a failure during IPL.

NC-IPL-ABORT (NC IPL ABORT)

Byte	Bit	Content
0		X′46′ request code
1– 4		Sense data

NC-IPL-FINAL (NC IPL FINAL)

PU T4|5→PU T2; NC

NC-IPL-FINAL contains the entry point location of the IPL module.

NC-IPL-FINAL (NC IPL FINAL)

Byte	Bit	Content
0		X′02′ request code
1– 4		Entry point location (hexadecimal address) within load module

NC-IPL-INIT (NC IPL INITIAL)

PU T4|5→PU T2; NC

NC-IPL-INIT is sent from a PU T4|5 to a PU T2 after the PU T4|5 processes an INITPROC(Type=IPL) RU.

NC-IPL-INIT (NC IPL INITIAL)

Byte	Bit	Content
0		X′03′ request code
1		Reserved
2– 9		IPL load module: an 8-character EBCDIC symbolic name of the IPL load module to be transmitted

NC-IPL-TEXT (NC IPL TEXT)

PU T4|5→PU T2; NC

NC-IPL-TEXT contains the IPL data.

NC-IPL-TEXT (NC IPL TEXT)

Byte	Bit	Content
0		X′04′ request code
1 – n		Text: A variable-length byte-string of IPL data, where the maximum value of n is 255

NMVT (NETWORK MANAGEMENT VECTOR TRANSPORT)

SSCP ←→ PU Norm; FMD NS(ma)

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

NMVT (NETWORK MANAGEMENT VECTOR TRANSPORT)

Byte	Bit	Content
0– 2		X′41038D′ NS header
3– 4		Retired: Set to network address by subarea node sender; set to 0, the PU local address, by peripheral node sender; ignored by receivers implementing the current level of SNA
5– 6	0– 1	Reserved
	2– 3	Retired: Set to 01 by subarea PU sender; set to 00 by peripheral node sender; ignored by receivers implementing the current level of SNA
	4– 1 5	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

NMVT (NETWORK MANAGEMENT VECTOR TRANSPORT)

Byte	Bit	Content
	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 *SNA Management Services Formats*.

NOTIFY (NOTIFY)

SSCP ←→ PU, Norm; FMD NS(c)

NOTIFY is used to synchronize awareness between the SSCP and PU of the status of a cross-network session.

NOTIFY is sent by the SSCP to inform the PU that a session initiation process could not complete. It is sent by the PU to inform the SSCP:

- That a session initiation sequence could not be completed because of inability to activate a VR
- That a session started normally
- That SON was received for a pending or pending active session
- That a session terminated normally

NOTIFY (NOTIFY)

Byte	Bit	Content
0– 2		X′410220′ NS header
3– 4		Element address of the PU, if ENA is supported; otherwise, its network address
5		NOTIFY vector key: X′05′ cross-network session synchronism: used to maintain synchronization between the SSCP and the PU for the mutually supported cross-network LU-LU or SSCP-SSCP session identified by the Network-Qualified Address Pair (X′15′) control vector (only value defined) X′11′ network identifier status: used to inform SSCP of a connection to a new network or the loss of a connection to a new network
6 – n		NOTIFY vector data

NOTIFY (NOTIFY)

Byte	Bit	Content
For NOTIFY vector key X′05′ (for the PU→SSCP flow)		
6		Cause: the PU sends NOTIFY to the SSCP when it has cleaned up the indicated cross-network session for the cause indicated by one of the following values: X′00′ VR activation failure: the gateway node was unable to activate a VR from the VRID list contained in bytes 12 – n. X′01′ session ended: the gateway node has received a response to a cross-network session-deactivation request X′02′ session-activation request rejected: the gateway node has received a negative response to a session activation request X′03′ session started: the gateway node has received a positive response to a session activation request
For NOTIFY vector key X′05′ (for the SSCP→PU flow)		
6		Cause: the SSCP sends NOTIFY to direct the PU to clean up the indicated cross-network session for the cause indicated by one of the following values: X′04′ session terminated: the SSCP is forcing the deactivation of the session (for example, because of operator action) X′05′ session setup failure: the SSCP has detected a session setup failure X′06′ session takedown failure: the SSCP has detected a session takedown failure
7– 1 0		If byte 6 = X′00′ or X′02′, this field contains the sense data from the negative response to a session activation request; otherwise, it contains 0s
11		Correlation indicators:
	0	Retired
	1– 7	Reserved
12 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL.

NOTIFY (NOTIFY)

Byte	Bit	Content
		X′15′ Network-Qualified Address Pair control vector Control vector X′15′ is used to identify the session to which this request applies and always precedes the additional control vectors, if present. The addresses may be in any order, except as noted below. • For the SSCP→PU flow, the addresses contained within the X′15′ control vector are as defined below: NAU1 and NAU2 are both addresses within the network identified by the Network ID field of the vector. NAU2 contains an alias address assigned within the gateway node receiving this vector. This order applies to the Address Pair session key on both the origin NAU and destination NAU side of the gateway node. Prior to the NAU Address control vector (X′1A′) being received by the gateway node, the X′15′ control vector contains the address pair for the network adjacent to the gateway node PU on the origin-NAU side of the PU. After X′1A′ has been received in the gateway node, the control vector may carry the session address pair on either side of the gateway node PU. • For the PU→SSCP flow, the control vector identifies the session in one network or the other as follows: – When the cause code (byte 6) is X′00′, VR activation failure, the control vector identifies the session in the network of the VR activation failure. – When the cause code (byte 6) is not X′00′, the control vector identifies the session in the network on the origin-NAU side of the gateway node PU. *Note:* The following additional control vectors may be used in NOTIFY(c) sent from the PU to an SSCP to carry information about a cross-network LU-LU session that could not be activated: X′1B′ VRID List control vector (this control vector included only for the PU-to-SSCP flow when byte 6 = X′00′—VR activation failure and the gateway node received a VR ID list for the session.) *Note:* The following additional control vectors may be used in NOTIFY(c) sent from the PU to SSCP to carry information about the cross-network LU-LU sessions that have been activated: X′1E′ VR-ER Mapping Data control vector *Note:* This information is the same as provided in the SESSST RUs that are sent by the LUs to the SSCPs with which they have active sessions. One pair of vectors specifies the required data for each side of the gateway. Thus, *four* vectors are appended to NOTIFY when the NOTIFY Vector Key field (byte 5) is set to X′05′ and the Cause field (byte 6) is set to X′03′. These vectors always appear in the order: • X′15′ • X′1E′ • X′15′ • X′1E′ In this use of control vector X′15′, NAU 1 is the PLU address and NAU 2 the SLU address.
For NOTIFY vector key X′11′ (for the PU→SSCP flow)		
6		Offset to control vectors
7		Action code

NOTIFY (NOTIFY)

Byte	Bit	Content
		X′01′ add network X′02′ delete network
8 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′12′ Network Identifier control vector

NOTIFY (NOTIFY)

SSCP→SSCP|LU, LU→SSCP, Norm; FMD NS(s)

NOTIFY is used to send information from an SSCP to another SSCP or 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 (for SSCP→LU and LU→SSCP)
0– 2		X′818620′ NS header (for SSCP→SSCP)

NOTIFY (NOTIFY)

Byte	Bit	Content
3 – p		One NOTIFY vector as described in detail below *Note:* One of the following NOTIFY vectors is included. X′01′ Resource Requested (retired—replaced by NOTIFY vector X′06′) X′03′ ILU/TLU or Third-party SSCP Notification: • ILU/TLU notification: used to inform the sender of an INIT or TERM request of the status of the procedure • third-party notification: used to inform a third-party SSCP (the SSCP whose LU issued an INIT-OTHER) of the status of the setup procedure X′04′ LU Deactivation (retired) X′06′ Resource Requested (replaces NOTIFY vector key X′01′): used to inform the current session partner (LU) that another LU wishes to use the resource. The NOTIFY request is routed to the SSCP of the requested LU′s current session partner by following some existing session setup path from the sending SSCP to the SSCP of the current session partner. *Note:* NOTIFY vector X′06′ is sent by an SSCP that supports it, as specified in the CDRM (X′06′) control vector, to an SSCP with the same capabilities or to an LU in its domain. X′07′ Resource Available (retired—replaced by NOTIFY vector X′08′) X′08′ Resource Available (replaces NOTIFY vector key X′07′): used to inform the sender of CDINIT that the required resource (DLU) is now available. *Note:* NOTIFY(X′08′) is sent by an SSCP that supports NOTIFY NS(s) key X′08′, as specified in the CDRM (X′06′) control vector, to an SSCP that has the same capabilities. X′09′ Cancellation of Request for Notification: used to cancel previous agreement for notification on resource availability. *Note:* NOTIFY(X′09′) is sent by an SSCP that supports NOTIFY NS(s) key X′09′, as specified in the CDRM (X′06′) control vector, to an SSCP that has the same capabilities. 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 X′10′ LU-LU Session Status: used to inform an SSCP of the referenced LU-LU session status (XRF-active or XRF-backup) change. This control vector is sent from a PLU to its SSCP, or from a boundary function of an SLU to the SSCP(SLU).

Resource Requested (retired) NOTIFY Vector

Resource Requested (retired) NOTIFY Vector

Byte	Bit	Content
0		Key: X′01′
1 – m		Network name of requested LU
1		Type: X′F3′ logical unit
2		Length, in binary, of symbolic name of LU
3 – m		Symbolic name in EBCDIC characters
m + 1 – n		Network name of requesting LU

Resource Requested (retired) NOTIFY Vector

Byte	Bit	Content
m + 1		Type: X′F3′ logical unit
m + 2		Length, in binary, of symbolic name
m + 3 – n		Symbolic name in EBCDIC characters

ILU/TLU or Third-party SSCP Notification NOTIFY Vector

ILU/TLU or Third-party SSCP 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 (i.e., all session activation signals have been sent or received) X′03′ procedure error X′04′ session started
2– 9		Retired
10		Reason (defined for Status field value of X′03′ only) *Note:* There are two encodings of the Reason byte: • If bit 4 = 0, the Reason byte is encoded for a setup procedure error. • If bit 4 = 1, the Reason byte is encoded for a takedown procedure error.
		Setup Procedure Error
	0	1 CINIT error in reaching the PLU
	1	1 BIND error in reaching the SLU
	2	1 setup reject at the PLU
	3	1 setup reject at the SLU
	4	0 setup procedure error
	5	Reserved
	6	1 setup reject at SSCP
	7	Reserved
		Takedown Procedure Error
	0	1 CTERM error in reaching the PLU
	1	1 UNBIND error in reaching the SLU
	2	1 takedown reject at the PLU
	3	1 takedown reject at the SLU
	4	1 takedown procedure error
	5	1 takedown reject at the SSCP
	6	0 (see following Note)
	7	Reserved *Note:* For bits 4 and 6, the bit combination of 11 is set aside for implementation internal use and will not be otherwise defined.
11– 14		Sense data (defined for Status value of X′03′ only)

ILU/TLU or Third-party SSCP Notification NOTIFY Vector

Byte	Bit	Content
15 – m		Session keys, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* The following session keys are used: X′06′ Network Name Pair: PLU and SLU X′15′ Network-Qualified Address Pair session key
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 *Note:* The URC length is 0 for SSCP-to-SSCP.
n + 1 – p		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′1C′ Network-Qualified Name Pair control vector (present only for LU-LU) X′5C′ APPN Message Transport control vector X′60′ Fully Qualified PCID control vector (present only for SSCP-to-SSCP)

LU Deactivation (retired) NOTIFY Vector

LU Deactivation (retired) NOTIFY Vector

Byte	Bit	Content
0		Key: X′04′
1		Type: X′01′ session count decremented; no corresponding INIT-SELF (only value defined)
2		Cause: cause of deactivating the (LU,LU) session, as specified in byte 4 of SESSEND
3		Action: any reactivation of the (LU,LU) session to be performed by either the PLU or SLU as specified in SESSEND
4 – m		Session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* One of the following session keys is used: X′07′ Network Address Pair: PLU and SLU, respectively X′15′ Network-Qualified Address Pair: PLU and SLU, respectively

Resource Requested NOTIFY Vector

Resource Requested NOTIFY Vector

Byte	Bit	Content
0		Key: X′06′
1 – m		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors are included; they are parsed according to subfield parsing rule KL. *Note:* The following three X′19′ control vectors are used in NOTIFY key X′06′ to identify the LUs involved in the resource request. X′19′ Resource Identifier control vector: identifies the current session partner of the requested LU (always present). This X′19′ control vector identifies the target of the NOTIFY. X′19′ Resource Identifier control vector: identifies the requested (single-session capable) LU (always present) X′19′ Resource Identifier control vector: identifies the requesting LU (always present) *Note:* The first two X′19′ control vectors always contain the indicated names. The third X′19′ control vector may not contain the indicated name. If the length of the third Resource Identifier control vector is 0, the requesting LU name is unavailable. X′5C′ APPN Message Transport control vector X′60′ Fully Qualified PCID control vector (present only for SSCP-to-SSCP)

Resource Available (retired) NOTIFY Vector

Resource Available (retired) NOTIFY Vector

Byte	Bit	Content
0		Key: X′07′
1– 4		Sense data: same as returned in corresponding -RSP(CDINIT)
5 – m		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* The following session key is used: X′01′ Network Name or Uninterpreted Name: this session key (X′01′) identifies the DLU of an earlier CDINIT request that failed because of the unavailability of a required resource. Session key X′01′ is used to signal the general availability of the resource to any and all LUs in the domain of the SSCP being notified. *Note:* This session key (X′01′) is applicable within a NOTIFY (X′07′) only for SSCP-to-SSCP. X′05′ PCID: The PCID session key carries the PCID used in the initial CDINIT attempt on the SSCP-SSCP session over which this NOTIFY flows. *Note:* This session key (X′05′) is applicable within a NOTIFY (X′07′) only for SSCP-to-SSCP. X′06′ Network Name Pair: OLU and DLU, respectively X′0A′ URC *Note:* Session keys X′06′ and X′0A′ are applicable within a NOTIFY(X′07′) only for SSCP-to-ILU. X′1C′ Network-Qualified Name Pair: This session key identifies an SSCP-based session that is now available for cross-network session initiation. The first name belongs to the SSCP originating the NOTIFY NS(s) RU and the second belongs to its session partner.

Resource Available NOTIFY Vector

Resource Available NOTIFY Vector

Byte	Bit	Content
0		Key: X′08′
1– 4		Sense data: same as returned in the corresponding -RSP(CDINIT) (retired) or in the Extended Sense Data control vector on the corresponding CDTERM that informed the receiver earlier that the resource was unavailable
5 – m		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′17′ SSCP Identifier control vector (retired) X′19′ Resource Identifier control vector: identifies the DLU named in a CDINIT request that has become available since the CDINIT request was processed (included if the available resource is the DLU) X′1C′ Network-Qualified Name Pair control vector X′5C′ APPN Message Transport control vector X′60′ Fully Qualified PCID control vector (present only for SSCP-to-SSCP)

Cancellation of Request for Notification NOTIFY Vector

Cancellation of Request for Notification NOTIFY Vector

Byte	Bit	Content
0		Key: X′09′
1– 4		Sense data describing the reason why a resource is not available
5 – m		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′19′ Resource Identifier control vector, which identifies the DLU for which an earlier agreement to notify on resource availability is being canceled X′1C′ Network-Qualified Name Pair control vector X′60′ Fully Qualified PCID control vector

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, in bytes, of LU-LU Session Capability field; valid values are 6 and 14.
2 – m		LU-LU Session Capability
2	0– 3	Primary LU capability (used between a subarea LU and its SSCP; also used between a peripheral LU and its SSCP if the BF(LU) supports the receipt of CINIT; otherwise, reserved): 0000 PLU capability is inhibited: sessions can be neither queued nor started 0001 PLU capability is disabled: sessions can be queued but not started 0010 reserved 0011 PLU capability is enabled: sessions can be queued or started
	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′0000′ by a subarea LU and to X′0001′ by a peripheral LU)
5– 7		Retired
8– 1 5		Retired (set to all space (X′40′) characters, or omitted)
m + 1 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′64′ TCP/IP Information control vector (optionally present if authorized by the SSCP to convey TCP/IP information associated with the TCP-connected client and if the Secondary LU Capability field indicates that secondary LU capability is enabled) *Note:* SSCP authorization is not required when the sender is a DLUR.

LU-LU Session Status NOTIFY Vector

LU-LU Session Status NOTIFY Vector

Byte	Bit	Content
0		Key: X′10′
1	0	Session State Data control vector status: 0 control vector built at time the switch occurred 1 control vector construction delayed and may therefore not reflect state of the session at the time the switch occurred
	1– 7	Reserved
2 – m		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields”: X′15′ network address pair: PLU and SLU, respectively

LU-LU Session Status NOTIFY Vector

Byte	Bit	Content
m+1 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′29′ Session State Data control vector

NS-IPL-ABORT (NS IPL ABORT)

SSCP→PU T2, Norm; FMD NS(c)

(Retired RU) NS-IPL-ABORT has been retired from SNA. Consult product documentation for further information and support.

NS-IPL-FINAL (NS IPL FINAL)

SSCP→PU T2, Norm; FMD NS(c)

(Retired RU) NS-IPL-FINAL has been retired from SNA. Consult product documentation for further information and support.

NS-IPL-INIT (NS IPL INITIAL)

SSCP→PU T2, Norm; FMD NS(c)

(Retired RU) NS-IPL-INIT has been retired from SNA. Consult product documentation for further information and support.

NS-IPL-TEXT (NS IPL TEXT)

SSCP→PU T2, Norm; FMD NS(c)

(Retired RU) NS-IPL-TEXT has been retired from SNA. Consult product documentation for further information and support.

NS-LSA (NS LOST SUBAREA)

PU T4|5→SSCP, Norm; FMD NS(c)

(Retired RU) NS-LSA has been retired from SNA. Consult product documentation for further information and support.

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
		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 keys, as described in “Session Keys” in Chapter 9, “Common Fields”: X′06′ uninterpreted name pair session key: PLU and SLU, respectively (always present) X′1C′ Network-qualified name pair session key (optionally present)
n + 1 – m		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL: X′0E′ Network Name control vector
Condensed Format		

NSPE (NS PROCEDURE ERROR)

Byte	Bit	Content
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

PROCSTAT (PROCEDURE STATUS)

PU T4|5→SSCP, Norm; FMD NS(c)

PROCSTAT reports to the SSCP either the successful completion or the failure of the load operation. If the procedure failed, the request code of the failing RU and sense data are included as parameters in the PROCSTAT RU.

PROCSTAT (PROCEDURE STATUS)

Byte	Bit	Content
0–2		X′410236′ NS header
3–6		Reserved
7–8		Element address of the PU for which this procedure was initiated, if ENA is supported; otherwise, its network address
9		Procedure type: X′00′ load (only value defined)
10		Procedure status: X′00′ successful (bytes 13–17 set to 0s) X′01′ reserved X′02′ failure occurred—procedure failure; bytes 13–17 contain additional information
11–12		Reserved
13–17		Status Qualifier

PROCSTAT (PROCEDURE STATUS)

Byte	Bit	Content
13		Request code of failing NC RU
14– l 7		Sense data returned in the -RSP for the failing NC RU

QC (QUIESCE COMPLETE)

LU→LU, Norm; DFC

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

QC (QUIESCE COMPLETE)

Byte	Bit	Content
0		X′81′ request code

QEC (QUIESCE AT END OF CHAIN)

LU→LU, Exp; DFC

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

QEC (QUIESCE AT END OF CHAIN)

Byte	Bit	Content
0		X′80′ request code

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

PU→SSCP, Norm; FMD NS(ma)

RECFMS has been retired from SNA for T2 nodes.

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

Consult product documentation for further information on product support.

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
0–2		X′410384′ NS header
3–4		CNM target ID, as specified in bytes 5–6, bits 2–3
5–6	0–1	Reserved
	2–3	CNM target ID descriptor: 00 byte 4 contains a local address for a PU or LU in a T2 node or an LSID for a PU or LU in a T1 node; byte 3 is reserved 01 bytes 3–4 contain the element address of a link, adjacent link station, PU, or LU in the origin subarea, if ENA is supported; otherwise, its network address
	4–15	Procedure related identifier (PRID) (see Note below)
7		Request-Specific Information
	0	Solicitation indicator: 0 unsolicited request 1 reply request
	1	Not-last request indicator: 0 last request in a series of related unsolicited or reply requests, e.g., last reply request in a series corresponding to a single soliciting request 1 not last request
	2–7	Request-specific type code (see below)
Note:		For reply (i.e., solicited) requests, bytes 3–6 and byte 7, bits 2–7, echo the corresponding fields in the CNM header received in the request that solicited the reply requests. For unsolicited requests, these fields—the CNM target ID descriptor, the CNM target ID, the PRID, and the request-specific information—are generated by the request sender. For unsolicited requests, the PRID field contains X′000′. The PU does not interleave requests belonging to different series of related unsolicited requests from the same target.
8–13		Node Identification
8–11	0–11	Block number: an IBM product-specific number; see the individual product specifications for the specific values used
	12–31	ID number: a binary value that, together with the block number, identifies a specific station uniquely within a customer network installation; the ID number can be assigned in various ways, depending on the product; see the individual product specifications for details
12–13		Reserved
7 – n		Alert (retired: supported only for PUs not at the current level of SNA)
7	0–1	00 (only value defined—Alert is always sent unsolicited and as a single RU)
	2–7	Type code: 000000
8–13		Node Identification
8–11	0–11	Block number
	12–31	ID number
12–13		Reserved
14–19		Alert Classification
14	0–1	Format: 01 format 1 (only value defined)
14	2–7	Reserved

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
15	0– 3	Alert type: indicates the reason for the Alert being generated and differentiates between errors, operational problems, performance problems, and other exceptional conditions; valid Alert types are: X′1′ permanent error: cannot be retried or recovered without help external to the SNA node X′2′ temporary error: recovered within recovery procedure limit X′3′ performance: exceeded performance parameter threshold X′4′ operational or procedure: unsupported or invalid use, busy X′5′ application generated X′6′ operator triggered X′7′ SNA summary: exceeded threshold count of SNA negative responses
	4– 7	Major probable cause: indicates the general category of the probable cause, e.g., hardware, software, or protocol failure; valid major probable cause (details of these causes are given in specific implementation documentation): X′1′ hardware X′2′ software X′3′ link connection: characterized by transmission medium, modem, DTE-DCE cable, drivers, X′4′ protocol: invalid response or command sequence, system definition error X′5′ environment: thermal, installation restriction X′6′ removable media, e.g., paper, cards, tape, pack, diskette X′7′ hardware or software X′8′ logical X′9′ operator of sending product X′F′ undetermined
16		Minor probable cause: indicates the lowest level category with which the Alert may be associated, e.g., printer, power, program, X.25 network; valid minor probable cause (details of these causes are given in specific implementation documentation): X′01′ base processor X′02′ service processor for support of maintenance services X′03′ microcode *Note:* Microcode may be classified as IBM Licensed Internal Code. See “Notices” at the beginning of this document for more information. X′04′ main storage X′05′ DASD drive X′06′ printer X′07′ card reader/punch X′08′ tape drive X′09′ keyboard X′0A′ selector pen X′0B′ magnetic stripe reader X′0C′ display or printer X′0D′ display unit X′0E′ remote product: error attributed to product at adjacent link station on this link X′0F′ power internal to this product X′10′ I/O attached controller if distinguishable from drive X′11′ communications controller scanner X′12′ communications link adapter X′13′ link adapter X′14′ channel adapter: secondary attachment to System/390 channel X′15′ loop adapter: attachment to loop communication link X′16′ adapter for directly attaching devices X′17′ miscellaneous adapter X′18′ System/390 channel X′19′ link: transmission medium—ownership unknown

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content	
		X′1A′	link: common carrier transmission medium
		X′1B′	link: customer transmission medium
		X′1C′	loop: transmission medium—ownership unknown
		X′1D′	loop: common carrier transmission medium
		X′1E′	loop: customer transmission medium
		X′1F′	X.21 link connection external to this product
		X′20′	X.25 network external to this product
		X′21′	local X.21 interface: DTE-DCE
		X′22′	local X.25 interface: DTE-DCE
		X′23′	local modem
		X′24′	remote modem
		X′25′	local modem interface: DTE-DCE
		X′26′	remote modem interface: DTE-DCE
		X′27′	local probe
		X′28′	remote probe
		X′29′	local probe interface
		X′2A′	remote probe interface
		X′2B′	network connection
		X′2C′	IBM host program if not distinguishable as control program, application, or access method
		X′2D′	IBM host application program supplied by IBM
		X′2E′	IBM host telecommunication access method
		X′2F′	customer host application program
		X′30′	IBM communication controller program
		X′31′	IBM control program
		X′32′	remote modem interface or remote product
		X′33′	transmission medium or remote modem
		X′34′	SDLC format exception
		X′35′	BSC format exception
		X′36′	start/stop format exception
		X′37′	SNA format exception
		X′38′	power external to product
		X′39′	thermal
		X′3A′	paper
		X′3B′	tape
		X′3C′	DASD: removable media
		X′3D′	card
		X′3E′	magnetic stripe card
		X′3F′	negative SNA response
		X′40′	system definition error (whether diskette loaded, keyed, or otherwise customized)
		X′41′	installation restrictions
		X′42′	adjacent link station offline: no status received
		X′43′	adjacent link station busy (switched link)
		X′44′	controller or device
		X′45′	local probe or modem
		X′46′	tape or drive
		X′47′	card reader/punch or display/printer
		X′48′	controller application program
		X′49′	keyboard or display
		X′4A′	storage control unit
		X′4B′	channel or storage control unit
		X′4C′	storage control unit or controller
		X′4D′	control unit
		X′4E′	DASD data or media or drive
		X′4F′	DASD data or media

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
		X'50' diskette X'51' diskette or drive X'FF' undetermined
17		Reserved
18		User action code: 0 reserved ¬0 a code associated with predefined text that describes user actions appropriate to the cause
19		Reserved
20 – m		Appended CNM vectors (described at the end of this RU): zero or more CNM vectors may be appended to the request to convey data available to the PUMS when the Alert event was originated; appended vectors are ordered according to the binary value of the Vector Type field (nondescending) *Note:* The sending of information in appended CNM vectors does *not* cause reset of any counters.
m+1(=n)		X'00' indicating end of appended vectors
7–17		SDLC Test Command/Response Statistics
7	0 1 2–7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000001; the CNM target ID identifies a PU T1|2
8–13		Node identification
	0–11	Block number
	12–31	ID number
12–13		Reserved
14–15		Counter: the number of times the secondary SDLC station has received an SDLC Test command with or without a valid FCS
16–17		Counter: the numberof times the secondary SDLC station has received an SDLC Test command with a valid FCS and has transmitted an SDLC Test response *Note:* All counters are in binary.
7–22		Summary error data
7	0 1 2–7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000010; the CNM target ID identifies a PU
8–13		Node identification
	0–11	Block number
	12–31	ID number
12–13		Reserved
14–16		Summary counter validity mask:
14	0 1 2 3–7	Set to 1 if product error counter is valid Set to 1 if communication adapter error counter is valid Set to 1 if SNA negative response counter is valid Reserved
15		Reserved

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

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

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
16		Mask byte 2 (bit is set to 1 if the counter is valid):
	0	Command reject counter
	1	SDLC DCE error counter
	2	Write time-out counter
	3	Invalid status counter
	4	Communication adapter machine check counter
	5–7	Reserved
17		Reserved
18		Nonproductive time-out counter: no valid SDLC frames have been received within the time interval specified by the communication adapter; or receive overrun counter: the line is "hung" or insufficient buffer space has been allocated *Note:* Receive overrun applies only to counter set 2.
19		Idle time-out counter: no SDLC Flag octets received for *n* seconds, where *n* is specified by the communication adapter
20		Write retry counter: the number of retransmissions of one or more SDLC I-frames
21		Overrun counter: the number of times one or more received characters have been overlaid
22		Underrun counter: the number of times one or more characters have been transmitted more than once
23		Connection problem counter: incremented by *1* for every *n* retries of commands that establish connection with a station, when RLSD drops, or whenever write retry is updated—*n* is specified by the communication adapter
24		FCS error counter: the number of times a received SDLC frame had an invalid FCS
25		Primary station abort counter: number of times seven or more consecutive *1* bits have been received
26		SDLC command reject counter
27		DCE error counter: number of DCE interrupts or other unexpected conditions (e.g., "data set ready" drops)
28		Write time-out counter: number of time-outs during write operations, e.g., because of transmit clock failures
29		Invalid status counter: number of times status generated by the adapter was not meaningful
30(=n)		Communication adapter machine check counter: number of times the communication adapter has been identified as causing a machine check *Note:* All counters are in binary.
15 – n		Data for Counter Set 3
15–17		Communication adapter counter validity mask (bit is set to 1 if the counter is valid):
15	0	Total transmitted I-frames counter
	1	Write retry counter
	2	Total received I-frames counter
	3	FCS error counter
	4	SDLC command reject counter
	5	DCE error counter
	6	Nonproductive time-out counter
	7	Reserved

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
16–17		Reserved
18–19		Total transmitted I-frames counter: the total number of SDLC I-frames transmitted successfully
20–21		Write retry counter: the number of retransmissions of one or more SDLC I-frames
22–23		Total received I-frames counter: the number of SDLC I-frames successfully received
24–25		FCS error counter: the number of SDLC frames received with FCS errors
26–27		SDLC command reject counter
28–29		DCE error counter: the number of DCE interrupts and other unexpected conditions (e.g., "data set ready" drops)
30–31 (=n)		Nonproductive time-out counter: the number of times an SDLC frame has not been received within the time interval specified by the adapter *Note:* All counters are in binary.
15 – n		Data for Counter Set 4 *Note:* For a definition of adapter, control unit, and System/390 channel commands, and orders see implementation documentation.
15–17		Adapter counter validity mask bytes
15		Mask byte 1 (bit is set to 1 if the counter is valid):
	0	Command-reject-while-not-initialized counter
	1	Command-not-recognized counter
	2	Sense-while-not-initialized counter
	3	Channel-parity-check-during-selection-sequence counter
	4	Channel-parity-check-during-data-write-sequence counter
	5	Output-parity-check-at-control-unit counter
	6	Input-parity-check-at-control-unit counter
	7	Input-parity-check-at-adapter counter
16		Mask byte 2 (bit is set to 1 if the counter is valid):
	0	Data-error-at-adapter counter
	1	Data-stop-sequence counter
	2	Short-frame-or-length-check counter
	3	Connect-received-when-already-connected counter
	4	Disconnect-received-while-PU-active counter
	5	Long-RU counter
	6	Connect-parameter-error counter
	7	Read-Start-Old-received counter
17		Reserved
18		Command-reject-while-not-initialized counter: an initial Control command containing a valid Connect order was not received prior to a Restart Reset, Read Start 0/1, Write Start 0/1, Read, Write, or Write Break command
19		Command-not-recognized counter: control unit channel adapter received a command code that it did not recognize (invalid or not supported)
20		Sense-while-not-initialized counter: Sense command was received in response to the initial asynchronous interrupt (device-end, unit check), or Sense command was received without a preceding unit check ending status
21		Channel-parity-check-during-selection-sequence counter: control unit channel adapter detected a parity error from the channel during the selection sequence from the channel

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
22		Channel-parity-check-during-data-write-sequence counter: control unit channel adapter detected a parity error on channel bus-out during a channel Write operation
23		Output-parity-check-at-control-unit counter: control unit channel adapter detected a control unit parity error during a channel Write operation
24		Input-parity-check-at-control-unit counter: control unit detected a control unit parity error during a channel Read operation
25		Input-parity-check-at-adapter counter: control unit channel adapter detected that it transmitted bad parity on channel bus-in during a channel Read operation
26		Data-error-at-adapter counter: control unit detected a channel adapter error during an internal channel adapter cycle-steal operation
27		Data-stop-sequence counter: the number of data bytes accepted by the System/390′s Read command was less than that specified in Connect
28		Short-frame-or-length-check counter: a minimum four bytes have not been transferred as a link header; or the byte count specified in the first two bytes of the header did not equal the number of bytes received during a Control, Write, or Write Break operation
29		Connect-received-when-already-connected counter: a Connect was received when the control unit was already connected; this is an error condition and the PU is deactivated
30		Disconnect-received-while-PU-active counter: a Disconnect order was received from the System/390 while the PU is active (i.e., with no DACTPU preceding the Disconnect); this is an error condition
31		Long-RU counter: primary link station has sent an RU greater than the secondary link station can accept
32		Connect-parameter-error counter: the Connect was rejected because it specified an odd-number buffer length, or it specified a buffer size insufficient to hold the link header, TH, RH, and at least a 64-byte RU
33(=n)		Read-Start-Old-received counter: the secondary link station received a Read Start Old command *Note:* All counters are in binary.
15 – n		Data for Counter Set 5 (for X.25 physical circuit) (retired: supported only for PUs not at the current level of SNA) *Note:* Sent only from the primary end of an X.25 physical circuit.
15– 17		Communication adapter counter validity mask
15		Mask byte 1 (bit is set to 1 if the counter is valid):
	0	Number of I-frames transmitted counter
	1	Number of I-frames received counter
	2	Number of RR frames transmitted counter
	3	Number of RR frames received counter
	4	Number of RNR frames transmitted counter
	5	Number of RNR frames received counter
	6	Number of REJ frames transmitted counter
	7	Number of REJ frames received counter

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

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

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
18–19		Number of I packets transmitted
20–21		Number of I packets received
22–23		Number of RR packets transmitted
24–25		Number of RR packets received
26–27		Number of RNR packets transmitted
28–29		Number of RNR packets received
30–31		Number of interrupt packets transmitted
32–33		Number of interrupt packets received
34–35		Total number of connection requests (call request and incoming calls)
36–37		Total number of connections (calls connected and accepted)
38–39		Number of reset indications
40–41		Number of clear indications
42–43		Number of data packets with D-bit transmitted
44–45 (=n)		Number of data packets with D-bit received *Note:* All counters are in binary.
7 – n		PU/LU Dependent Data
7	0 1 2–7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000100; the CNM target ID identifies a PU|LU
8–13		Node identification
	0–11 12–31	Block number ID number
12–13		Reserved
14 – n		PU/LU dependent data
7 – n		Engineering Change Levels
7	0 1 2–7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000101; the CNM target ID identifies a PU
8–13		Node identification
	0–11 12–31	Block number ID number
12–13		Reserved
14 – n		Implementation defined data describing hardware, microcode, and programming levels
7 – n		Link Connection Subsystem Data (retired: supported only for PUs not at the current level of SNA)
7	0 1 2–7	Solicitation indicator (see above) Not-last request indicator (see above) Type code: 000110; the CNM target ID identifies an adjacent link station in the origin subarea

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
8–13		Node identification:
	0–11	Block number
	12–31	ID number
12–13		Reserved
14		Data selection, echoed from the soliciting REQMS command: X′02′ link status command sequence X′03′ remote DTE interface status X′04′ remote modem self test
15		Link connection subsystem type: X′01′ link type 1 (links that use 3863, 3864, or 3865 modems; also links that use 5865, 5866, or 5868 modems running LPDA-1) X′02′ link type 2 (3867 link diagnostic unit)
16–17		Validity indicators, bits 0–9 (how the PU sending this RU views the data): *Note:* The values to follow are used in each of the validity indicator fields. 00 data valid, from the modem 01 data invalid, no response from the modem 10 data invalid, response in error from the modem 11 data invalid, execution not attempted by the PU sending this RU
	0–1	Remote modem status
	2–3	Local modem status
	4–5	Modem self test *Note:* If byte 14 = X′02′, bits 4–5 are for local modem self-test. If byte 14 = X′04′, bits 4–5 are for remote modem self-test.
	6–7	Reserved
	8–9	Remote DTE interface status
	10–13	Reserved
	14–15	Link connection subsystem data format indicator: 00 format 0 01 format 1: same as format 0, plus; remote modem self test results, channelization status, local and remote modem status extensions, and general status extensions
18–19		Remote modem status:
	0–5	Hit count (noise spikes) for link type 1, reserved for link type 2 *Note:* For bits 6–7 and 12–14, when the condition exists, the bit value will be 1.
	6	Modem reinitialization was performed
	7	Loss of receive line signal
	8–11	Quadratic error value for link type 1, number of byte errors during test for link type 2
	12	Remote DTE power off detected
	13	Data Terminal Ready loss detected
	14	Switched-Network-Back-Up connected
	15	DTE streaming condition detected
20–21		Local modem status:
	0–5	Hit count (noise spike) for link type 1, reserved for link type 2 *Note:* For bits 6–7 and 12–14, when the condition exists, the bit value will be 1.
	6	Modem reinitialization was performed
	7	Loss of receive line signal
	8–11	Quadratic error value for link type 1, number of byte errors during test for link type 2
	12	Remote modem power loss detected
	13	Speed,for link type 1 (always full for link type 2): 0 half 1 full
	14	Switched-Network-Back-Up connected
	15	Reserved

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
22–24		Local Modem Self-Test and Remote-Tone Results, Remote Modem Self-Test Results:
		Note: If byte 14 is X′02′, link status command sequence, then bytes 22–24 pertain to the local modem. If byte 14 is X′04′, remote modem self-test, then bytes 22–24 pertain to the remote modem.
	0–2	Model bits, concatenated to the right of the bit-string formed by bits 18, 19, 8, and 15 (in this order) represents the modem model returned as modem self-test result in the bit-string formed by bits 2 and 3 of byte 3, bits 0 and 7 of byte 2, and bits 0, 1, and 2 of byte 1 (in this order), see “LPDA-1 Results Message Information Fields” in *IBM 5865/5866 Modem Models 2, 3 Maintenance Information and Parts Catalog*, SY33-2048.
	3	Link connection type: 0 nonswitched 1 switched
	4	Configuration: 0 point to point 1 multipoint
	5	Modem role: 0 primary or control modem 1 secondary or tributary modem
	6	Clear To Send delay for link type 1 (reserved for link type 2): 0 normal 1 exceptional
	7	Received line signal detector sensitivity for link type 1 (reserved for link type 2): 0 normal 1 limited
	8	Model bit, see bits 0–2 specification
	9	Modem self-test result: 0 passed 1 failed
	10	Remote tone test result for local modem self test (reserved for remote modem self test): 0 passed 1 failed
		Note: For the following bits, when the condition exists, the bit value will be 1.
	11	Feature card suspected in error
	12	Receiver card suspected in error for link type 1 (reserved for link type 2)
	13	Receiver card extension suspected in error for link type 1 (reserved for link type 2)
	14	Front end card is suspected in error for link type 1 (reserved for type 2)
	15	Model bit, see bits 0–2 specification
	16	Feature card installed (tone alarm card installed if nonswitched link connection; integral protection coupler installed if switched link connection)
	17	Switched-Network-Back-Up installed
	18	Model bit, see bits 0–2 specification; also if its value is 1 then channelization feature installed
	19	Model bit, see bits 0–2 specification; also if its value is 1 then fan-out feature installed
	20–23	Microcode EC level
25–26		Remote DTE Interface Status
25		Current state of the RS-232C or V.24 interface leads (for bits 0–5 and 7, when the condition exists, the bit value is set to 1):
	0	Request To Send
	1	Clear To Send
	2	Reserved
	3	Transmit Data
	4	Reserved
	5	Data Terminal Ready

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
	6	Speed: 0 half 1 full
	7	DTE power loss
26		Indication of transition of RS-232C or V.24 leads since last test occurrence (for the following bits, when the condition exists, the bit value is set to 1):
	0	Request To Send changed at least once
	1	Clear To Send changed at least once
	2	Received Data changed state
	3	Transmit Data changed state
	4	Received Line Signal loss was detected at least once
	5	Data Terminal Ready dropped at least once
	6	Modem speed was changed at least once
	7	DTE power loss was detected at least once
End of format 0, Format 1 continues below.		
27–29		Channelization status
27		Channelization and tailing flags (for the following bits, when the condition exists, the bit value is set to 1):
	0	This data is associated with a channelized modem
	1	This data is associated with a tailed link of a channelized modem
	2	This data is associated with channel A of a channelized modem
	3–7	Reserved
28–29		Channelization correlation number: a user assigned value used to correlate link connections with a channelized modem. The same value may be assigned to each of the link connections of a channelized modem so that those link connections can be associated with that particular modem
30–37		Local Modem Status Extension
30		Local modem receive dB level (with all code points representing dB units): X'00' function not supported X'01' – X'40' ignore data X'41' not available X'42' – X'4B' < -48 dB X'4C' -48 dB X'4D' -47 dB X'4E' – X'60' -46 dB to -28 dB X'61' -27 dB X'62' – X'6B' -26 dB to -17 dB X'6C' -16 dB X'6D' – X'75' -15 dB to - 7 dB X'76' - 6 dB X'77' – X'7D' - 5 dB to + 1 dB X'7E' + 2 dB X'7F' > + 2 dB X'80' – X'FF' ignore data
31–37		Reserved
38–45		Remote Modem Status Extension
38		Remote modem receive dB level (with all code points representing dB units): X'00' function not supported X'01' – X'40' ignore data X'41' not available

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

Byte	Bit	Content
		X′42′ – X′4B′ < -48 dB X′4C′ -48 dB X′4D′ -47 dB X′4E′ – X′60′ -46 dB to -28 dB X′61′ -27 dB X′62′ – X′6B′ -26 dB to -17 dB X′6C′ -16 dB X′6D′ – X′75′ -15 dB to - 7 dB X′76′ - 6 dB X′77′ – X′7D′ - 5 dB to + 1 dB X′7E′ + 2 dB X′7F′ > + 2 dB X′80′ – X′FF′ ignore data
39–45		Reserved
46–53 (=n)		General status extension
46		Link-level address used to address the remote modem
47		Remote DTE Interface Extension
48–53 (=n)		Reserved
CNM Vectors (described 0-origin)		
(Retired: CNM vectors are supported only for PUs not at the current level of SNA)		
0		Vector length: a binary count of the length in bytes of this vector (bytes 1 – n)
1		Type field:
	0–1	Reserved
	2–7	Vector type: an identifier of the information contained in bytes 2 – n.
2 – n		Vector data
0 – n		Embedded Text Vector
0		Vector length: a binary count of the length in bytes of this vector (bytes 1 – n)
1		Type field:
	0–1	Reserved
	2–7	Vector type: 000000 the vector contains a text message, composed of SCS characters (only value defined)
2 – n		Vector data in SCS text
0 – n		Embedded Name List Vector
0		Vector length: a binary count of the length in bytes of this vector (bytes 1 – n)
1		Type field:
	0–1	Reserved
	2–7	Vector type: 001100
2		Hierarchy name options: X′01′ reserved X′02′ only value defined X′03′ retired
3		Number of name entries to follow

RECFMS (RECORD FORMATTED MAINTENANCE STATISTICS)

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

RECMS (RECORD MAINTENANCE STATISTICS)

PU T4|5→SSCP, Norm; FMD NS(ma)

(Retired RU) RECMS has been retired from SNA. Consult product documentation for further information and support.

RECSTOR (RECORD STORAGE)

PU T4→SSCP, Norm; FMD NS(ma)

RECSTOR carries the storage dump as requested by a DISPSTOR RU, or the names and related information for load modules and dump (if present) on the disk attached to the T4 node.

RECSTOR (RECORD STORAGE)

Byte	Bit	Content
0– 2		X′010334′ NS header
3– 4		Element address of resource to be displayed, if ENA is supported; otherwise, its network address

RECSTOR (RECORD STORAGE)

Byte	Bit	Content
5		Display source and type:
	0– 3	Source (address space) of storage display *Note:* Refer to implementation documentation for description of these values.
	4– 7	Display type: 0001 nonstatic storage display 0010 static snapshot display 1000 names and related information for load modules and dump (if present) on the disk attached to the T4 node
6		Reserved
7– 8		Number of bytes of program storage following in this record
9– 1 2		Beginning location
13 – n		Storage display

RECTD (RECORD TEST DATA)

PU T4|5→SSCP, Norm; FMD NS(ma)

RECTD returns the status and results of a test requested by EXECTEST to SSCP maintenance services.

RECTD (RECORD TEST DATA)

Byte	Bit	Content
0– 2		X′010382′ NS header
3– 4		Element address of resource under test, if ENA is supported; otherwise, its network address
5– 8		Binary code selecting the test
9 – n		Test status and results

RECTR (RECORD TEST RESULTS)

PU T4|5→SSCP, Norm; FMD NS(ma)

RECTR is the reply request corresponding to a TESTMODE request. It returns the results and status for the test. Multiple reply requests may be sent in answer to a single soliciting TESTMODE request. When TESTMODE initiates a continuous test, the RECTR(s) is sent in reply to the TESTMODE request that terminates the test. However, the PRID that is echoed in the CNM header of the replying RECTR is the PRID received in the TESTMODE that initiated the test.

RECTR (RECORD TEST RESULTS)

Byte	Bit	Content
0–2		X′410385′ NS header
3–4		CNM target ID, as specified in bytes 5–6, bits 2–3
5–6	0–1	Reserved
	2–3	CNM target ID descriptor: 00 byte 4 contains a local address for a PU or LU in a T2 node or an LSID for a PU or LU in a T1 node; byte 3 is reserved 01 bytes 3–4 contain the element address of a link, adjacent link station, PU, or LU in the origin subarea, if ENA is supported; otherwise, its network address
	4–15	Procedure related identifier (PRID) (see Note below)
7		Request-Specific Information
	0	Solicitation indicator: 0 unsolicited request 1 reply request
	1	Not-last request indicator: 0 last request in a series of related unsolicited or reply requests, e.g., last reply request in a series corresponding to a single soliciting request 1 not last request
	2–7	Request-specific type code (see below)
Note:		For reply (i.e., solicited) requests, bytes 3–6 and byte 7, bits 2–7, echo the corresponding fields in the CNM header received in the request that solicited the reply requests. For unsolicited requests, these fields—the CNM target ID descriptor, the CNM target ID, the PRID, and the request-specific information—are generated by the request sender. For unsolicited requests, the PRID field contains X′000′. The PU does not interleave requests belonging to different series of related unsolicited requests from the same target.
		Link Level 2 Test Statistics
7	0	Solicitation indicator (see above)
	1	Not-last request indicator (see above)
	2–7	Type code: 000001; the CNM target ID specifies an adjacent link station attached to a T4|5 node *Note:* When the attached adjacent link station is in a T1|2 node, the PU CNM ID is used as the adjacent link station CNM ID.)
8		Reserved
9–10		Number of DLC link test frames transmitted
11–12		Number of DLC link test frames received with or without link errors
13–14		Number of DLC link test frames received without link errors
15–16		Reason for test termination: X′0000′ test completed without error X′0001′ test completed with error—see bytes 9–14 X′0002′ test ended because of link inoperative condition X′0003′ test initialization failure; bytes 9–14 contain 0′s

RECTRD (RECORD TRACE DATA)

PU T4|5→SSCP, Norm; FMD NS(ma)

RECTRD returns data collected during a trace of the specified resource (or hierarchy of resources for a generalized PIU trace).

RECTRD (RECORD TRACE DATA)

Byte	Bit	Content
0–2		X′010383′ NS header
3–4		Element address of the resource associated with the trace, if ENA is supported; otherwise, its network address *Note:* If generalized PIU trace data is included (byte 5, bit 1 set to 1), bytes 3–4 contain the address of the PU sending RECTRD.
5		Flags and trace data format
	0	Transmission group trace: 0 no transmission group trace data included 1 transmission group trace data included
	1	Generalized PIU trace: 0 no generalized PIU trace data included 1 generalized PIU trace data included
	2–3	Reserved
	4–7	Format of bytes 9 – n: X′0′ bytes 11 – n contain intermixed status entries (length in bytes 9–10) and data entries (length of each data entry is equal to the length of a status entry plus the value of the length field in the data entry) *Note:* This format is used only when byte 5, bit 1 is set to 1 (i.e., when generalized PIU trace data is included). X′1′ retired X′3′ retired X′5′ bytes 11 – n contain status entries (length in bytes 9–10) X′7′ bytes 11 – n contain intermixed status entries (length in bytes 9–10) and data entries (length in each data entry) X′9′ bytes 11 – n contain self-defining data vectors (see below); bytes 9–10 are reserved X′D′ bytes 11 – n contain self-defining data vectors (see below); bytes 9–10 are reserved *Note:* This format is the same as format X′9′ above, but is used only when ACTTRACE byte 5, bit 4 is set to 1 (in the ACTTRACE that activated the trace for which this RECTRD is being sent); i.e., this format is used only when scanner internal trace was selected.
6		Relative time value: the time elapsed since the ACTTRACE RU was received, in units of .1 seconds (modulo 256) *Note:* This byte is reserved if byte 5, bit 1 = 1 (generalized PIU trace data included).
7		Data count: maximum number of bytes traced in each PIU, where X′FF′ means all data was traced (no maximum); echoes byte 7 of ACTTRACE *Note:* This byte is retired if byte 5, bit 0 = 1 (transmission group trace data included) and reserved if byte 5, bit 1 = 1 (generalized PIU trace data included).

RECTRD (RECORD TRACE DATA)

Byte	Bit	Content
8		Status:
	0	0 half-duplex 1 full-duplex *Note:* This bit is reserved if byte 5, bit 1 = 1 (generalized PIU trace data included).
	1	0 receive 1 transmit *Note:* This bit is reserved if bit 0 = 0 (half-duplex) or byte 5, bit 1 = 1 (generalized PIU trace data included).
	2–3	Reserved
	4	Scanner internal trace error flag (see also bit 7 below): 0 no scanner internal trace error 1 scanner internal trace error *Note:* This bit is reserved if ACTTRACE byte 5, bit 4 = 0 (in the ACTTRACE that activated the trace for which this RECTRD is being sent); i.e., reserved unless scanner internal trace was selected.
	5	0 primary 1 secondary *Note:* This bit is reserved if byte 5, bit 1 = 1 (generalized PIU trace data included).
	6	0 not last record 1 last record
	7	Trace termination flag: *Note:* This bit is retired if bit 6 = 0 (not last record)
		For bit 4 = 0 (not a scanner internal trace error): 0 trace termination by SSCP (DACTTRACE) 1 trace termination by PU sending RECTRD (resulting from an abnormal condition in the PU)
		For bit 4 = 1 (scanner internal trace error): 0 scanner resources unavailable for scanner internal trace 1 scanner hardware error during scanner internal trace
9–10		Content as described for format value (byte 5, bits 4–7)
11 – n		Format-specific trace data: implementation defined *Note:* For trace formats X′9′ and X′D′ (byte 5, bits 4–7), the trace data consists of a series of trace data vectors. Each trace data vector has a 4-byte header followed by a variable amount of data. The data is implementation defined. The format is described below.
0 – m		Trace data vector:
0		Trace category (EBCDIC character): P parameters of command sent to scanner S command status (results) reported by scanner X data transmitted by the scanner to the link connection (for scanner trace; i.e., ACTTRACE byte 5, bit 4 = 1) or by a local link station to the scanner (for link trace; i.e., ACTTRACE byte 5, bit 7 = 1) R data received by the scanner from the link connection (for scanner trace; i.e., ACTTRACE byte 5, bit 4 = 1) or by a local link station from the scanner (for link trace; i.e., ACTTRACE byte 5, bit 7 = 1) I I/O command sent to scanner (format X′D′ only) C checkpoint: audit trail of internal commands issued by scanner (format X′D′ only)
1–2		Length of data portion *Note:* For format X′9′, length includes bytes 4 – m; for format X′D′, length includes bytes 3 – m.
3		Reserved or category-specific data

RECTRD (RECORD TRACE DATA)

Byte	Bit	Content
4 – m		Category-specific data

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

REQACTCDRM (REQUEST ACTIVATION OF CROSS-NETWORK RESOURCE MANAGER)

PU→SSCP, Exp; FMD NS(c)

REQACTCDRM prompts the receiving SSCP to issue RNAA and SETCV to set up a cross-network address transform. ACTCDRM will then be sent to activate an SSCP-SSCP session with the other-network SSCP identified in this request.

REQACTCDRM (REQUEST ACTIVATION OF CROSS-NETWORK RESOURCE MANAGER)

Byte	Bit	Content
0– 2		X′41028A′ NS Header
3– 4		Reserved
5		Format: X′01′ (only value defined)
6		Activation subfunction indicators:
	0	Transform setup requirement: 0 transform setup not required; the addresses are left over from a previous session setup request 1 RNAA required to set up a cross-network address transform
	1	VRID list setup required: 0 SETCV not required with VRID list 1 SETCV required with at least one VRID list
	2– 7	Reserved

REQACTCDRM (REQUEST ACTIVATION OF CROSS-NETWORK RESOURCE MANAGER)

Byte	Bit	Content
7 – m		Session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* The following session key is used: X′15′ Network-Qualified Address Pair: sending SSCP′s real address and target SSCP′s alias addresses, respectively, in the address space defined by the network ID
m + 1 – n		Received ACTCDRM that failed because of incomplete gateway node transform: This field contains the received TH, RH, and the complete ACTCDRM RU (including control vectors).

REQACTLU (REQUEST ACTIVATE LOGICAL UNIT)

PU T4|5→SSCP, Norm; FMD NS(c)

REQACTLU is sent from the PU to an SSCP to request that ACTLU be sent to the LU named in the RU.

REQACTLU (REQUEST ACTIVATE LOGICAL UNIT)

Byte	Bit	Content
0– 2		X′410240′ NS header
3– 4		Element address of LU to be sent ACTLU, if ENA is supported; otherwise, its network address
5 – m		Network Name of LU
5		Type: X′F3′ logical unit
6		Length, in binary, of network name
7 – m		Symbolic name in EBCDIC characters

REQACTPU (REQUEST ACTIVATE PHYSICAL UNIT)

PU→SSCP, Norm; FMD NS(c)

REQACTPU is sent from the PU to an SSCP to request that ACTPU be sent to the PU named in the corresponding FID2 Encapsulation (X′1500′) GDS variable.

REQACTPU (REQUEST ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
0– 2		X′41023E′ NS header
3– 4		Reserved

REQACTPU (REQUEST ACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
5	0– 3	Format: X′0′ Format 0: external PU X′1′ Format 1: internal PU
	4– 7	Reserved

REQCONT (REQUEST CONTACT)

PU T4|5→SSCP, PU→PUCP, Norm; FMD NS(c)

REQCONT notifies the SSCP that a connection with an adjacent secondary link station (in a T1|2 node) has been activated via a successful connect-in or connect-out procedure. A DLC-level identification exchange (XID) is required before issuing REQCONT.

REQCONT (REQUEST CONTACT)

Byte	Bit	Content
0– 2		X′010284′ NS header
3– 4		Element address of link or link station. When the element address contains a link station address, the link station must be T2.1 and the connection type must be non-switched.
5 – n		XID I-field image: the bytes received in the information field of the SDLC XID response *Note:* The XID3 carried in the RU may have any exchange state (pn, np, or na); see Chapter 3, “Exchange Identification (XID) Information Fields” for format details
n + 1 – m		One or more control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT. X′25′ Security ID Control control vector X′56′ Call Security Verification control vector X′57′ DLC Connection Data control vector X′80′ Request Contact Extension control vector

Request Contact Extension (X′80′) REQCONT Control Vector

Request Contact Extension (X′80′) REQCONT Control Vector

Byte	Bit	Content
0– 1		Vector header; Key= X′80′

Request Contact Extension (X′80′) REQCONT Control Vector

Byte	Bit	Content
2		Vector Data
	0	TG status indicator: 0 REQCONT was sent for an inactive APPN TG. 1 REQCONT was sent for an active APPN TG.
	1–7	Reserved

REQDACTPU (REQUEST DEACTIVATE PHYSICAL UNIT)

PU→SSCP, Norm; FMD NS(c)

REQDACTPU is sent from the PU to an SSCP to request that DACTPU be sent to the PU.

REQDACTPU (REQUEST DEACTIVATE PHYSICAL UNIT)

Byte	Bit	Content
0–2		X′41023F′ NS header
3–4		Reserved
5	0–3	Format: X′0′ Format 0 (only value defined)
	4–7	Reserved
		Format 0
6		Cause: X′00′ Connectivity was lost between the dependent LU requester and the downstream PU. X′01′ Dependent LU requester received REQDISCONT(normal) from its serviced PU. X′02′ Dependent LU requester received REQDISCONT(immediate) from its serviced PU.

REQDISCONT (REQUEST DISCONTACT)

PU T1|2→SSCP, Norm; FMD NS(c)

With REQDISCONT, the PU T2 requests the SSCP to start a procedure that will ultimately discontact the secondary station in the T2 node.

REQDISCONT (REQUEST DISCONTACT)

Byte	Bit	Content
0–2		X′01021B′ NS header

REQDISCONT (REQUEST DISCONTACT)

Byte	Bit	Content
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.

REQECHO (REQUEST ECHO TEST)

LU→SSCP, Norm; FMD NS(ma)

(Retired RU) REQECHO has been retired from SNA. Consult product documentation for further information and support.

REQFNA (REQUEST FREE NETWORK ADDRESS)

PU T4|5→SSCP, Norm; FMD NS(c)

REQFNA is sent from a PU T4|5 to an SSCP to request the SSCP to send FNA to the PU T4|5 in order to free all addresses for the specified LU.

REQFNA (REQUEST FREE NETWORK ADDRESS)

Byte	Bit	Content
0–2		X′410286′ NS header
3–4		Element address of LU to be deleted, if ENA is supported; otherwise, its network address
5		Reserved
6		Type of request: X′01′ request X′02′ normal X′03′ forced X′04′ cleanup

REQMS (REQUEST MAINTENANCE STATISTICS)

SSCP→PU, Norm; FMD NS(ma)

REQMS has been retired from SNA for T2 nodes.

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

Consult product documentation for further information on product support.

REQMS (REQUEST MAINTENANCE STATISTICS)

Byte	Bit	Content
0–2		X′410304′ NS header
3–4		CNM target ID, as specified in bytes 5–6, bits 2–3
5–6	0–1	Reserved
	2–3	CNM target ID descriptor: 00 byte 4 contains a local address for a PU or LU in a T2 node or an LSID for a PU or LU in a T1 node; byte 3 is reserved 01 bytes 3–4 contain the element address of a link, adjacent link station, PU, or LU in the destination subarea, if ENA is supported; otherwise, its network address
	4–15	Procedure related identifier (PRID): a CNM application program generated value for CNM application program correlation, or an SSCP generated value for SSCP routing
7		Request-Specific Information
	0	Reset indicator (or reserved, as shown below for each Type code): 0 do not reset data when RECFMS is sent in reply 1 reset data when RECFMS is sent in reply
	1	Reserved
	2–7	Request-specific type code (see below)
Note:		For reply (i.e., solicited) requests, bytes 3–6 and byte 7, bits 2–7, echo the corresponding fields in the CNM header received in the request that solicited the reply requests.
7		SDLC Test Command/Response Statistics
	0	Reset indicator
	1	Reserved
	2–7	Type code: 000001; the CNM target ID identifies a PU T1|2
7		Summary Error Data
	0	Reset indicator
	1	Reserved
	2–7	Type code: 000010; the CNM target ID identifies a PU
7		Communication Adapter Data
	0	Reset indicator
	1	Reserved
	2–7	Type code: 000011; the CNM target ID identifies a PU T1|2
7 – n		PU- or LU-Dependent Data
7	0	Reset indicator
	1	Reserved
	2–7	Type code: 000100; the CNM target ID identifies a PU|LU

REQMS (REQUEST MAINTENANCE STATISTICS)

Byte	Bit	Content
8 – n		PU- or LU-dependent request parameters: implementation-dependent information (See CNM application product specifications for details).
7		Engineering Change Levels
	0– 1	Reserved
	2– 7	Type code: 000101; the CNM target ID identifies a PU
7– 8		Link Connection Subsystem Data (retired: supported only for PUs not at the current level of SNA)
7	0	Reset indicator
	1	Reserved
	2– 7	Type code: 000110; the CNM target ID identifies an adjacent link station in the destination subarea
8		Data selection requested: X′02′ link status command sequence X′03′ remote DTE interface status X′04′ remote modem self test

RNAA (REQUEST NETWORK ADDRESS ASSIGNMENT)

SSCP→PU T4|5, Norm; FMD NS(c)

RNAA requests the PU to assign addresses:

- To an adjacent link station, as identified in the RNAA request by a link element address and secondary link station link-level address
- To a dependent LU, where the LU is identified in the RNAA by an adjacent link station address and the LU local address
- To a subarea LU that supports parallel sessions; in order to assign an additional element address, the LU is identified in the RNAA request by the LU element address used for the SSCP-LU session
- As alias addresses for a cross-network SSCP-SSCP or LU-LU session, where the name pair and session characteristics are identified in the RNAA request
- To an LU to be used as a PLU address for an independent LU or to be used as an SLU address for an independent or dependent LU, where the LU is identified by an ALS address, an LU name, and an optional LU address

If ENA is not supported on this SSCP-to-PU T4|5 session, the entire network address is in each Element Address field throughout this RU.

RNAA (REQUEST NETWORK ADDRESS ASSIGNMENT)

Byte	Bit	Content
0– 2		X′410210′ NS header
3– 4		Element address of target link, adjacent link station, LU, or PU

RNAA (REQUEST NETWORK ADDRESS ASSIGNMENT)

Byte	Bit	Content
5	0–3	Address type (reserved when assignment type, bits 4–7, is set to X′0′ or X′3′): X′0′ pre-ENA compatible address required X′1′ ENA compatible address preferred X′2′ pre-ENA compatible address preferred
	4–7	Assignment type: X′0′ request is for an element address assignment of the adjacent link station associated with the target link (address type, bits 0–3, reserved for this assignment type) X′1′ request is for element address assignment of LUs associated with the target adjacent link station X′2′ request is for an additional element address assignment for the target subarea LU; bytes 3–4 contain the element address used in the SSCP-LU session X′3′ request is for cross-network address transform (address type, bits 0–3, reserved for this assignment type) X′4′ request is for an element address assignment for the target independent PLU or dependent or independent SLU X′5′ request is for an element address assignment of an adjacent link station associated with the target link
For Assignment Type X′0′		
6		Number of addresses to be defined, set to 1
7		Reserved
8		DLC header link station address associated with the adjacent link station for which an element address is requested
For Assignment Type X′1′		
6		Number of addresses to be defined, set to 1 or more
7		Reserved
8		Local address of a BF.LU for which an element address is requested, where the local address has either the one-byte format of FID2 or the six-bit local address format of FID3 (in which case, bits 0–1 of byte 8 are reserved)
9 – n		Any additional two-byte entries in the same format as bytes 7–8
For Assignment Type X′2′		
6		Retired, set to 1
7–8		Reserved
For Assignment Type X′3′		
6		Retired, set to 2

RNAA (REQUEST NETWORK ADDRESS ASSIGNMENT)

Byte	Bit	Content
7		Session characteristics *Note:* For an SSCP-SSCP session, origin-NAU and destination-NAU refer to the SSCP that sent RNAA and to the target SSCP, respectively. The parallel session indicators specify that the SSCPs do *not* have the capability to support parallel SSCP-SSCP sessions (byte 7, bit 0 and bit 1 = 0).
	0	Parallel session capability of the adjacent SSCP on the origin-NAU side of the PU: 0 SSCP does not have parallel session capability 1 SSCP does have parallel session capability
	1	Parallel session capability of the adjacent SSCP on the destination-NAU side of the PU: 0 SSCP does not have parallel session capability 1 SSCP does have parallel session capability
	2	Primary/secondary nature of OLU (reserved for session type = SSCP-SSCP, i.e., bit 4 = 1): 0 OLU=PLU 1 OLU=SLU
	3	Retired
	4	Session type: 0 LU-LU session 1 SSCP-SSCP session
	5	ENA capability of adjacent SSCP on origin NAU side of PU: 0 must be pre-ENA compatible 1 may be ENA compatible
	6	ENA capability of adjacent SSCP on destination NAU side of PU: 0 must be pre-ENA compatible 1 may be ENA compatible
	7	Reserved
8		Reserved
9–14		Origin-NAU's (real) address as known in the network adjacent to the PU and on the origin-NAU side of the PU
15–22		Network ID of the network adjacent to the PU, on the origin-NAU side of the PU
23–30		Network ID of network adjacent to the PU, on the destination-NAU side of the PU
31–38		Network ID of the origin-NAU's network
39		Length of origin-NAU name
40 – m		Origin-NAU's (real) name
m+1 – m+8		Network ID of the destination-NAU's network
m+9		Length of the destination-NAU name
m+10 – n		Destination-NAU's (real) name
For Assignment Type X'4'		
6		Reserved

RNAA (REQUEST NETWORK ADDRESS ASSIGNMENT)

Byte	Bit	Content
7		Indicators:
	0	Reserved
	1	LU role indicator: 0 The requested address is for an SLU role. 1 The requested address is for a PLU role.
	2	Authorized LU indicator: 0 the LU requires system definition to receive network services 1 the LU does not require system definition to receive network services, as it is authorized for automatic receipt of network services
	3	Reserved
	4–7	Dynamic reconfiguration types: X′0′ Add operation X′1′ Move operation: The request is for an address for an existing LU that is to be moved to a different PU. Bytes 8–9 specify the current location of the LU to be moved. X′2′ – X′F′ reserved
8–9		Element address of LU: When an Add is done for an independent LU, the first request contains a 0 in this field; subsequent Add requests contain a previously assigned address. When an Add is done for a dependent LU, the field is reserved. For Move operations, the field carries the current address of the LU that is the target of the operation.
10		Length, in binary, of Local Address field
11 – m		Local address *Note:* For independent LUs, the local address is 0; for dependent LUs, the local address is nonzero.
m + 1		Length, in binary, of the LU Name field (values 1 to 17 are valid)
m + 2 – n		LU name
n + 1 – p		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′30′ Assign LU Characteristics control vector
For Assignment Type X′5′		
6		Reserved
7		Indicators:
	0–3	Reserved
	4–7	Dynamic reconfiguration types: X′0′ Add operation X′1′ Move operation: The request is for an address for an existing adjacent link station (ALS) that is to be moved to a different link. Bytes 8–9 specify the current location of the ALS to be moved. X′2′ – X′F′ reserved
8–9		Element address of ALS. For Move operations, this field carries the current address of the ALS that is the target of the operation; otherwise, this field is reserved.
10		Length, in binary, of DLC Header Link Station Address field
11 – m		DLC header link station address associated with the adjacent link station for which an element address is requested
m + 1		Length, in binary, of the ALS Name field (values 1 to 8 are valid)

RNAA (REQUEST NETWORK ADDRESS ASSIGNMENT)

Byte	Bit	Content
m+2 – n		ALS name
n+1 – p		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X'41' Station Parameters control vector X'43' Extended SDLC Station control vector

ROUTE-INOP (ROUTE INOPERATIVE)

PU→SSCP, Norm; FMD NS(c)

ROUTE-INOP notifies the CP when either a virtual or an explicit route has become inoperative as the result of a transmission group having become inoperative somewhere in the network. This RU is retired for route dynamics.

ROUTE-INOP (ROUTE INOPERATIVE)

Byte	Bit	Content
0– 2		X'410289' NS header *Note:* Format 1 follows
3		Format: X'01' FID4 processing
4		Reason code: X'01' unexpected routing interruption over a transmission group; e.g., the last active link in a TG has failed X'02' controlled routing interruption, such as the result of DISCONTACT
5– 8		Subarea address of the PU that originated the NC-ER-INOP
9– 12		Subarea address on other end of the transmission group that had the routing interruption
13– 16		Subarea address at the route origin *Note:* This is the subarea address of the sender in the address space as defined by the network ID contained in bytes 18– 25.
17		TGN of the transmission group that had the routing interruption
18– 25		Network ID of the subnetwork in which this inoperative report applies *Note:* If the Network ID field contains all space (X'40') characters, the reported route is in the receiver's own subnetwork
26		Number of Route fields that follow
27– 42		Route Field
27– 30		Subarea address for which routing has been interrupted
31– 32		ER mask: a bit is *on* for each ER to the destination subarea identified in bytes 27– 30 for which routing has been interrupted (Bit n corresponds to ERN n)

ROUTE-INOP (ROUTE INOPERATIVE)

Byte	Bit	Content
33–34		VR mask: a bit is *on* for each VR to the destination subarea identified in bytes 27–30 for which routing has been interrupted (Bit n corresponds to VRN n)
35–42		VR-to-ER mapping list: each 4-bit field (16 fields in all) corresponds to a VR number and contains the ERN to which that VR number is assigned (If field 0 contains B′0010′, VR0 assigned to ER 2 has failed.)
43 – m		Additional Route fields in the same format as bytes 27–42

ROUTE SETUP

CP→CP, Exp; NC

ROUTE SETUP is used between adjacent HPR CPs when at least one of them does not support the Control Flows over RTP (1402) option set; it carries the Route Setup GDS variable and is preceded by a FID2 TH set to FID2, BBIU, EBIU, ODAI=0, EFI=1, DAF′=0, OAF′=0, and SNF=0 (i.e., TH=X′2D0000000000′); and an RH set to RQ, NC, FI=1, SDI=0, BC, EC, DR1=DR2=ECI=0 (RQN or no-response requested), with all other bits 0 (i.e., RH=X′2B0000′).

Route Setup

Byte	Bit	Content
0		X′10′ request code
1 – n		Route Setup GDS variable (X′12CE′)

ROUTE-TEST (ROUTE TEST)

SSCP→PU T4|5, Norm; FMD NS(ma)

ROUTE-TEST requests the PC_ROUTE_MGR component of PU.SVC_MGR to return the status (for example, active, operative, not defined), as known in the control blocks in the node, of various explicit and/or virtual routes.

ROUTE-TEST (ROUTE TEST)

Byte	Bit	Content
0–2		X′410307′ NS header
3–4		Element address of PU originating the test (as known in the sender′s subnetwork), if ENA is supported; otherwise, its network address
5		Format: X′01′ (only value defined)

ROUTE-TEST (ROUTE TEST)

Byte	Bit	Content
6		Test code: X′01′ test regardless of the states of ERs X′02′ test each ER that is not inoperative X′03′ test each ER that is inoperative X′04′ do not test the ER; respond with the current ER state (See RSP(ROUTE-TEST))
7		Choice of routes to be tested: X′01′ test the ERs corresponding to the ERNs specified in bytes 15–16 and also report the status of the VRs supported by these ERs X′02′ test the VRs corresponding to the VRNs specified in bytes 15–16; byte 6 applies to the underlying ERs for the VRs; these ERs are also tested X′03′ test the ERs corresponding to the defined TG for the ERNs specified in bytes 15–16 and also report the status of the VRs supported by these ERs
8	0–5	Reserved
	6–7	Transmission priority for NC-ER-TEST (reserved if the value of byte 6 is X′04′): 00 low priority 01 medium priority 10 high priority
9	0	Reserved
	1	Congestion data collection indicator: 0 do not collect congestion data 1 collect congestion data; VR Congestion (X′3B′) control vectors and ER Congestion (X′20′) control vectors are to be appended to RSP(ROUTE-TEST)
	2	VR data collection indicator: 0 return RSP(ROUTE-TEST) Format 1 1 return RSP(ROUTE-TEST) Format 2
	3–7	Reserved
10		Maximum expected ER length of any ER being tested (reserved if the value of byte 6 is X′04′)
11–14		Subarea address of destination PU for the NC-ER-TEST request
15–16		A bit is *on* if the corresponding ERN or VRN (depending on the route type specified in byte 7) is to be tested (Bit 0 corresponds to ERN or VRN 0, bit 1 to ERN or VRN 1, and so forth.)
17–26		Request correlation field: an implementation-defined value that is returned in ER-TESTED for correlation of reply to request (reserved if the value of byte 6 is X′04′)
27–34		Network ID of the subnetwork wherein the route to be tested resides *Note:* If the Network ID field contains all space (X′40′) characters, the route to be tested is in the sender's own network.

RPO (REMOTE POWER OFF)

SSCP→PU T4|5, Norm; FMD NS(c)

RPO causes the receiving PU T4|5 to initiate a DLC-level power-off sequence to the T4 node specified by the adjacent link station address conveyed in the request. The node being powered off does not need to have an active SSCP-PU half-session nor be contacted.

RPO (REMOTE POWER OFF)

Byte	Bit	Content
0– 2		X′010209′ NS header
3– 4		Element address of adjacent link station associated with the node to be powered off, if ENA is supported; otherwise, its network address

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

SESSEND (SESSION ENDED)

LU→SSCP, Norm; FMD NS(s)

SESSEND is sent, with no-response requested, by the LU (or boundary function on behalf of the LU in a peripheral node) to notify the SSCP that the session between the specified LUs has been successfully deactivated.

SESSEND (SESSION ENDED)

Byte	Bit	Content
0–2		X′810688′ NS header

SESSEND (SESSION ENDED)

Byte	Bit	Content
3	0– 3	Format: X′0′ Format 0 (supported only for nodes that are not at the current level of SNA) X′2′ Format 2 X′3′ Format 3
	4– 7	Reserved
		Format 0
4– 8		Session key, as described in the section “Session Keys” in Chapter 9, “Common Fields.” *Note:* The following session key is used; it is parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′07′ network address pair: PLU and SLU, respectively (only value defined)
		Format 2
4		Cause: indicates the reason for the deactivation of the LU-LU session (see UNBIND for values)
5		Retired
6 – n		Session key, as described in the section “Session Keys” in Chapter 9, “Common Fields.” *Note:* One of the following session keys is used; it is parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′07′ Network Address Pair (retired): PLU and SLU, respectively X′15′ Network-Qualified Address Pair: PLU and SLU, respectively
n + 1 – p		Control vectors, as described in the section “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′35′ Extended Sense Data control vector X′60′ Fully Qualified PCID control vector
		Format 3
4		Cause, indicating the reason for the deactivation of the LU-LU session (see UNBIND for values)
5		Retired
6 – n		Control vectors, as described in the section “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′35′ Extended Sense Data control vector X′60′ Fully Qualified PCID control vector

SESSST (SESSION STARTED)

LU→SSCP, Norm; FMD NS(s)

SESSST is sent, with no response requested, to notify the SSCP that the session between the specified LUs has been successfully activated.

SESSST (SESSION STARTED)

Byte	Bit	Content
0– 2		X′810686′ NS header
3		Format: X′00′ Format 0 (retired): no control vectors present X′01′ Format 1: control vectors present in bytes m+1 – n X′03′ Format 3: control vectors present in bytes 4 – n Formats 0 & 1
4 – m		Session key, as described in the section “Session Keys” in Chapter 9, “Common Fields.” *Note:* One of the following session keys is used; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′07′ Network Address Pair (retired): PLU and SLU, respectively X′15′ Network-Qualified Address Pair: PLU and SLU, respectively *Note:* End of Format 0; Format 1 continues below
m + 1 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′1E′ VR-ER Mapping Data control vector X′23′ Local-Form Session Identifier control vector X′28′ Related Session Identifier control vector X′2B′ Route Selection control vector X′60′ Fully Qualified PCID control vector X′68′ XRF/Session Cryptography control vector (present when session cryptography may be required for an XRF backup session with this SLU) *Note:* End of Format 1; Format 3 continues below Format 3
4 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′2A′ Session Information control vector X′2B′ Route Selection control vector (included by the DLUR as copied from the corresponding BIND) X′31′ BIND Image control vector

SETCV (SET CONTROL VECTOR)

SSCP→PU T4|5, Norm; FMD NS(c)

SETCV sets control vectors that are maintained by the PU receiving the request and that are associated with the network address specified in the RU.

SETCV (SET CONTROL VECTOR)

Byte	Bit	Content
0– 2		X′010211′ NS header
3– 4		Element address of resource (see key/resource table in the Note under bytes 5 – n) to which control vectors apply, if ENA is supported; otherwise, its network address *Note:* For control vectors X′15′, X′16′, X′1A′, X′1B′, this field contains the PU address, and the X′15′ control vector identifies the cross-network session to which the data contained in these control vectors applies.

SETCV (SET CONTROL VECTOR)

Byte	Bit	Content
5 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X'01' Date-Time control vector X'02' Subarea Routing control vector X'03' SDLC Secondary Station control vector X'04' LU control vector X'05' Channel control vector X'15' Network-Qualified Address Pair control vector X'16' Names Substitution control vector X'1A' NAU Address control vector X'1B' VRID List control vector X'38' Short-Hold Mode Emulation control vector X'41' Station parameters control vector X'42' Dynamic Path Update Data control vector X'43' Extended SDLC Station control vector X'56' Call Security Verification control vector X'60' Fully Qualified PCID control vector X'80' Frame-Relay Switching Equipment SETCV control vector X'81' Query Command control vector (set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.) X'82' Command control vector (set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.) *Note:* The following table shows the relationship between the control vectors included and the resource identified in bytes 3 and 4.

Key	Resource
X'01'	PU T4|5 receiving the SETCV
X'02'	Link to be used for routing to the subarea specified in the control vector
X'03'	SPU
X'04'	LU
X'05'	Link (S/390 channel)
X'15'	PU T4|5 receiving the SETCV
X'16'	PU T4|5 receiving the SETCV
X'1A'	PU T4|5 receiving the SETCV
X'1B'	PU T4|5 receiving the SETCV
X'38'	PU T4|5 receiving the SETCV
X'41'	Adjacent link station
X'42'	PU T4|5 receiving the SETCV
X'43'	Adjacent link station
X'56'	Adjacent link station
X'60'	PU T4|5 receiving the SETCV
X'80'	PU T4|5 receiving the SETCV
X'81'	PU T4|5 receiving the SETCV
X'82'	PU T4|5 receiving the SETCV

SETCV (SET CONTROL VECTOR)

Byte	Bit	Content
Note:		Control vector X′15′ is used to identify the session to which vectors X′16′, X′1A′, and X′1B′ apply and precedes these vectors. The addresses contained within this X′15′ control vector are as defined below: NAU 1 and NAU 2 are both addresses within the network carried in the network ID field of the vector. NAU 2 contains an alias address assigned within the gateway node receiving this vector. This order applies to the address pair session key on both the origin NAU and destination NAU side of the gateway node. Prior to the X′1A′ vector being received by the gateway node, the X′15′ control vector contains the address pair for the network adjacent to the gateway node PU on the origin-NAU side of the PU. After X′1A′ has been received in the gateway node, the X′15′ control vector may carry the session address pair on either side of the gateway node PU.

Frame-Relay Switching Equipment (X′80′) SETCV Control Vector

Frame Relay-Switching Equipment (X′80′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′80′
2 – n		Vector data
2– 3		Element address of PU associated with the first subport in an intra-FRSE PVC segment pair
4– 5		Element address of PU associated with the second subport in an intra-FRSE PVC segment pair
6– 7		Element address of PU associated with the subport that can substitute for the first subport in an intra-FRSE PVC segment, or X′0000′ if no alternate is specified
8– 9		Element address of PU associated with the subport that can substitute for the second subport in a intra-FRSE PVC segment, or X′0000′ if no alternate is specified

SETCV (SET CONTROL VECTOR)

SSCP→PU T4|5, Norm; FMD NS(ma)

SETCV sets the Intensive Mode (X′08′) control vector that is maintained by the PU receiving the request and that is associated with the network address specified in the RU.

SETCV (SET CONTROL VECTOR)

Byte	Bit	Content
0– 2		X′010311′ NS header

SETCV (SET CONTROL VECTOR)

Byte	Bit	Content
3– 4		Element address of resource to which control vector applies (as described in the Note below), if ENA is supported; otherwise, its network address
5 – n		Control vector, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to the sub-field parsing rule KL. X′08′ Intensive Mode control vector *Note:* The following table shows the relationship between the control vector included and the resource identified in bytes 3 and 4. **Key** **Resource** X′08′ Adjacent link station

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
1	0–1	Action code for S→P flow (related data in bytes 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
	2–3	
	4–7	Reserved
2–3		Secondary-to-primary sequence number data to support S→P action code
4–5		Primary-to-secondary sequence number data to support P→S action code

STSN (SET AND TEST SEQUENCE NUMBERS)

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

SWITCH (SWITCH DATA TRAFFIC)

PLU→SLU, Exp; SC

SWITCH is sent by the PLU to the SLU to change the (XRF) state of their LU-LU session from XRF-backup to XRF-active.

SWITCH (SWITCH DATA TRAFFIC)

Byte	Bit	Content
0		X'33' request code
1	0–7	Type: X'22' Switch this session from XRF-backup to XRF-active (only value defined)

TERM-OTHER (TERMINATE-OTHER)

TLU→SSCP, Norm; FMD NS(s)

TERM-OTHER from the TLU requests that the SSCP assist in terminating one or more sessions between the two LUs named in the RU. The requester may be a third-party LU or one of the two named LUs. This RU is not used by LU 6.2, although it can be used by a third-party LU for LU 6.2.

TERM-OTHER (TERMINATE-OTHER)

Byte	Bit	Content
0–2		X'810682' NS header
3	0–3	Format: 0001 Format 1 (only value defined)
	4–7	Reserved
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 10 the request applies to queued sessions only 11 the request applies to pending-active and queued sessions only

TERM-OTHER (TERMINATE-OTHER)

Byte	Bit	Content
	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 LU1; another session initiation request will be sent for LU1 1 send DACTLU to LU1 when appropriate; no further session initiation request will be sent (from this sender) for LU1
	4	0 do not send DACTLU to LU2; another session initiation request will be sent for LU2 1 send DACTLU to LU2 when appropriate; no further session initiation request will be sent (from this sender) for LU2
	5–6	Session selection (reserved for all session keys except X′06′ and X′1C′): 00 select sessions for which LU1 is PLU 01 select sessions for which LU2 is PLU 10 select sessions regardless of whether LU is PLU or SLU 11 reserved
	7	0 orderly or forced (see byte 4, bit 2) 1 cleanup
5		Reason:
	0	0 network user 1 network manager
	1	0 normal termination 1 abnormal termination
	2–7	Reserved
6		NOTIFY specifications:
	0–5	Reserved
	6	NOTIFY(X′03′) condition: 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		Type extension (see byte 4): X′01′ terminate specified session and perform notification based on the NOTIFY specification (byte 6) X′02′ retired X′03′ retired
8 – n		Session key, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* One of the following session keys is used: X′05′ PCID X′06′ Network Name Pair or Uninterpreted Name Pair session key *Note:* If the length of one of the names (LU1 or LU2, but not both) is 0, then all sessions for the named LU, as specified by the Type byte, are terminated as a result of this TERM-OTHER request. X′07′ Network Address Pair session key (retired): PLU and SLU, respectively X′0A′ URC session key *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. X′15′ Network-Qualified Address Pair session key: PLU and SLU, respectively X′1C′ Network-Qualified Name Pair session key (LU-LU session) *Note:* Either one, but not both, of the X′0E′ control vectors embedded in the X′1C′ session key may be null. If this is the case, the sessions specified by the Type byte for the named LU are terminated as a result of this TERM-OTHER request.
n + 1 – n + 2		Retired

TERM-OTHER (TERMINATE-OTHER)

Byte	Bit	Content
n + 3 – p		User Request Correlation (URC) Field
n + 3		Length, in binary, of the URC *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
p + 1 – q		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to the subfield parsing rule KL. X′35′ Extended Sense Data control vector X′60′ Fully Qualified PCID control vector

TERM-SELF Format 0 (TERMINATE-SELF)

TLU→SSCP, Norm; FMD NS(s)

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

TERM-SELF Format 0 (TERMINATE-SELF)

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

TERM-SELF Format 0 (TERMINATE-SELF)

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

TERM-SELF Format 1 (TERMINATE-SELF)

TLU→SSCP, Norm; FMD NS(s)

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

TERM-SELF Format 1 (TERMINATE-SELF)

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

TERM-SELF Format 1 (TERMINATE-SELF)

Byte	Bit	Content
		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		One of the following session keys, as described in the “Session Keys” in Chapter 9, “Common Fields”: X′0A′ URC session key *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. X′01′ Network or Uninterpreted Name session key X′07′ Network Address Pair session key X′15′ Network-Qualified Address Pair session key
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
p + 1 – s		Control vectors, as described in the section “Control Vectors” in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′0E′ Network Name control vector (present if needed to specify the network-qualified PLU name)

TESTMODE (TEST MODE)

SSCP→PU T4|5, Norm; FMD NS(ma)

TESTMODE requests the management services associated with the PU to manage a test procedure. The test procedure begins with the TESTMODE request that initiates a test and ends when the test results and status are returned in a RECTR reply request corresponding to the initial TESTMODE request.

TESTMODE (TEST MODE)

Byte	Bit	Content
0– 2		X′410305′ NS header

TESTMODE (TEST MODE)

Byte	Bit	Content
3–4		CNM target ID, as specified in bytes 5–6, bits 2–3
5–6	0–1	Reserved
	2–3	CNM target ID descriptor: 00 byte 4 contains a local address for a PU or LU in a T2 node or an LSID for a PU or LU in a T1 node; byte 3 is reserved 01 bytes 3–4 contain the element address of a link, adjacent link station, PU, or LU in the destination subarea, if ENA is supported; otherwise, its network address
	4–15	Procedure related identifier (PRID): a CNM application program generated value for CNM application program correlation, or an SSCP generated value for SSCP routing
7		Request-Specific Information
	0–1	Reserved
	2–7	Request-specific type code (see below)
Note:		For reply (i.e., solicited) requests, bytes 3–6 and byte 7, bits 2–7, echo the corresponding fields in the CNM header received in the request that solicited the reply requests.
7 – n		Link Level 2 Test Statistics
7	0	Enhanced address management indicator: 0 sender does not support enhanced address management 1 sender does support enhanced address management
	1	Static/dynamic address indicator (reserved if byte 7, bit 0 = 0): 0 sender considers the LU address to be static 1 sender considers the LU address to be dynamic
	2–7	Type code: 000001; the CNM target ID specifies an adjacent link station attached to a T4|5 node (*Note*: When the attached adjacent link station is in a T1|2 node, the PU CNM ID is used as the adjacent link station CNM ID.)
8		Reserved
9–10		Test initiation/termination code: X′0000′ (=n1) terminate an ongoing link test previously initiated X′FFFF′ (=n2) initiate a link test and run it continuously n = ¬(n1|n2) initiate a link test and transmit *n* test frames
11–12		For point-to-point links, this field is reserved; for multipoint links, this field specifies the number of test frame transmissions to be sent each time the secondary link station is serviced, e.g., in SDLC the time interval during which frames are being sent and received from a single secondary link station without another secondary link station on the link being polled or being sent frames
13 – n		Data to be sent in the data field of the link test frame

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′12′ XRF-backup hierarchical reset: the identified XRF-backup LU-LU session is being deactivated because the related XRF-active session terminated normally. The sending LU is resetting its half-session before receiving the response from the partner LU. X′13′ XRF-active hierarchical reset: the identified XRF-active LU-LU session is being deactivated because the related XRF-backup session performed a take-over of this session (via SWITCH). The sending LU is resetting its half-session before receiving the response from the partner LU. 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 2–5 are included only for Type = X′FE′ and bytes 6 – n are always omitted.

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′, this sense data value is the same as that in bytes 2-5 of the following Extended Sense Data control vector (if included); for all other UNBIND types, this field (bytes 2-5 of UNBIND) is reserved.
6 – n	Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.”

UNBIND (UNBIND SESSION)

Byte	Bit	Content
		Note: The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields"). X′35′ Extended Sense Data control vector (present if and only if the Fully Qualified PCID [X′60′] control vector is also present) X′60′ Fully Qualified PCID control vector (present on session stages that were established with extended BIND)
Note:		An UNBIND is sent instead of a -RSP(BIND) as a reply to BIND (to reject the BIND) only if the BIND is extended and no errors limit recognition of the BIND as extended.

UNBINDF (UNBIND FAILURE)

PLU→SSCP, Norm; FMD NS(s)

UNBINDF is sent, with no-response requested, by the PLU to notify the SSCP that the attempt to deactivate the session between the specified LUs has failed (for example, because of a path failure).

UNBINDF (UNBIND FAILURE)

Byte	Bit	Content
0– 2		X′810687′ NS header
3– 6		Sense data
7		Reason (a bit is set to 1 if the indicated error occurred):
	0	Reserved
	1	UNBIND error in reaching SLU
	2	takedown reject at PLU
	3– 7	Reserved
8 – m		Session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* One of the following session keys is used: X′07′ Network Address Pair session key (retired): PLU and SLU, respectively X′15′ Network-Qualified Address Pair session key: PLU and SLU, respectively
m + 1 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′35′ Extended Sense Data control vector X′60′ Fully Qualified PCID control vector

VR-INOP (VIRTUAL ROUTE INOPERATIVE)

PU T4|5→SSCP, PU T4→PUCP, Norm; FMD NS(c)

(Retired RU) VR-INOP has been retired from SNA. Consult product documentation for further information and support.

Introduction to Response Units

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

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

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

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

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

Some positive response units return the request code or NS header followed by additional data. “Positive Response Units with Extended Formats” on page 6-168 contains detailed formats of these response units, arranged in alphabetical order. Each format description begins with the following heading:

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

Positive Response Units with Extended Formats

RSP(ACTCDRM)

SSCP→SSCP, Exp; SC

RSP(ACTCDRM)

Byte	Bit	Content
0		X′14′ request code
1	0– 3	Format: X′0′ (only value defined)
	4– 7	Type activation performed: X′1′ cold X′2′ ERP
2		FM profile (see Chapter 7, “Profiles”)
3		TS profile (see Chapter 7, “Profiles”)
4– 1 1		Contents ID: 8-character EBCDIC symbolic name that represents implementation and installation dependent information about the SSCP issuing the response to ACTCDRM; eight space (X′40′) characters is the value used if no information is to be conveyed (This field could be used to provide a check for a functional and configurational match between the SSCPs.)
12– 1 7		SSCP ID: a 6-byte field that includes the ID of the SSCP issuing the ACTCDRM response; the first four bits specify the format for the remaining bits:
	0– 3	0000 Format 0 (only value defined)
	4– 7	Physical unit type of the node containing the SSCP
	8– 4 7	Implementation and installation dependent binary identification
18		TS Usage
	0– 1	Reserved
	2– 7	Secondary CPMGR receive window size (0 means no pacing of requests flowing to the secondary)
19 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL. X′06′ CDRM control vector X′09′ Activation Request/Response Sequence Identifier control vector X′13′ Gateway Support Capabilities control vector X′18′ SSCP Name control vector X′FE′ one or more control vector keys not recognized in the corresponding request

RSP(ACTLINK)

PU T4|5→SSCP, PU→PUCP, Norm; FMD NS(c)

RSP(ACTLINK)

Byte	Bit	Content
0– 2		X′01020A′ NS header
3– 4		Element address of link, if ENA is supported; otherwise, its network address
5 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′0F′ Link Capabilities and Status control vector

RSP(ACTLU)

LU→SSCP, Exp; SC

RSP(ACTLU)

Byte	Bit	Content
0		X′0D′ request code
1		Type of activation selected: X′01′ cold (retired) X′02′ ERP
2	0– 3	FM profile: X′0′ FM Profile 0 X′6′ FM Profile 6 *Note:* This field contains the same value as the FM profile field received in the ACTLU request except in the following case. If the request specified FM profile 0, the T4|5 LU may respond either FM profile 0 or FM profile 6.
	4– 7	TS profile: same as the corresponding request

RSP(ACTLU)

Byte	Bit	Content
3 – m		Control vectors, as described in "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields"). X′00′ SSCP-LU Session Capabilities control vector (present to override the defaults of a 2-byte response, in which case always first) X′0C′ LU-LU Session Services Capabilities control vector (present to override the defaults of a 2-byte response, in which case always second) X′64′ TCP/IP Information control vector (optionally present following the LU-LU Session Services Capabilities control vector if authorized by the SSCP to convey TCP/IP information associated with the TCP-connected client) *Note 1:* SSCP authorization is not required when the sender is a DLUR. *Note 2:* The TCP/IP Information control vector should be included only if the LU-LU Session Services Capabilities (X′0C′) control vector indicates that secondary LU capability is enabled. *Note:* The following control vectors are appended by the T4 boundary node. X′28′ Related Session Identifier control vector (present when the LU has an XRF backup session) X′2A′ Session Information control vector (present when the LU already has an active session; may appear more than once for LUs that support XRF) X′68′ XRF/Session Cryptography control vector (present when the LU has an active XRF session and is using session cryptography) *Note:* The following control vector is appended by the dependent LU requester. X′2A′ Session Information control vector (present when the LU already has an active session)

A two-byte response may be sent; 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|PUCP, Exp; SC

RSP(ACTPU)

Byte	Bit	Content
0		X′11′ request code
	0– 3	Format of response: X′0′ format 0 X′1′ format 1 X′2′ format 2 (this format requires that bits 4–7 be set to X′3′) X′3′ format 3 (only for PU T4|5s)
	4– 7	Type activation selected: X′1′ cold, IPL not required X′2′ ERP X′3′ cold, IPL required
2– 9		Contents ID: 8-character EBCDIC symbolic name of the load module currently operating in the node; eight space (X′40′) characters is the default value
Note:		End of Format 0 ; Format 1 continues below.

RSP(ACTPU)

Byte	Bit	Content
10 – n		Format 1 Continues
10– 11		Reserved
12 – n		Control vectors as described in the section "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule KL (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields"). X'07' PU FMD-RU-Usage control vector X'24' IPL Load Module Request control vector
Note:		End of Format 1 ; Formats 2–3 continue below.
10 – n		Format 2 Continues
10– 17		Load module ID: an eight-character EBCDIC symbolic name of the requested IPL load module: X'4040...40' any load module will be accepted ¬ X'4040...40' identifies specific load module name
18– 19		Reserved
20 – n		Control vectors as described in the section "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors may be included (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields"). X'07' PU FMD-RU-Usage
10 – n		Format 3 Continues
10 – n		Control vectors as described in the section "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors may be included (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields"). X'09' Activation Request/Response Sequence Identifier control vector (always present) X'0B' SSCP-PU Session Capabilities control vector (always present) X'11' Load Module Correlation control vector (always present)

RSP(ADDLINK)

PU T4|5→SSCP, Norm; FMD NS(c)

RSP(ADDLINK)

Byte	Bit	Content
0– 2		X'41021E' NS header
3– 4		Element address of link, if ENA is supported; otherwise, its network address

RSP(ADDLINKSTA)

PU T4|5→SSCP, Norm; FMD NS(C)

RSP(ADDLINKSTA)

Byte	Bit	Content
0–2		X′410221′ NS header
3–4		Element address of adjacent link station, if ENA is supported; otherwise, its network address

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
2–24		Bytes 2–24 of the BIND request: for an extended or negotiable response, the negotiated values may differ; for a cryptography response, the values are the same as those received in the BIND request
25	0	Reserved
	1	Negotiated or echoed from the BIND as described above for bytes 2-24.

RSP(BIND)

Byte	Bit	Content
	2– 5	Reserved
	6– 7	Length-checked compression response — an LU that is not compression-capable will return a short RSP(BIND), echo these bits as received, or set these bits to 0′s (in any of these three events, the compression is refused): 00 no compression — returned by a compression-capable LU that is refusing compression, or by an LU that is not compression-capable, or by any LU when the PLU has indicated that compression will not be used on this session 01 no compression — returned only by an LU that is not compression-capable and is echoing the BIND 10 compression accepted — returned only by a compression-capable LU to indicate compression is to be established on the session 11 no compression — returned only by an LU that is not compression-capable and is echoing the BIND *Note:* Bits 6-7, as defined, apply also to LU types 0, 1, 2, 3, and 6.1.
26 – k		Cryptography Options (see Note 3)
	0– 3	Private and session-level cryptography options: for a nonnegotiable response, same value returned as received; for a negotiable response, a value equal to one already in use for any active session with the same PLU and mode, or a value set according to the SLU node′s definition of the specified mode (but not less restrictive than requested in the BIND)
	4– 7	Session-level cryptography options field length: same value as received 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
p + 1 (= r)		Retired: set to 0 by implementations at the current level of SNA

RSP(BIND)

Byte	Bit	Content
r+1 – s		Control vectors, as described in "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields"). X'0E' Network Name control vector: CP network name (present in extended RSP(BIND)s when neither the Fully Qualified PCID [X'60'] control vector nor the Route Selection [X'2B'] control vector contains the CP(PLU) name) X'2B' Route Selection control vector (optionally present in extended RSP(BIND)s if received on BIND) X'60' Fully Qualified PCID control vector (present if received on BIND and the SLU supports sending extended RSP(BIND)) X'62' Session Address control vector (present on positive RSP(BIND)s carried in FID5 PIUs over RTP connections) X'66' Length-Checked Compression control vector (present on extended RSP(BIND)s when the SLU received a X'66' control vector on the BIND and also requests compression)
Note 1:		On a response, if the last byte of a response without control vectors (byte 7, bit 6 = 0) is a length field and that field is 0, that byte may be dropped from the response. This applies also to byte 26 (where the count occupies only bits 4–7) if bits 0–3 are also 0 — the entire byte may be dropped if no bytes follow.
Note 2:		In negotiable or extended BIND responses, reserved fields in the BIND are set by the SLU to binary 0's in the RSP(BIND); any fields at the end of the BIND that are not recognized by the SLU are discarded and not returned in the RSP(BIND).
Note 3:		The first byte of the Cryptography Options field (byte 26) is returned on the response for a nonextended nonnegotiable BIND only when session-level cryptography was specified in the BIND. Byte 26 is always present in any extended response. It is also present in any nonextended negotiable response if not truncated as allowed in Note 1. In all cases, however, the remaining bytes of the Cryptography Options field (bytes 27 – k) are present only if session-level cryptography was specified in the BIND.
Note 4:		On a response, when the adaptive session-level pacing support bit (byte 9, bit 0) is set to 1 (adaptive session pacing supported), the window sizes (byte 8, bits 2–7; byte 9, bits 2–7; byte 12, bits 2–7 and byte 13, bits 2–7) are all set to 0.
Note 5:		An extended short response to a nonnegotiable BIND is of the following form:
0		X'31' request code
1	0–3 4–7	Format: 0000 (only value defined) 0001 nonnegotiable
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– 2 4		Reserved
25	0– 5	Reserved
	6– 7	Length-checked compression response — an LU that is not compression-capable will return a short RSP(BIND), echo these bits as received, or set these bits to 0′s. In any of these three events, the offered or mandated compression is refused (see Chapter 5, “Request/Response Headers (RHs)” for details of the compression header): 00 no compression — returned by a compression-capable LU that is refusing the offered or mandated compression, or by an LU that is not compression-capable, or by any LU when the PLU has indicated that compression will not be used on this session 01 no compression — returned only by an LU that is not compression-capable and is echoing the BIND 10 offered or mandated compression accepted — returned only by a compression-capable LU to indicate acceptance of compression on the session 11 no compression — returned only by an LU that is not compression-capable and is echoing the BIND *Note:* Bits 6-7, as defined, apply also to LU types 0, 1, 2, 3, and 6.1.
26– 3 0 (= r)		Reserved
r + 1 – s		Control vectors, as described in the section “Control Vectors” in Chapter 9, “Common Fields” *Note:* The following control vectors may be used; they are parsed according to subfield parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′2B′ Route Selection control vector (optionally present if received on the BIND) X′60′ Fully Qualified PCID control vector (always present) X′62′ Session Address control vector (present on positive RSP(BIND)s carried in FID5 PIUs over RTP connections) X′66′ Length-Checked Compression control vector (present when the SLU received a X′66′ control vector on the BIND and also requests compression)

RSP(CDINIT)

SSCP→SSCP, Norm; FMD NS(s)

RSP(CDINIT)

Byte	Bit	Content
0–2		X′818641′ NS header
3	0–3	Format: same value as received in corresponding request
Formats 0–4 continue (see Format 5 continuation below)		
	4–7	Reserved
4	0–1	Reserved
	2	Network-qualified names support indicator (defined only for non-DQ CDINIT formats): 0 A BIND for this session sent or received in the domain of the sending SSCP will not contain network-qualified LU names in bytes k+2 – m and p+2 – r. 1 A BIND for this session sent or received in the domain of the sending SSCP may contain network-qualified LU names in bytes k+2 – m and p+2 – r.
	3	SLU XRF support indicator: 0 SLU does not support XRF. 1 SLU supports XRF.
	4–7	Procedure status at SSCP receiving CDINIT: 0000 reserved 0001 initiate successful—proceed 0010 initiate successful—queued 0011 dequeued—successful
5–6		Network address of DLU for CDINIT; for CDINIT(DQ), it is the network address of the LU associated with the SSCP receiving the CDINIT(DQ) request *Note:* Retired for formats 3 and 4. (DLU address is in control vector X′1A′ for format 3.)
7		LU status for LU associated with the SSCP receiving the CDINIT request:
	0	(defined only for non-DQ CDINIT formats) 0 DLU does not support extended BIND 1 DLU supports extended BIND
	1	LU availability 0 LU is unavailable 1 LU is available
	2–3	(reserved if LU is available) 00 LU session limit exceeded 01 reserved 10 LU is not currently able to comply with the PLU/SLU specification 11 reserved
	4	0 existing SSCP to LU path 1 no existing SSCP to LU path
	5	(reserved in formats 0, 1 and 4) 0 UNBIND and SESSEND cannot be sent by the LU or by its boundary function (retired) 1 UNBIND and SESSEND will be sent by the LU or by its boundary function
	6–7	(defined only for non-DQ CDINIT formats) 00 reserved 01 LU is PLU 10 LU is SLU 11 reserved

RSP(CDINIT)

Byte	Bit	Content
8 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′2B′ Route Selection control vector X′31′ BIND image control vector X′3F′ SSCP(SLU) Capabilities control vector X′5C′ APPN Message Transport control vector X′63′ Cryptography Key Distribution control vector (present only when key distribution information is to be transferred). X′64′ TCP/IP Information control vector (present to forward SLU TCP/IP information to the SSCP(PLU) if that information was provided by the SLU) X′65′ Device Characteristics control vector X′68′ XRF/Session Cryptography control vector (present in format 4 when session cryptography is required on an XRF backup session)
End of Formats 0, 1, and 4; Formats 2 and 3 continue below		
8	0	COS origin: 0 no COS name from ILU 1 COS name from ILU
	1– 2	(reserved if byte 8, bit 0 ≠ 0) 01 SSCP(DLU) chose COS name (DLU is SLU) 10 SSCP(OLU) chose COS name (OLU is SLU)
	3– 7	Reserved
9– 1 6		COS name (if byte 8, bit 0 = 0 and bits 1– 2 ≠ 01, this field carries unpredictable values and is not used): symbolic name of class of service in EBCDIC characters *Note:* For format 3, this COS name represents the COS name as known in the network of the DLU.
17– 2 4		Mode name (if byte 8, bits 1– 2 ≠ 01, this field carries unpredictable values and is not used): an 8-byte symbolic name (implementation and installation dependent) that identifies the set of rules and protocols to be used for the session (included here for use in reactivating the (LU,LU) session, if necessary; see CINIT and SESSEND for other details)
End of Format 2; Format 3 continues below		
25 – n		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. If present, they appear in the order specified. X′1A′ NAU Address control vector: contains the DLU network address (always present) *Note:* Between gateways, the DLU address is an address recognized in the subnetwork on the OLU side of the sending gateway. Within a gateway, the DLU address is an address recognized in the network on the DLU side of the gateway node, until the SSCP with address alias responsibility is reached. This SSCP replaces the received address with an address recognized in the network on the OLU side of the gateway node. The network ID is identified in the X′19′ vector for the DLU. X′14′ Session Initiation control vector (always present) X′19′ Resource Identifier control vector for destination LU (always present) X′19′ Resource Identifier control vector for origin LU (always present) X′2B′ Route Selection control vector X′2C′ COS/TPF control vector X′2F′ LU Definition control vector (present if data is available to be sent and if the RSP(CDINIT) flows from the SSCP(SLU))

RSP(CDINIT)

Byte	Bit	Content
		X′2F′ LU Definition control vector (present only when immediately preceded by the previous X′2F′ control vector and model name or associated LU data is available)
		X′31′ BIND Image control vector
		X′3E′ Directory Entry Characteristic control vector
		X′3F′ SSCP(SLU) Capabilities control vector
		X′59′ Installation-Defined CDINIT Data control vector (present when an SSCP exit is invoked)
		X′5A′ Session Services Extension Support control vector
		X′5B′ Interchange Node Parameters control vector
		X′5C′ APPN Message Transport control vector
		X′60′ Fully Qualified PCID control vector
		X′63′ Cryptography Key Distribution control vector (present only when key distribution information is to be transferred)
		X′64′ TCP/IP Information control vector (present to forward SLU TCP/IP information to the SSCP(PLU) if that information was provided by the SLU)
		X′65′ Device Characteristics control vector
		X′66′ Length-Checked Compression control vector
		X′68′ XRF Session Cryptography control vector (present when session cryptography is required on an XRF backup session)
End of Format 3; Format 5 continues below		
4	0	1 SLU supports network-qualified names.
	1	1 SLU supports extended BIND.
	2	1 SLU is available.
	3–4	LU status (meaningful only if SLU unavailable; otherwise, reserved): 00 session limit exceeded 10 LU disabled
	5	1 SSCP at PLU end of VRTG must send CDSESSST when the session goes active.
	6	1 SSCP should not include an SLU RSCV on the CINIT/BFCINIT.
	7	Reserved
5–12		Subarea COS name (left-justified and padded on the right by X′40′ characters if necessary)
13		Uninterpreted name length
14 – m		Uninterpreted name
m – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL and appear in the order specified up through the second X′19′ control vector: X′1A′ NAU Address control vector (always present): contains the SLU network address X′2B′ Route Selection control vector (optionally present): BIND RSCV for adjacent subnet X′2C′ COS/TPF control vector (optionally present): APPN COS name for adjacent subnet X′60′ Fully Qualified PCID control vector, as obtained from the incoming CDINIT request (always present)

RSP(CDTERM)

SSCP(DLU)→SSCP(OLU), Norm; NS(s)

RSP(CDTERM)

Byte	Bit	Content
0– 2		X′818643′ NS header
3 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′5C′ APPN Message Transport control vector

RSP(CINIT)

PLU→SSCP, Norm; FMD NS(s)

RSP(CINIT)

Byte	Bit	Content
0– 2		X′810601′ NS header
3 – n		Control vectors as described in the section “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vector may be included (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”) X′FE′ control vector not recognized

RSP(DSRLST)

SSCP→SSCP, Norm; NS(s)

RSP(DSRLST)

Byte	Bit	Content
0– 2		X′818627′ NS header
3 – n		Control list entry data for list type (See the “Control Lists” discussion in Chapter 9, “Common Fields” for detailed descriptions): X′01′ LU Status List

RSP(DSRLST)

Byte	Bit	Content
n + 1 – p		Control vectors, as described in the "Control Vectors" discussion in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X'19' Resource Identifier control vector (always present): identifies the LU as known by the SSCP sending the RSP(DSRLST) X'31' Bind Image control vector X'3E' Directory Entry Characteristic control vector X'5A' Session Services Extensions Support control vector X'5B' Interchange Node Parameters control vector X'5C' APPN Message Transport control vector X'60' Fully Qualified PCID control vector X'63' Cryptography Key Distribution control vector (present only when key distribution information is to be transferred) X'65' Device Characteristics control vector

RSP(DUMPINIT)

PU T4|5→SSCP, Norm; FMD NS(c)

RSP(DUMPINIT)

Byte	Bit	Content
0– 2		X'010206' NS header
3 – n		Dump data

RSP(DUMPTEXT)

PU T4|5→SSCP, Norm; FMD NS(c)

RSP(DUMPTEXT)

Byte	Bit	Content
0– 2		X'010207' NS header
3 – n		Dump data

RSP(INIT-OTHER-CD)

SSCP→SSCP, Norm; FMD NS(s)

RSP(INIT-OTHER-CD)

Byte	Bit	Content
0– 2		X′818640′ NS header
3	0– 3	Format: same value as received in corresponding request
	4– 7	Reserved
4		Procedure status:
	0– 3	Status for SSCP(LU1): 0000 reserved 0001 initiate successful—proceed 0010 initiate successful—queued
	4– 7	Status for SSCP(LU2): 0000 reserved 0001 initiate successful—proceed 0010 initiate successful—queued
5		LU1 status:
	0	Reserved
	1	0 LU1 is unavailable 1 LU1 is available
	2– 3	(reserved if LU1 is available) 00 LU1 session limit exceeded 01 reserved 10 LU1 is not currently able to comply with the PLU/SLU specification 11 reserved
	4– 5	Reserved
	6– 7	00 reserved 01 LU1 is PLU 10 LU1 is SLU 11 reserved
6		LU2 status:
	0	Reserved
	1	0 LU2 is unavailable 1 LU2 is available
	2– 3	(reserved if LU2 is available) 00 LU2 session limit exceeded 01 reserved 10 LU2 is not currently able to comply with the PLU/SLU specification 11 reserved
	4– 5	Reserved
	6– 7	00 reserved 01 LU2 is PLU 10 LU2 is SLU 11 reserved
7 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule KL. X′5C′ APPN Message Transport control vector

RSP(RNAA)

PU T4|5→SSCP, Norm; FMD NS(c)

If ENA is not supported on this SSCP-to-PU T4|5 session, the entire network address is in each Element Address field throughout this RU.

RSP(RNAA)

Byte	Bit	Content
0–2		X′410210′ NS header
For assignment type X′0′:		
3–5		Set to same value as bytes 3–5 in RNAA request
6		Number of element addresses returned, set to 1
7–8		Element address assigned for adjacent link station
For assignment type X′1′:		
3–5		Set to same value as bytes 3–5 in RNAA request
6		Number of element addresses returned, set to 1 or more
7–8		Element address assigned for LU
9 – n		Any additional element addresses assigned (in 2-byte multiples), in the same format as bytes 7–8; the order of the element addresses returned corresponds to the order of the entries (bytes 7 – n) in the RNAA request
For assignment type X′2′:		
3–5		Set to same value as bytes 3–5 in RNAA request
6		Retired, set to 1
7–8		Element address assigned for LU
For assignment type X′3′:		
3–5		Set to same value as bytes 3–5 in RNAA request
6		Retired, set to 2
7–12		Destination-NAU alias address, applicable in the subnetwork adjacent to the PU, on the origin-NAU side of the PU
13–18		Origin-NAU alias address, applicable in the subnetwork adjacent to the PU, on the destination-NAU side of the PU
For assignment type X′4′:		
3–4		Element address depending on type of dynamic reconfiguration operation (byte 7, bits 4–7 of RNAA): • For Add (X′0′) operation: Set to same value as bytes 3–4 in RNAA request • For Move (X′1′) operation: Carries the attachment address (ALS) from which the LU was moved
5		Set to same value as byte 5 in RNAA request
6		Reserved
7–8		Element address assigned for LU

RSP(RNAA)

Byte	Bit	Content
For assignment type X′5′:		
3–4		Element address depending on type of dynamic reconfiguration operation (byte 7, bits 4–7 of RNAA): • For Add (X′0′) operation: set to same value as bytes 3–4 in RNAA request • For Move (X′1′) operation: carries the attachment address (link) from which the adjacent link station was moved
5		Set to same value as byte 5 in RNAA request
6		Reserved
7–8		Element address assigned for adjacent link station

RSP(ROUTE-TEST)

PU T4|5→SSCP, Norm; FMD NS(ma)

RSP(ROUTE-TEST)

Byte	Bit	Content
0–2		X′410307′ NS header
Format 1 follows.		
3		Format: X′01′
4		Number of Route Data fields
5–14		Route Data: information about the ERs and VRs that were tested
5		Virtual route identifier:
	0–3	VRN of the VR tested
	4–5	Reserved
	6–7	Transmission priority field of the VR tested
6		VR status: X′00′ VR is not defined X′01′ VR is in reset state X′02′ activation of the VR is pending notification of the activation of the underlying ER X′03′ an NC-ACTVR was sent to activate the VR, but no RSP(NC-ACTVR) has been received X′04′ an NC-ACTVR was received to activate the VR, but no RSP(NC-ACTVR) has been sent X′05′ an NC-DACTVR(Orderly) has been sent, but no RSP(NC-DACTVR) has been received X′06′ an NC-DACTVR(Orderly) was received, but no RSP(NC-DACTVR) has been sent X′07′ an NC-DACTVR(Forced) was received, but no RSP(NC-DACTVR) has been sent X′08′ an NC-DACTVR(Forced) was sent, but no RSP(NC-DACTVR) has been received X′09′ VR is active X′0A′ retired

RSP(ROUTE-TEST)

Byte	Bit	Content
7	0–3	Reserved
	4–7	ERN of the ER tested
8		ER status: X′00′ ER is not defined and not currently operative X′01′ ER is defined, but not currently operative X′02′ ER is defined and operative, but not currently active X′03′ an NC-ER-ACT was sent, but no NC-ER-ACT-REPLY has been received X′04′ an NC-ER-ACT was received, but no NC-ER-ACT-REPLY has been sent X′05′ an NC-ER-ACT was received and an NC-ER-ACT-REPLY was sent; an NC-ER-ACT was sent, but no NC-ER-ACT-REPLY has been received X′06′ an NC-ER-ACT was received, but no ER is defined; should the ER subsequently become defined, an NC-ER-ACT will be sent X′07′ an NC-ER-ACT was received and an NC-ER-ACT-REPLY was sent (no NC-ER-ACT has been sent from this end) X′08′ ER is active and each node on the ER supports ER-VR protocols X′09′ ER is operative, but not currently defined X′0A′ ER is active and traverses a node that does not support ER-VR protocols
9–12		Subarea address of the adjacent node through which the ER being tested flows from this node
13		Transmission group number of the TG (to the node identified in bytes 9–12) over which the ER being tested flows from this node
14		Reserved
15 – m		Any additional 10-byte entries in the same format as bytes 5–14
m+1 – m+4		Subarea address at the route origin *Note:* This is the subarea address of the sender in the address space as defined by the network ID contained in bytes m+5 – m+12
m+5 – m+12		Network ID of the subnetwork wherein the tested route resides (same as bytes 27–34 of corresponding ROUTE-TEST request)
End of Format 1; Format 2 follows.		
3		Format: X′02′
4		Reserved
5–8		Subarea address at the route origin *Note:* This is the subarea address of the sender in the address space as defined by the network ID contained in bytes 9–16.
9–16		Network ID of the subnetwork wherein the tested route resides (same as bytes 27–34 of corresponding ROUTE-TEST request)
17 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vectors may be included in RSP(ROUTE-TEST) format 2; they are parsed according to subfield parsing rule KL. X′3A′ Route Status Data control vector (One or more X′3A′ control vectors may be included for each VR and ER for which status was requested in ROUTE-TEST.) X′3B′ VR Congestion Data control vector (One X′3B′ control vector may be included for each VR for which congestion data was requested in ROUTE-TEST.) X′6A′ ER Congestion Data control vector (One X′6A′ control vector may be included.)

RSP(SETCV)

PU T4|5→SSCP, Norm; FMD NS(c)

RSP(SETCV)

Byte	Bit	Content
0– 2		X′010211′ NS header
3 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vector may be included in RSP(SETCV) (configuration services); it is parsed according to subfield parsing rule KL. X′42′ Dynamic Path Update Data control vector

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.

RSP(SWITCH)

SLU→PLU, Exp; SC

RSP(SWITCH) is sent by the boundary function to the primary LU to inform the PLU that the requested change has occurred and to provide the session state information that may be needed by the new XRF-active PLU to restart the session after SWITCH.

RSP(SWITCH)

Byte	Bit	Content
0		X′33′ request code
1 – n		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note:* The following control vector may be included; it is parsed according to subfield parsing rule KL. X′29′ Session State Data control vector

End of Chapter 6

Chapter 7. 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; those profiles that refer to SSCP options for cross-domain support are called cross-domain resource manager (CDRM) 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 ACTCDRM, ACTPU, ACTLU, BIND, and their responses in Chapter 5); the CDRM profile is specified at SSCP-SSCP session activation, via a control vector parameter carried in ACTCDRM and +RSP(ACTCDRM).

Transmission Services (TS) Profiles

This section describes the transmission services (TS) profiles and their use for sessions defined in SNA. Profile numbers not shown are reserved.

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 7-1 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) (note 1) SSCP–LU	– –
2	LU–LU	0
3	LU–LU	0, 1, 2, 3
4	LU–LU	0, 1, 6.1
5	SSCP–PU(T4|5)	–
7	LU–LU CP–CP	0, 4, 6.2, 7 –
17	SSCP–SSCP	–

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

Figure 7-1. 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 5

Profile 5 (used on SSCP-PU sessions) specifies the following session rules:

- No pacing.
- Sequence numbers are used on normal flows.
- SDT is supported.
- CLEAR, RQR, STSN, and CRV are not supported.
- Maximum RU size on the normal flow between subarea nodes is limited only by the maximum PIU length allowed on the ER in the direction of flow. (The maximum PIU length on the ER is the smallest of the maximum PIU lengths allowed on any TG on the ER.)

This profile does not require the use of the TS Usage field.

TS Profile 7

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

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

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

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

TS Profile 17

Profile 17 (used on SSCP-SSCP sessions) specifies the following session rules:

- Primary-to-secondary and secondary-to-primary normal flows are paced.
- Identifiers rather than sequence numbers are used on the normal flows.
- SDT is supported.
- STSN, CLEAR, RQR, and CRV are not supported.
- Maximum RU size on the normal flow between subarea nodes is limited only by the maximum PIU length allowed on the ER in the direction of flow. (The maximum PIU length on the ER is the smallest of the maximum PIU lengths allowed on any TG on the ER.)

The TS Usage subfield defining the options for this profile is:

- Pacing window counts

Function Management (FM) Profiles

This section describes the function management (FM) profiles and their use for sessions defined in SNA. Profile numbers not shown are reserved.

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 7-2 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) (note 1) SSCP–LU	– –
2	LU–LU	0
3	LU–LU	0, 1, 2, 3
4	LU–LU	0, 1
5	SSCP–PU(T4|5)	–
6	SSCP–LU	–
7	LU–LU	0, 4, 7
17	SSCP–SSCP	–
18	LU–LU	0, 6.1
19	LU–LU CP–CP	6.2 –

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

Figure 7-2. FM Profiles and Their Usage

FM Profile 0

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

- Primary and secondary half-sessions use immediate request mode and immediate response mode.
- Only single-RU chains allowed.
- Primary and secondary half-session chains indicate definite response. Half-session chains generated by a boundary function on behalf of the peripheral LU may indicate no-response or definite response.
- No compression.
- Primary half-session sends no DFC RUs.

- Secondary LU half-session may send LUSTAT.
- No FM headers.
- No brackets.
- No alternate code.
- Normal-flow send/receive mode is full-duplex.

FM Profile 2

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

- Secondary LU half-session uses delayed request mode.
- Secondary LU half-session uses immediate response mode.
- Only single-RU chains allowed.
- Secondary LU half-session requests indicate no-response.
- No FMH-1 SCB compression.
- Length-checked compression allowed.
- No DFC RUs.
- No FM headers.
- Secondary LU half-session is first speaker if brackets are used.
- Bracket termination rule 2 is used if brackets are used.
- Primary LU half-session will send EB.
- Secondary LU half-session will not send EB.
- Normal-flow send/receive mode is FDX.
- Primary LU half-session is responsible for recovery.

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

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

FM Profile 3

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

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

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

- Chaining use (primary and secondary)
- Request control mode selection (primary and secondary)
- Chain response protocol (primary and secondary)

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

FM Profile 4

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

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

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

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

FM Profile 5

Profile 5 (used on SSCP-PU 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 indicate no-response or definite response.
- No compression.
- No DFC RUs.
- No FM headers.
- No brackets.
- No alternate code.
- Normal-flow send/receive mode is FDX.

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.
- Secondary half-session may send LUSTAT.
- No FM headers.
- No compression.
- No brackets.
- No alternate code.
- Normal-flow send/receive mode is full-duplex.

FM Profile 7

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

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

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

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

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

FM Profile 17

Profile 17 (used on SSCP-SSCP 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 indicate definite response or no-response.
- No DFC RUs.
- No FM headers.
- No compression.
- No brackets.
- No alternate code.
- Normal-flow send/receive mode is full-duplex.

FM Profile 18

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

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

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

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

- Half-duplex flip-flop reset states

FM Profile 19

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

- Primary LU half-session and secondary LU half-session use immediate request and immediate response mode.
- Multiple RU chains allowed.
- Primary LU half-session and secondary LU half-session chains indicate definite or exception response.
- No FMH-1 SCB compression.
- Length-checked compression allowed.
- Brackets are used.
- FM headers (types 5, 7, and 12 only) are allowed.
- Conditional termination for brackets (specified by CEB) will be used—primary and secondary half-sessions may send CEB. For full-duplex conversations, two CEBs are required to end the bracket — one from the primary half-session and one from the secondary half-session.
- Normal-flow send/receive mode may be half-duplex flip-flop or full-duplex; an FM Usage field in BIND specifies which one.
  - Specifying half-duplex flip-flop indicates that the LU supports only half-duplex conversations.
  - Specifying full-duplex indicates that the LU is able to support both half-duplex and full-duplex conversations on a particular session.
- 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).

  Note:

  The reset state refers to the session and not conversations; therefore, the half-duplex flip-flop reset state protocol is valid for LUs that support both half-duplex and full-duplex conversations, in addition to LUs that support only half-duplex conversations.

- 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

  In addition, LUs that support both half-duplex and full-duplex conversations also support the EXPD DFC function.
- Alternate code permitted.
- The following combinations of RQE, RQD, CEB, and CD codings are allowed on *end-chain* requests:

Figure 7-3. RH Encodings for End-Chain Requests

For half-duplex conversations:	For full-duplex conversations:
RQE*, CD, ¬CEB RQD2, CD, ¬CEB RQD3, CD, ¬CEB RQE1, ¬CD, CEB RQD*, ¬CD, CEB RQD*, ¬CD, ¬CEB	RQE1, ¬CD, CEB RQD1, ¬CD, CEB
Note: For full-duplex conversations only, the following coding combination on a request is used to indicate that the partner transaction program issued an (MC_)FLUSH verb: ¬EC, RQE1, CD, ¬CEB.	

Cross-Domain Resource Manager (CDRM) Profiles

The CDRM profile is specified in a control vector carried in ACTCDRM and RSP(ACTCDRM) to define the cross-domain capabilities of an SSCP. CDRM Profile 0 is described here. All other profile numbers are reserved.

CDRM Profile 0

Profile 0, along with the CDRM usage fields in the control vector, specifies functional capabilities of the SSCP.

The options specified in the CDRM usage fields for Profile 0 are:

- Network name pair session key (X′06′) supported
- Network address pair session key (X′07′) supported
- PCID session key (X′05′) supported
- URC support by the SSCP (and all PLUs within its domain) in cross-domain session initiation, i.e., (1) BINDs issued from all PLUs in this domain carry URC if the INIT specified a URC, and an SLU in the other domain issued the INIT; and (2) the BIND image in CDCINITs issued from the SSCP in this domain carry URC if the INIT specified a URC, and an SLU in this domain issued the INIT

End of Chapter 7

Chapter 8. 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, Nonce Data, Random Data, Security Mechanisms Data, Security Reply 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: values 3 to 19 (X′13′) are valid
1		X′01′
2		Length of primary resource qualifier: values 0 to 8 are valid (X′00′ means no primary resource qualifier is present)
3 – m		Primary resource qualifier
m + 1		Length of secondary resource qualifier: values 0 to 8 are valid (X′00′ means no secondary resource qualifier is present)
m + 2 – n		Secondary resource qualifier

Mode Name Structured Data Subfield

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

Mode Name Structured Data Subfield

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

Session Instance Identifier Structured Data Subfield

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

Session Instance Identifier Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Session Instance Identifier subfield: values 2 to 9 are valid
1		X′03′
2 – n		Session Instance Identifier
2		Format: X′00′ retired in BIND, used in RSP(BIND) when Format X′00′ was used in BIND and PLU name ≤ SLU name; or when Format X′01′ was used in BIND, the SLU does not support extended BINDs, and PLU name ≤ SLU name X′01′ used in BIND only X′02′ used in RSP(BIND) in response to Format X′01′ in BIND when the SLU supports extended BINDs X′F0′ used in RSP(BIND) when Format X′00′ was used in BIND and PLU name > SLU name; or when Format X′01′ was used in BIND, the SLU does not support extended BINDs, and PLU name > SLU name
3 – n		Type-G symbol string identifying the session instance: a 1- to 7-byte string generated by the PLU and echoed by the SLU

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 upper-case letters A through Z; numerics 0 through 9; the first character of which is an upper-case 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 upper-case letters A through Z; numerics 0 through 9; the first character of which is an upper-case 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 LU-LU verification. When LU-LU verification is in effect, this subfield is present in both BIND and RSP(BIND). In BIND, it carries random data to be processed and returned in the Security Reply Data structured data subfield in RSP(BIND); in RSP(BIND), it carries random data to be processed and returned in FMH-12.

Random Data Structured Data Subfield

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

Security Reply Data Structured Data Subfield

The Security Reply Data subfield is present in the RSP(BIND) when LU-LU verification is in effect; i. e., when the Random Data structured data subfield was received in BIND.

Security Reply Data Structured Data Subfield

Byte	Bit	Content
0		Length, in binary, of the remainder of the Security Reply Data subfield: 9 is the only valid value
1		X′12′
2–9		When the *basic* LU-LU verification protocol is supported, this subfield contains the enciphered version of the clear random data received in BIND. When the *enhanced* LU-LU verification protocol is supported, this subfield contains the DES Message Authentication Code value of the 3-part string composed of the following parts: • The random data value received in the BIND • The random data value to be sent on the RSP(BIND) • The network-qualified SLU network name to be sent in the RSP(BIND) exclusive ORed with the random data value received in the BIND The DES Message Authentication Code algorithm is a standard cryptographic algorithm used to generate a value that can be used to verify the contents of a data field. In both cases, the installation defined LU-LU password is used as the cryptographic key for the DES algorithm.

Nonce Data Structured Data Subfield

The Nonce Data subfield contains the random data used in password substitutes. When password substitution is in effect and LU-LU verification is not in effect, this subfield is present in both BIND and RSP(BIND).

Nonce Data Structured Data Subfield

Byte	Bit	Content
0		Length of the remainder of the Nonce Data subfield: 10 is the only valid value
1		X′13′
2		Reserved
3–10		Nonce data: a type-G random value generated for use in building password substitutes

Security Mechanisms Data Structured Data Subfield

Security Mechanisms Data Structured Data Subfield

Byte	Bit	Content
0		Length (n≥5), in binary, of the remainder of the Security Mechanisms Data subfield
1		X′14′
2		Length (≥0) of the following Security Policy Information field
3 – m		Security Policy Information
3	0	Performance optimization indicator: 0 The LU does not support the use of a dedicated session for GSS-API ticket exchanges after the GSS-API context is in prot_ready_state; i.e., prot_ready_state is ignored and all GSS_S_CONTINUE_NEEDED statuses will require a line turnaround (only value defined if parallel sessions are not supported). 1 The LU supports the out-of-band performance optimization based on the prot_ready_state state of the GSS_S_CONTINUE_NEEDED status.
	1–7	Reserved
4 – m		Reserved
m+1		Length of the vector of supported extended security mechanism identifiers and mechanism policy values: an even value ≥ 2
m+2 – n		List of 2-byte supported mechanism identifiers and policy fields: The mechanism identifiers are in ascending order by value. All mechanism identifiers that the sending LU can support or accept are included. The first byte of each pair is a mechanism identifier; the second byte is a mechanism-dependent policy field. Figure 8-1 lists the allowable values.

Mechanism ID	Policy Byte	Mechanism Name	Mechanism OID[1]
X′01′	reserved	DCE Authentication	1.3.24.9.8
X′02′	reserved	reserved	reserved
X′03′	reserved	Kerberos V5	1.3.5.1.5.2
X′04′	reserved	DCE Performance Mechanism	Not yet assigned

Figure 8-1. Extended Security Mechanisms

End of Chapter 8

[1] DCE Authentication, OID 1.3.24.9.8, has the BER encoding of X′2B 18 09 08′; Kerberos V5, OID 1.3.5.1.5.2, has the BER encoding of X′2B 05 01 05 02′.

Chapter 9. Common Fields

Introduction

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

- Control vectors
- Control lists
- Session keys

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 such as control vectors, subvectors, and subfields that are carried within some enclosing structure and that have one-byte keys identifying the substructures. The terms *key* and *type* are used interchangeably here, since both terms are used in the substructures to which the following rules apply.

Partitioning of Key/Type Values

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

Category-wide Keys: Within the category of control vectors, keys in the range X′00′ to X′7F′ are unique; within the independent category of management services (MS) subvectors (described in *Systems Network Architecture Management Services Formats*), they are also unique.

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

Parsing Rules

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

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

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

Example of Common Substructure Format

Byte	Bit	Content
The general format of a control vector, for example, is shown as:		
0– 1		Vector header; Key=X′45′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		Vector Data
When the enclosing structure indicates use of parsing rule KL, the first two bytes are interpreted as:		
0		Key
1		Length (n–1), in binary, of the Vector Data field (i.e., excluding the length of the Vector Header field)
When the enclosing structure indicates use of parsing rule LT, the first two bytes are interpreted as:		
0		Length (n+1), in binary, of the control vector (i.e, including the Vector Header and Vector Data fields)
1		Type (=Key)

Some early control vectors (i.e., with key values X′00′ – X′08′, as well as X′24′) have no explicit length field; these perforce appear only in the KL parsing context.

Enclosing Rule for Substructures

In general, 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. Exceptions include session-activation RUs (e.g., ACTPU, ACTLU, BIND, and their responses) and subarea-only RUs — for which, see the descriptions of the individual RUs and their nested substructures.

Consider the Product Set ID (X′10′) control vector as an example of the general rule. Imbedded within this substructure are other substructures, specifically Product Identifier (X′11′) MS common subvectors. When the Product Set ID (X′10′) is present in XID format 3, it is parsed KL, whereas when it is present within a major vector in NMVT, it is parsed LT. In both cases, the Product Identifier (X′11′) subvectors are parsed LT.

Control Vectors

Introduction

The following table shows, by (category-wide) key value, the control vector (i.e., with Key = X′FE′ or Key < X′80′), and the message-unit structures that can carry the control vector. This section defines only the category-wide control vectors. (Note: HPR redefines control vector keys X′00′, X′03′, X′05′, and X′28′ in violation of the customary rule. The HPR-only uses of these keys are so indicated in the table below in the "Key" column. These keys have long been used with other meanings in the context of subarea networks.)

Context-sensitive control vectors (Keys X′80′ – X′FD′) are defined in-line with their enclosing structures (e.g., GDS variables).

Figure 9-1 (Page 1 of 6). Control Vector Usage

Key	Control Vector Name	Applicable Message-Unit Structures by Parsing Rule	
		KL	LT
X′00′	SSCP-LU Session Capabilities	RSP(ACTLU)	
X′00′ (HPR only)	Node Identifier		RTP THDR Connection Setup (X′0D′) segment, RTP THDR Network Address (X′05′) control vector
X′01′	Date-Time	SETCV	
X′02′	Subarea Routing	SETCV	
X′03′	SDLC Secondary Station	SETCV	
X′03′ (HPR only)	Network Identifier		RTP THDR Connection Setup (X′0D′) segment, RTP THDR Network Address (X′05′) control vector
X′04′	LU	SETCV	
X′05′	Channel	SETCV	
X′05′ (HPR only)	Network Address		RTP THDR Connection Qualifier field
X′06′	Cross-Domain Resource Manager (CDRM)	ACTCDRM, RSP(ACTCDRM)	
X′07′	PU FMD-RU-Usage	RSP(ACTPU)	
X′08′	Intensive Mode	SETCV	
X′09′	Activation Request / Response Sequence Identifier	ACTCDRM, ACTPU, RSP(ACTCDRM), RSP(ACTPU)	
X′0A′	User Request Correlator (URC)	INIT-SELF Format 0	Cross-Domain Initiate GDS variable
X′0B′	SSCP-PU Session Capabilities	ACTPU, RSP(ACTPU)	
X′0C′	LU-LU Session Services Capabilities	RSP(ACTLU)	
X′0D′	Mode / Class-of-Service / Virtual-Route-Identifier List	BFCINIT, CINIT	

Figure 9-1 (Page 2 of 6). Control Vector Usage

Key	Control Vector Name	Applicable Message-Unit Structures by Parsing Rule	
		KL	LT
X′0E′	Network Name	ACTLU, ACTPU, BFCINIT, BIND, CINIT, CONTACT, INIT-OTHER, INIT-SELF Format 1, NSPE, TERM-SELF Format 1, RSP(BIND), XID3, Resource Identifier (X′19′) control vector, Session Information (X′2A′) control vector	CONNOUT, EXR, XID2, Locate GDS variable, Route Selection (X′2B′) control vector, Network-Qualified Name Pair (X′1C′) session key, Initiate-Other Cross-Domain GDS variable, FID2 Encapsulation GDS variable
X′0F′	Link Capabilities and Status	RSP(ACTLINK)	
X′10′	Product Set ID	XID3	XID2
X′11′	Load Module Correlation	RSP(ACTPU)	
X′12′	Network Identifier	NOTIFY	CONNOUT, XID2
X′13′	Gateway Support Capabilities	ACTCDRM, RSP(ACTCDRM)	
X′14′	Session Initiation	CDINIT, RSP(CDINIT)	
X′15′	Network-Qualified Address Pair	BFCINIT, CINIT, NOTIFY, SETCV, Session Information (X′2A′) control vector	
X′16′	Names Substitution	BFCINIT, CINIT, SETCV	
X′17′	SSCP Identifier	Resource Available NOTIFY vector	
X′18′	SSCP Name	ACTCDRM, ACTPU, RSP(ACTCDRM)	
X′19′	Resource Identifier	CDINIT, RSP(CDINIT), CDTERM, DSRLST, RSP(DSRLST), INIT-OTHER-CD, Resource Requested NOTIFY vector, Resource Available NOTIFY vector, Cancellation of Request for Notification NOTIFY vector	
X′1A′	NAU Address	CDINIT, SETCV, RSP(CDINIT)	
X′1B′	VRID List	BFTERM, NOTIFY, SETCV	
X′1C′	Network-Qualified Name Pair	CDTERM, Resource Available NOTIFY vector, Cancellation of Request for Notification NOTIFY vector	
X′1E′	VR-ER Mapping Data	BFSESSST, NOTIFY, SESSST, Session Information (X′2A′) control vector	
X′1F′	ER Configuration	ER-TESTED, NC-ER-TEST-REPLY	
X′23′	Local-Form Session Identifier	BFINIT, BFSESSST, SESSST, Session Information (X′2A′) control vector	
X′24′	IPL Load Module Request	RSP(ACTPU)	
X′25′	Security ID Control		REQCONT, FID2 Encapsulation GDS variable
X′26′	Network Connection Endpoint Identifier		Find Resource GDS variable, Found Resource GDS variable, Route Setup GDS variable, RTP THDR Network Address (X′05′) control vector
X′27′	XRF Session Activation	BFCINIT, BIND	
X′28′	Related Session Identifier	SESSST, RSP(ACTLU)	
X′28′ (HPR only)	Topic Identifier		RTP THDR Connection Setup (X′0D′) segment

Figure 9-1 (Page 3 of 6). Control Vector Usage

Key	Control Vector Name	Applicable Message-Unit Structures by Parsing Rule	
		KL	LT
X′29′	Session State Data	LU-LU Session Status NOTIFY vector, RSP(SWITCH)	
X′2A′	Session Information	BFSESSINFO, SESSST, RSP(ACTLU)	
X′2B′	Route Selection	BFCINIT, BFSESSST, BIND, CDINIT, CINIT, RSP(BIND), RSP(CDINIT), SESSST	Cross-Domain Initiate (Reply from NN server to client EN) GDS variable, Locate GDS variable
X′2C′	COS/TPF	BFCINIT, BIND, CDINIT, CINIT, DSRLST, RSP(CDINIT)	Cross-Domain Initiate GDS variable
X′2D′	Mode	BFCINIT, BIND, CINIT, DSRLST	
X′2F′	LU Definition	CDINIT, CINIT, RSP(CDINIT)	Cross-Domain Initiate GDS variable
X′30′	Assign LU Characteristics	RNAA	FID2 Encapsulation GDS variable
X′31′	BIND Image	BFSESSST, CDINIT, SESSST, RSP(CDINIT), RSP(DSRLST)	Cross-Domain Initiate GDS variable
X′32′	Short-Hold Mode	XID3	XID2
X′33′	ENCP Search Control		CP Capabilities GDS variable
X′34′	LU Definition Override	CDINIT, INIT-OTHER, INIT-OTHER-CD, INIT-SELF Format 0, INIT-SELF Format 1	Cross-Domain Initiate GDS variable, Initiate-Other Cross-Domain GDS variable
X′35′	Extended Sense Data	BFCLEANUP, BFSESSEND, BFTERM, BINDF, CDSESSEND, CDTERM, CLEANUP, SESSEND, TERM-OTHER, UNBIND, UNBINDF	Locate (Reply) GDS variable
X′36′	Directory Error		Delete Resource (Reply) GDS variable, Register Resource (Reply) GDS variable
X′37′	Directory Entry Correlator		Delete Resource (Request|Reply) GDS variable, Register Resource (Request|Reply) GDS variable
X′38′	Short-Hold Mode Emulation	SETCV	
X′39′	Network Connection Endpoint (NCE) Instance Identifier		Route Setup GDS variable, RTP THDR Connection Setup (X′0D′) segment, RTP THDR Network Address (X′05′) control vector
X′3A′	Route Status Data	RSP(ROUTE-TEST)	
X′3B′	VR Congestion Data	RSP(ROUTE-TEST)	
X′3C′	Associated Resource Entry		Delete Resource (Request) GDS variable, Find Resource GDS variable, Found Resource GDS variable, Register Resource (Request) GDS variable

Figure 9-1 (Page 4 of 6). Control Vector Usage

Key	Control Vector Name	Applicable Message-Unit Structures by Parsing Rule	
		KL	LT
X′3D′	Directory Entry		Delete Resource (Request) GDS variable, Find Resource GDS variable, Found Resource GDS variable, Register Resource (Request) GDS variable
X′3E′	Directory Entry Characteristic	CDINIT, CDSESSST, DSRLST, RSP(CDINIT), RSP(DSRLST)	Find Resource GDS variable, Found Resource GDS variable, Register Resource GDS variable
X′3F′	SSCP(SLU) Capabilities	CDCINIT, CDINIT, CINIT, RSP(CDINIT)	Cross-Domain Initiate GDS variable
X′40′	Real Associated Resource Entry		Find Resource GDS variable, Found Resource GDS variable
X′41′	Station Parameters	RNAA, SETCV (NS(c))	
X′42′	Dynamic Path Update Data	SETCV, RSP(SETCV)	
X′43′	Extended SDLC Station	RNAA, SETCV	FID2 Encapsulation GDS variable
X′44′	Node Descriptor		Topology Database Update GDS variable
X′45′	Node Characteristics		Topology Database Update GDS variable
X′46′	TG Descriptor	ACTCONNIN, CONTACT, XID3	CONNOUT, Route Selection (X′2B′) control vector, Cross-Domain Initiate GDS variable, FID2 Encapsulation GDS variable, Topology Database Update GDS variable, RTP THDR Switching Information (X′14′) segment′s HPR Return Route TG Descriptor (X′85′) control vector
X′47′	TG Characteristics		Cross-Domain Initiate GDS variable, Topology Database Update GDS variable
X′48′	Topology Resource Descriptor		Topology Database Update GDS variable
X′49′	Multinode Persistent Sessions (MNPS) LU Name		Route Setup GDS variable
X′4A′	Real Owning Control Point		Delete Resource GDS variable, Find Resource GDS variable, Found Resource GDS variable, Register Resource GDS variable
X′51′	DLUR/S Capabilities		FID2 Encapsulation GDS variable
X′52′	Primary Send Pacing Window Size	BFCINIT	
X′56′	Call Security Verification	SETCV	REQCONT, XID2
X′57′	DLC Connection Data		CONTACTED, REQCONT
X′59′	Installation-Defined CDINIT Data	CDINIT, CINIT, RSP(CDINIT)	

Figure 9-1 (Page 5 of 6). Control Vector Usage

Key	Control Vector Name	Applicable Message-Unit Structures by Parsing Rule	
		KL	LT
X′5A′	Session Services Extension Support	CDINIT, DSRLST, RSP(CDINIT), RSP(DSRLST)	
X′5B′	Interchange Node Support	CDINIT, DSRLST, RSP(CDINIT), RSP(DSRLST)	
X′5C′	APPN Message Transport	CDINIT, CDSESSST, CDTERM, DSRLST, INIT-OTHER-CD, ILU/TLU or Third-party SSCP Notification NOTIFY vector, Resource Requested NOTIFY vector, Resource Available NOTIFY vector, RSP(CDINIT), RSP(CDTERM), RSP(DSRLST), RSP(INIT-OTHER-CD)	
X′5D′	Subarea Message Transport	INIT-OTHER-CD	Cross-Domain Initiate GDS variable
X′5E′	Related Request	CDTERM	
X′5F′	Extended Fully Qualified PCID	BFCINIT, CDINIT, CINIT, INIT-OTHER, INIT-OTHER-CD	Cross-Domain Initiate GDS variable, Initiate-Other Cross-Domain GDS variable
X′60′	Fully Qualified PCID	ACTCDRM, BFCINIT, BFCLEANUP, BFSESSEND, BFSESSST, BFTERM, BIND, BINDF, CDCINIT, CDINIT, CDSESSEND, CDSESSST, CDTERM, CINIT, DSRLST, INIT-OTHER-CD, ILU/TLU or Third-party SSCP Notification NOTIFY vector, Resource Requested NOTIFY vector, Resource Available NOTIFY vector, Cancellation of Request for Notification NOTIFY vector, SESSEND, SESSST, SETCV, TERM-OTHER, UNBIND, UNBINDF, RSP(BIND), RSP(CDINIT), RSP(DSRLST), Session Information (X′2A′) control vector	FID2 Encapsulation GDS variable, Locate GDS variable, Notify GDS variable
X′61′	HPR Capabilities	CONTACT, XID3	
X′62′	Session Address	RSP(BIND) on RTP FID5 flows	
X′63′	Cryptography Key Distribution	CDINIT, DSRLST, RSP(CDINIT), RSP(DSRLST)	Cross-Domain Initiate (Request|Reply) GDS variable
X′64′	TCP/IP Information	CDINIT, CINIT, NOTIFY, RSP(ACTLU), RSP(CDINIT)	Cross-Domain Initiate (Request|Reply) GDS variable
X′65′	Device Characteristics	CDINIT, RSP(CDINIT), RSP(DSRLST)	Cross-Domain Initiate GDS variable
X′66′	Length-Checked Compression	BFCINIT, BIND, CDCINIT, CDINIT, CINIT, RSP(BIND), RSP(CDINIT)	Cross-Domain Initiate GDS variable
X′67′	Automatic Network Routing (ANR) Path		Route Setup GDS variable's Route Information (X′80′) control vector, HPR THDR Switching Information (X′14′) segment's HPR Switching Information (X′83′) control vector
X′68′	XRF/Session Cryptography	CDCINIT, CDINIT, CINIT, SESSST, RSP(ACTLU), RSP(CDINIT)	Cross-Domain Initiate GDS variable
X′69′	Switched Parameters	CDINIT, INIT-OTHER	
X′6A′	ER Congestion Data	RSP(ROUTE-TEST)	

Figure 9-1 (Page 6 of 6). Control Vector Usage

Key	Control Vector Name	Applicable Message-Unit Structures by Parsing Rule	
		KL	LT
X′71′	Triple DES Cryptography Key Continuation	BIND, CINIT, CDCINIT	
X′FE′	Control Vector Keys Not Recognized	ACTCDRM, RSP(ACTCDRM), RSP(CINIT)	

Control Vector Formats

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

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

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

Note: An HPR-only control vector also uses this control vector key.

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 the figure following BIND(BIND SESSION) in Chapter 6, “Request/Response Units (RUs)” for all possible values.
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

Node Identifier (X′00′) Control Vector

The Node Identifier control vector identifies a node by carrying the nonqualified CP name of the node.

Note: Another control vector also uses this control vector key in a non-HPR context.

Node Identifier (X′00′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′00′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data Node identifier: the nonqualified CP name of the node represented as a 1- to 8-byte type-1134 character-string, the first byte being limited to uppercase letters. The node identifier is unique when qualified by the Network ID (X′03′) control vector.

Date-Time (X′01′) Control Vector

Date-Time (X′01′) Control Vector

Byte	Bit	Content
0		Key: X′01′
1–12		Date, in EBCDIC: MM/DD/YY.ddd (MM = month; DD = day of month; YY = year; ddd = Nth day of year, 1–366) *Note:* Since a sender supplies only the last two digits of the year, the receiver must convert the year into 4-digit format before performing any operation that requires a 4-digit year. This conversion uses a sliding window based on the receiver′s understanding of the current date. The window extends from 90 years prior to the receiver′s current date to 10 years in the future. The receiver determines which operations require conversion to 4-digit years, such as whether or not to convert to 4-digit format before displaying a year to an operator.
13–20		Time, in EBCDIC: HH.MM.SS (HH = hours; MM = minutes; SS = seconds)

Subarea Routing (X′02′) Control Vector

Subarea Routing (X′02′) Control Vector

Byte	Bit	Content
0		Key: X′02′
1		Subarea address (left-justified)

SDLC Secondary Station (X′03′) Control Vector

Note: An HPR-only control vector also uses this control vector key.

SDLC Secondary Station (X′03′) Control Vector

Byte	Bit	Content
0		Key: X′03′
1		Reserved
2		Node type identifier for SPU:
	0–4	Reserved
	5–6	01 T2 10 T1
	7	Reserved
3		Type modifier:
	0	(reserved except when byte 2 identifies T1) 0 ¬ TS Profile 2 1 TS Profile 2
	1	0 discontinue link-level contact with adjacent T1|2 node if the T4 initiates an auto network shutdown procedure for the SSCP controlling that T1|2 node 1 continue link-level contact with adjacent T1|2 node if the T4 initiates an auto network shutdown procedure for the SSCP controlling that T1|2 node
	2	Polling type: 0 SNRM polling 1 XID polling
	3	0 modem test support for the SPU is as specified during system definition for the link 1 modem test is not supported for the SPU
	4–7	Reserved
4		SDLC BTU send limit *Note:* This value has an implied modulus for the SDLC send and receive counts. Less than 8 implies a modulus of 8; 8 or greater implies a modulus of 128.
5		Maximum number of consecutive BTUs sent from the primary station to the specified secondary station without another secondary station on the link being polled or being sent BTUs
6		Error retry indicator: X′00′ no immediate retry X′10′ immediate retry
7–8		Retired
9–10		Byte count of maximum BTU size permitted to be sent to the adjacent link station represented by the specified SPU

Network ID (X′03′) Control Vector

The Network ID control vector identifies a network.

Note: Another control vector also uses this control vector key in a subarea network context.

NETID (X′03′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′03′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data Network identifier: a unique 1- to 8-byte type-1134 symbol string, the first byte being limited to uppercase letters

LU (X′04′) Control Vector

LU (X′04′) Control Vector

Byte	Bit	Content
0		Key: X′04′
1		Local address form of LU network address
2	0– 1 2– 7	Reserved Secondary TC′s receive window size
3		Reserved, set to a value of 1
4		Retired, set to X′02′

Channel (X′05′) Control Vector

Note: An HPR-only control vector also uses this control vector key.

Channel (X′05′) Control Vector

Byte	Bit	Content
0		Key: X′05′
1– 2		Channel delay: minimum interval between successive inbound transmissions (binary, in tenths of a second)

Network Address (X′05′) Control Vector

The Network Address control vector contains addressing information pertaining to the sender.

Note: Another control vector also uses this control vector key in a subarea network context.

Network Address (X′05′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′05′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Network address type:
	0	Set to 1 if the network address is for the target of a point-to-point network connection
	1– 7	Reserved
3		Reserved.
4 – n		Control vectors, the details of which are described elsewhere in this chapter. *Note:* The following control vectors are always included and in the order specified. Each of the control vectors starts on a word boundary in this structure but the associated length field of each one contains the exact number of bytes of each control vector. (The length field in the imbedding X′05′ control vector does include any alignment pad bytes.) X′03′ Network ID control vector X′00′ Node Identifier control vector X′26′ NCE Identifier control vector X′39′ NCE Instance Identifier control vector

Cross-Domain Resource Manager (X′06′) Control Vector

Cross-Domain Resource Manager (X′06′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′06′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		CDRM profile: X′00′ (only value defined)
3– 5		CDRM usage:
	0	0 Name Pair (X′06′) session key supported
		1 Name Pair session key not supported
	1	0 Address Pair (X′07′) session key not supported

Cross-Domain Resource Manager (X′06′) Control Vector

Byte	Bit	Content
		1 Address Pair session key supported
		Note: If the control vector is omitted or the length is 0, the corresponding request or response implicitly specifies that the Name Pair (X′06′) session key is supported and the others are not.
	2	0 parallel sessions not supported
		1 parallel sessions supported
	3	0 URC not supported by SSCP (and all PLUs within its domain) in cross-domain session initiation
		1 URC supported by SSCP (and all PLUs within its domain) in cross-domain session initiation
	4	0 CDINIT (Type=DQ) (Format 1 or 4) with Type field bits specifying “leave on queue if dequeue retry is unsuccessful” not supported
		1 CDINIT (Type=DQ) (Format 1 or 4) with Type field bits specifying “leave on queue if dequeue retry is unsuccessful” supported (retired)
	5	0 PCID (X′05′) session key not supported
		1 PCID session key supported
	6	0 CDSESSEND from SSCP(SLU) and CDINIT(Format 2) not supported; requires NS-LSA to reset session knowledge; therefore, all sessions managed by the SSCP use virtual routes mapping to ER0 from the subarea of the SLU to the subarea of the PLU
		1 CDSESSEND from SSCP(SLU) and CDINIT(Format 2) supported; NS-LSA is not used to reset session knowledge; therefore, no ER restrictions exist for sessions managed by this SSCP
	7	0 The sender is not an APPN/subarea interchange node.
		1 The sender is an APPN/subarea interchange node.
	8	0 SSCP does not support the PLU Capability indicator in LU Status (X′01′) control list
		1 SSCP supports the PLU Capability indicator in LU Status (X′01′) control list
	9	0 Network-Qualified Address Pair (X′15′) session key not supported
		1 Network-Qualified Address Pair session key supported
	10	0 INIT-OTHER-CD Format 2 not supported
		1 INIT-OTHER-CD Format 2 supported
	11	0 INIT-OTHER-CD Format 3 not supported
		1 INIT-OTHER-CD Format 3 supported
	12	0 Formats 3 and 4 of CDINIT not supported
		1 Formats 3 and 4 CDINIT supported: includes NAU Address (X′1A′) control vector
		Note: If control vector X′13′ is also included in this ACTCDRM request or response, CDINIT format 3 or 4 may include additional control vectors for cross-network session setup.
	13	0 Format 1 of CDCINIT not supported: includes Network-Qualified Address Pair (X′15′) session key
		1 Format 1 CDCINIT supported
	14	0 NOTIFY NS(s) key X′06′, NOTIFY NS(s) key X′08′, and DSRLST (X′02′) not supported
		1 NOTIFY NS(s) key X′06′, NOTIFY NS(s) key X′08′, and DSRLST (X′02′) supported
	15	0 notification of lost session (LU-LU) awareness not supported
		1 notification of lost session (LU-LU) awareness supported (The SSCP sends CDSESSEND if it has lost awareness of the session identified by the Session Key Content field in the CDSESSEND.)
	16	Support of CDINIT request for notification of DLU availability:
		0 CDINIT (byte 20, bit 7) request for notification of DLU availability not supported
		1 CDINIT (byte 20, bit 7) request for notification of DLU availability supported. NOTIFY NS(s) key X′08′ is sent if NOTIFY NS(s) key X′08′ is supported (see bit 14 above); otherwise, NOTIFY NS(s) key X′07′ with session key X′01′ is sent.
	17	0 backup session request not supported in CDINIT
		1 backup session request supported in CDINIT

Cross-Domain Resource Manager (X′06′) Control Vector

Byte	Bit	Content
	18	ENA support: 0 ENA not supported 1 ENA supported
	19	0 Network-Qualified Names Support indicator in CDINIT and RSP(CDINIT) not supported 1 Network-Qualified Names Support indicator in CDINIT and RSP(CDINIT) supported; this implies that the sending SSCP supports sending and receiving fully qualified PCIDs on the SSCP-SSCP session
	20	Support of NOTIFY X′09′ to indicate DLU unavailability and cancel any request for notification (see CDINIT byte 20, bit 7): 0 NOTIFY X′09′ not supported 1 NOTIFY X′09′ supported
	21	Use of CDTERM with an Extended Sense Data (X′35′) control vector as a means of reporting session failures: 0 CDTERM with control vector X′35′ not supported 1 CDTERM with control vector X′35′ supported
	22	Intermediate gateway SSCP response to unrecognized control vectors on a session services RU: 0 The gateway SSCP negatively responds to the RU. 1 The gateway SSCP passes unrecognized control vectors through without change.
	23	Nonnative network LU attachment: 0 not supported 1 supported
6	0	Extended subarea address support: 0 Extended subarea address not supported 1 Extended subarea address supported
	1–3	Reserved
	4–7	Extended subarea address limit: 0000 subarea address limit = 255 0001 subarea address limit = 511 0010 subarea address limit = 1023 0011 subarea address limit = 2047 0100 subarea address limit = 4095 0101 subarea address limit = 8191 0110 subarea address limit = 16383 0111 subarea address limit = 32767 1000 subarea address limit = 65535
7(=n)	0	0 Sender is not capable of resolving ACTCDRM contention based on the network-qualified SSCP name. 1 Sender is capable of resolving ACTCDRM contention based on the network-qualified SSCP name.
	1	0 Sender is not gateway-SSCP capable. 1 Sender is gateway-SSCP capable.
	2	0 Adjacent SSCP does not support DACTCDRM type 4. 1 Adjacent SSCP supports DACTCDRM type 4.
	3	0 Sender does not fully support network-qualified names; the SSCP may send alias names for resources in the X′06′ session key. 1 Sender fully supports network-qualified names; the SSCP will never send alias names in the X′06′ session key.
	4–7	Reserved

Cross-Domain Resource Manager (X′06′) Control Vector

Byte	Bit	Content
Note:		This control vector is sent on ACTCDRM and RSP(ACTCDRM) to define the *receive* capabilities of the SSCP building the request or response. An SSCP reports all its capabilities. If an SSCP does not report support of a particular function, its session partner SSCP is responsible for not invoking that function. An SSCP receiving the control vector in an ACTCDRM request or response ignores bits within the usage indicators that it does not understand. In its own control vector, the SSCP sets such bits to 0, indicating that it does not support the function.

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

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

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

Intensive Mode (X′08′) Control Vector

Intensive Mode (X′08′) Control Vector

Byte	Bit	Content
0		Key: X′08′
1	0	0 reset intensive mode 1 set intensive mode
	1– 7	Reserved
2– 3		Maximum number of intensive mode records (IMRs)

Activation Request/Response Sequence Identifier (X′09′) Control Vector

Activation Request/Response Sequence Identifier (X′09′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′09′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2– 9 (= n)		Activation request/response sequence identifier: an 8-byte binary value, generated by the sender of ACTCDRM, RSP(ACTCDRM), ACTPU, and echoed in RSP(ACTPU), and used by the receiver to determine whether the current RU supersedes a previously received RU from the same sender (If the current RU has an activation request/response sequence identifier value greater than the corresponding activation request/response sequence identifier value of the earlier ACTPU, ACTCDRM, or RSP(ACTCDRM), the current RU is accepted and processed, while the earlier RU is superseded. The 8-byte field has the following characteristic: If n1 was generated at time t1, and n2 was generated at time t2, and t1 < t2, then n1 < n2.)

User Request Correlation (URC) (X′0A′) Control Vector

The User Request Correlation control vector carries an implementation-defined correlator that allows an SLU to correlate an incoming BIND with a previous SLU-initiated session request.

User Request Correlation (URC) (X′0A′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′0A′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2 – n		URC: implementation-defined correlator

Session Capabilities (X′0B′) Control Vector

SSCP-PU Session Capabilities (X′0B′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′0B′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data

SSCP-PU Session Capabilities (X′0B′) Control Vector

Byte	Bit	Content
2–3	0	Lost subarea requirement: 0 NS-LSA required 1 NS-LSA not required
	1	ALS station support: 0 adjacent link station network address not supported 1 adjacent link station network address supported
	2	SNA Network Interconnect (SNI) gateway support: 0 no SNI gateway support 1 this component supports SNI gateway function
	3	Notification of other-network lost route: 0 the sending SSCP does not request, or the sending PU T4|5 does not support, notification of the SSCP (via ROUTE-INOP or ER-TESTED) of an inoperative route in subnetworks other than the SSCP′s network 1 the sending SSCP does request, or the sending PU T4|5 does support, notification of the SSCP (via ROUTE-INOP or ER-TESTED format 2) of an inoperative route in subnetworks other than the SSCP′s network
	4	Notification of same-network lost route: 0 the sending SSCP does not request, or the sending PU T4|5 does not support ROUTE-INOP; ER-TESTED format 1 is used for routes in the sender′s subnetwork 1 the sending SSCP does request, or the sending PU T4|5 does support, ROUTE-INOP for lost VRs or ERs in the sender′s subnetwork; ER-TESTED format 2 may be used *Note:* An SSCP always receives ER-TESTED for routes in the SSCP′s own subnetwork; additionally, this bit indicates whether ROUTE-INOP may flow for lost ERs or VRs in the SSCP′s own subnetwork.
	5	CONTACTED(Loaded) format: 0 CONTACTED (X′09′) is not supported; CONTACTED (X′04′) is used 1 CONTACTED (X′09′) is supported
	6	ENA support: 0 ENA not supported 1 ENA supported
	7	Extended BIND support indicator: 0 extended BIND not supported 1 extended BIND supported *Note:* The value 1 indicates the sending SSCP supports receipt of BFINIT, BFTERM, BFSESSINFO, RSP(ACTLU) with the Session Information control vector, BFSESSST, BFSESSEND, CONTACTED X′0A′, and CONTACTED X′0B′; and the sending PU T4|5 supports receipt of extended BINDs, stand-alone BINDs from APPN nodes, nonactivation CONTACT, RNAA assignment types 4 and 5, control vector X′43′, and BFCLEANUP from its SSCP.
	8	Enhanced disk support: 0 enhanced disk support not supported 1 enhanced disk support supported
	9	Extended SDLC Secondary Station control vector (X′43′) support: 0 receipt of base control vector (bytes 0–10 with bytes 7–8 reserved) supported 1 receipt of extended control vector (bytes 0–16) supported
	10	Move PU support bit: 0 Move PU is not supported. 1 Move PU is supported.
	11	HPR support bit: 0 HPR is not supported. 1 HPR is supported.
	12	Reserved
	13	Nonnative network LU attachment: 0 not supported 1 supported

SSCP-PU Session Capabilities (X′0B′) Control Vector

Byte	Bit	Content
	14	Switched session continuation support: 0 nondisruptive takeover of sessions on switched links not supported 1 nondisruptive takeover of sessions on switched links supported
	15	Dynamic path update (DPU) capability: 0 DPU not supported 1 DPU supported
4	0	Extended subarea address support: 0 extended subarea address not supported 1 extended subarea address supported
	1	Border node and connection network support indicator: 0 extended border node connections and connection networks not supported 1 extended border node connections and connection networks supported
	2	HPR border node support indicator: 0 HPR border node not supported 1 HPR border node supported
	3	PLU BFSESSST/BFSESSEND required indicator: 0 BFSESSST/BFSESSEND required 1 BFSESSST/BFSESSEND not required
	4–7	Extended subarea address limit: 0000 subarea address limit = 255 0001 subarea address limit = 511 0010 subarea address limit = 1023 0011 subarea address limit = 2047 0100 subarea address limit = 4095 0101 subarea address limit = 8191 0110 subarea address limit = 16383 0111 subarea address limit = 32767 1000 subarea address limit = 65535
5	0	Extended REQUEST CONTACT support: 0 extended REQUEST CONTACT not supported 1 extended REQUEST CONTACT supported
	1	Forced NCP dump support: 0 Forced NCP dump not supported 1 Forced NCP dump supported
	2	SNI gateway session accounting support: 0 not included in this gateway node 1 included in this gateway node
	3	Dynamic network-connection notification support: When sent by a gateway PU, this bit indicates that the gateway PU supports sending of the Dynamic Network-Connection Notification NOTIFY vector; and when sent by a gateway SSCP, this bit indicates that the gateway SSCP supports reception of the Dynamic Network-Connection Notification NOTIFY vector. This bit is defined only when the Gateway Support bit (byte 2, bit 2) is 1. 0 Dynamic Network-Connection Notification NOTIFY vector not supported 1 Dynamic Network-Connection Notification NOTIFY vector supported
	4	APPN networking functions support implies indicator — support of parallel TGs through the boundary function, CP-CP sessions through the boundary function, and APPN networking level indicators and services: 0 APPN networking functions not supported 1 APPN networking functions supported
	5	XRF cryptography support: 0 XRF cryptography not supported 1 XRF cryptography supported
	6	Processing support for length-checked compression information on XRF BIND: 0 not supported 1 supported

SSCP-PU Session Capabilities (X′0B′) Control Vector

Byte	Bit	Content
	7	Selective ROUTE-INOP support: 0 selective ROUTE-INOP not supported; sent for all operative ERs 1 selective route INOPs supported; sent for operative ERs that have active VRs
6(=n)	0	Network name forwarding support: 0 Network name forwarding not supported: The sender does not support NCP forwarding of Network Name control vectors on ACTPUs and ACTLUs to PUs and their associated LUs. 1 Network name forwarding supported: The sender supports NCP forwarding of Network Name control vectors on ACTPUs and ACTLUs to PUs and their associated LUs.
	1–7	Reserved

Note: This control vector is sent on ACTPU and RSP(ACTPU) for PU T4|5 to define the capabilities of the SSCP or PU T4|5 building the request or response. The NAU reports all its capabilities; if a NAU does not report support for a particular function, its session partner is responsible for not invoking that function. The receiving NAU ignores bits that it does not understand.

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 9-5 and Figure 9-1 on page 9-7)
2 – m		Vector Data
2	0–3	Primary LU capability (used between a subarea LU and its SSCP; also used between a peripheral LU and its SSCP if the BF(LU) supports the receipt of CINIT; otherwise, reserved): 0000 PLU capability is inhibited: Sessions can be neither queued nor started. 0001 PLU capability is disabled: Sessions can be queued but not started. 0010 reserved 0011 PLU capability is enabled: Sessions can be queued or started.
	4–7	Secondary LU capability: 0000 SLU capability is inhibited: Sessions can be neither queued nor started. 0001 SLU capability is disabled: Sessions can be queued but not started. 0010 reserved 0011 SLU capability is enabled: Sessions can be queued or started.
3–4		LU-LU session limit: 0000 no session limit specified (only value allowed for subarea LUs) 0001 session limit of 1 (only value allowed for peripheral LUs)

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

Byte	Bit	Content
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 *Note:* For XRF, this field applies to only the XRF-active session. XRF-backup sessions are not included in this count.
7	0	Parallel session support: 0 parallel sessions not supported 1 parallel sessions supported
	1	NOTIFY support: 0 NOTIFY not supported 1 NOTIFY supported
	2	SESSST capability (used between a subarea or dependent LU or BF and its SSCP; otherwise, reserved): 0 SESSST RU is suppressed if SLU 1 SESSST RU is sent if SLU
	3	XRF Session Activation (X′27′) control vector support (used between a BF and its SSCP; otherwise, reserved): 0 XRF Session Activation (X′27′) control vector not supported on BIND 1 XRF Session Activation (X′27′) control vector supported on BIND
	4	Peripheral node extended BIND support indicator (used between a dependent LU and its BF; otherwise, reserved): 0 Dependent LU does not support receipt of extended BINDs. 1 Dependent LU does support receipt of extended BINDs.
	5	Network-qualified names support indicator (used between an LU and its SSCP; passed through the BF to the SSCP if sent by a peripheral LU): 0 A BIND received by this LU cannot contain network-qualified LU names in bytes k+2 – m and p+2 – r. 1 A BIND received by this LU can contain network-qualified LU names in bytes k+2 – m and p+2 – r.
	6	Subarea node extended BIND support indicator (used between a subarea LU or BF and its SSCP; otherwise, reserved): 0 Subarea LU or BF(LU) does not support sending and receiving extended BIND. 1 Subarea LU or BF(LU) does support sending and receiving extended BIND.
	7	Retired (set to 0)
8–15		Retired (set to X′4040404040404040′)
16(=m)	0	Unrecognized-control-vectors-on-CINIT support indicator: 0 CINIT containing unrecognized control vectors will be rejected. 1 CINIT containing unrecognized control vectors will be accepted.
	1–7	Reserved

Mode/Class-of-Service/Virtual-Route-Identifier List (X′0D′) Control Vector

Mode/Class-of-Service/Virtual-Route-Identifier List (X′0D′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′45′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data

Mode/Class-of-Service/Virtual-Route-Identifier List (X′0D′) Control Vector

Byte	Bit	Content
2– 9		Mode name: an 8-character symbolic name (implementation and installation dependent) of type-A symbol-string characters 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
10– 1 7		COS name: symbolic name of class of service in EBCDIC characters
18 – n		Virtual Route Information
18		Length (in bytes)—including format, type, number of entries, and entries of Virtual Route Information field
19		Format of virtual route identifier list: X′00′ format 0 (only value defined)
20		Type of virtual route required: X′00′ only virtual routes mapping to ER0 from the subarea of the SLU to the subarea of the PLU may be used X′01′ virtual routes mapping to any ERN may be used
21		Number of entries in the virtual route identifier list
22 – n		Virtual route identifier list: 2-byte (VRN, TPF) entries where VRN is one byte and TPF is one byte

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 9-5 and Figure 9-1 on page 9-7) *Note:* A null X′0E′ control vector consists of a vector header with no vector data. The length field is set appropriately.

Network Name (X′0E′) Control Vector

Byte	Bit	Content
2 – n		Vector Data
2		Network name type: X′C•′ Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage X′F1′ PU name X′F3′ LU name X′F4′ CP name (see Note) X′F5′ SSCP name X′F6′ NNCP name X′F7′ link station name (not network-qualified) X′F8′ CP name of CP(PLU) (used only when this control vector is carried on the X′2A′ control vector) X′F9′ CP name of CP(SLU) (used only when this control vector is carried on the X′2A′ control vector) X′FA′ generic resource name or uninterpreted name *Note:* When this control vector is carried in some message units, such as XID3 or BIND, X′F4′ means simply “CP name,” without specifying the CP type (e.g., EN or NN), and X′F6′ is not used; see each individual message-unit structure in which this control vector appears for details on such usage.
For byte 2 = X′F1′ through X′F9′		
3 – n		Network-qualified name: a 1 to 17-byte name consisting of an optional qualifier concatenated to a 1 to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1 to 8-byte type 1134-symbol-string network identifier concatenated with a period (when the qualifier is not present, the period is omitted). The network-qualified name appears, for example, as follows: NETID.NAME, with optional (but not significant) trailing, but no imbedded, space (X′40′) characters. As noted in Appendix A, “SNA Character Sets and Symbol-String Types,” implementation usage constrains the leading character of the name to be alphabetic. *Note:* When this control vector is carried in an RSCV, the net ID may be omitted from this field in contiguous control vectors *after* the first in a series pertaining to the same net-ID subnetwork (to conserve RSCV total length).
For byte 2 = X′FA′		
3 – n		Generic resource name or uninterpreted name: a 1 to 17-byte EBCDIC character string

Link Capabilities and Status (X′0F′) Control Vector

Link Capabilities and Status (X′0F′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′0F′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data

Link Capabilities and Status (X′0F′) Control Vector

Byte	Bit	Content
2(=n)	0	Control vector X′43′ support: 0 control vector X′43′ not supported 1 control vector X′43′ supported
	1	DACTLINK(Give-back) support: 0 DACTLINK(Give-back) not supported 1 DACTLINK(Give-back) supported
	2	Link connection status: 0 link does not have a switched connection 1 link already has a switched connection
	3	DUMPINIT DLC-level support: 0 DLC-level dump is supported 1 DLC-level dump is not supported
	4–5	Link management protocol: 00 Managed objects do not exist for this resource 01 Managed objects do exist for this resource 10 and 11 are reserved
	6	Physical resource identifier: 0 Not a physical resource 1 A physical resource
	7	Support for Control Vector X′69′ (with DLCADDR specification of DLC data on CONNOUT: 0 control vector X′69′ not supported 1 control vector X′69′ supported

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 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Retired
3 – n		Network product identifier: one or two Product Identifier (X′11′) MS common subvectors, as described in *SNA Management Services Formats*, one for each hardware product and software product in the implementation of the node

Load Module Correlation (X′11′) Control Vector

Load Module Correlation (X′11′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′11′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2 – n		Load module correlator

Network Identifier (X′12′) Control Vector

Network Identifier (X′12′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′12′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data.
2 – n		Network identifier: a 1- to 8-byte type-1134 symbol string

Gateway Support Capabilities (X′13′) Control Vector

Gateway Support Capabilities (X′13′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′13′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – m		Vector Data
2 – m		A pair of session keys, as described in the “Session Keys” discussion in Chapter 9, “Common Fields” *Note:* These session keys appear in the following order: X′15′ network-qualified address pair (NAU 1 and NAU 2 define the sender′s address and the destination address, respectively, as known in the network of the sender.) X′15′ network-qualified address pair (NAU 1 and NAU 2 define the origin address and the destination address, respectively, as known in the network adjacent to the sender.)

Session Initiation (X′14′) Control Vector

Session Initiation (X′14′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′14′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2–9		Network ID 1: for CDINIT, the network ID of the subnetwork containing the DLU; for RSP(CDINIT), the network ID of the subnetwork on the DLU side of the gateway node *Note:* Network ID 1 and COS name 1 fields are reserved in CDINIT if COS name was not received from the ILU or if control vector X′19′ for the DLU indicates that the translation of alias to real DLU name has not occurred.
10–17		COS name 1: for CDINIT and RSP(CDINIT), the COS name as known in the above network
18–25		Network ID 2: The network ID of the subnetwork on the OLU side of the SNI gateway node; used along with the subsequent COS name 2 field in conjunction with the network ID 1 and COS name 1 fields to distribute the COS name as known on both sides of the gateway node to the gateway SSCP responsible for COS name to VRID resolution in the shared control gateway environment *Note:* Network ID 2 is reserved for CDINIT or when RSP(CDINIT) is flowing from one gateway to another and if COS name translation has not yet occurred in this gateway for RSP(CDINIT).
26–33		COS name 2: for RSP(CDINIT), the COS name as known in the above network *Note:* COS name 2 is reserved for CDINIT or when RSP(CDINIT) is flowing from one gateway to another and if COS name translation has not yet occurred in this gateway for RSP(CDINIT).
34		Usage indicators:
	0	Parallel session capabilities of adjacent SSCP on the OLU side of the gateway node: 0 parallel sessions not supported 1 parallel sessions supported *Note:* This bit is set by the first gateway SSCP on each gateway for CDINIT and is reserved for RSP(CDINIT).
	1	Configuration information: 0 RU sender and receiver not in the same SNI gateway (i.e., the gateway that is to be used by the intended cross-network LU-LU session) 1 RU sender and receiver in the same gateway (The gateway reference is relative to the cross-network LU-LU session, not the cross-network SSCP-SSCP session between the RU sender and receiver.) *Note:* Network ID 2, COS name 2, and the second instance of session key X′15′ are reserved in CDINIT and RSP(CDINIT) flowing from one gateway to another (when bit 1 = 0).
	2	Address aliasing support (reserved when byte 34, bit 1 is 0): 0 sender not designated for address aliasing support 1 sender designated for address aliasing support *Note:* This bit is used to signal whether the sender is the predesignated gateway SSCP for address aliasing support of the gateway node specified in bytes 43–46.
	3–7	Reserved
35–42		Mode name as known in the destination network *Note:* The mode name as known in the network of the OLU is contained in bytes 21–28 of CDINIT.

Session Initiation (X′14′) Control Vector

Byte	Bit	Content
43–46		Subarea address of the SNI gateway PU that is to be used for the intended LU-LU session (reserved when byte 34, bit 1 is 0)
47–54		Network ID of the subnetwork in which the above subarea address is valid (reserved when byte 34, bit 1 is 0)
55 – n		A pair of session keys, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* The following session keys are used: X′15′ network-qualified address pair (session key 1): OLU and DLU respectively Between gateways, usage is as follows: • CDINIT: address pair on the OLU side of the gateway when received, on the DLU side of the gateway when sent. The DLU address is reserved. • RSP(CDINIT): address pair on the DLU side of the gateway when received, on the OLU side of the gateway when sent. Within a gateway, usage is as follows: • CDINIT: address pair on the DLU side of the gateway; if the SSCP with alias address responsibility has not been reached, session key 1 is reserved (i.e., it is present with a length of 0, or present with nonzero length and contents of 0′s). Otherwise, only the DLU portion of the address pair is reserved. • RSP(CDINIT): address pair on the OLU side of the gateway. *Note:* Session key 1 is used both between gateways (byte 34, bit 1 = 0) and within a gateway (byte 34, bit 1 = 1). X′15′ network-qualified address pair (session key 2): OLU and DLU respectively Between gateways, session key 2 is reserved (as described above). Within a gateway, usage is as follows: • CDINIT: address pair on the OLU side of the gateway; upon entry into a gateway, session key 1 in the input CDINIT is copied into session key 2 of the output CDINIT. The DLU address is reserved until the SSCP with alias address responsibility is reached. • RSP(CDINIT): address pair on the DLU side of the gateway; upon entry into a gateway, session key 1 in the input RSP(CDINIT) is copied into session key 2 of the output RSP(CDINIT). *Note:* Session key 2 is used only within a gateway; otherwise, it is reserved (i.e., it is present with a length of 0, or present with nonzero length and contents of 0′s).

Network-Qualified Address Pair (X′15′) Control Vector

Network-Qualified Address Pair (X′15′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′15′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2–7		NAU 1 network address

Network-Qualified Address Pair (X′15′) Control Vector

Byte	Bit	Content
8–13		NAU 2 network address *Note:* See the RUs that carry this vector for NAU1/NAU2 definitions and order requirements.
14–21 (=n)		Network ID of the subnetwork in which the above addresses are valid *Note:* If the Network ID field contains all space (X′40...40′) characters, the network addresses are in the sender′s network; if the network ID is not included (i.e., Vector Data length = 12), the network addresses are in the sender′s network.

Names Substitution (X′16′) Control Vector

Names Substitution (X′16′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′16′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Length of PLU name
3 – m		PLU name
m + 1		Length of SLU name
m + 2 – n		SLU name
Note:		When this control vector is contained in SETCV: • The PLU name is an alias name • The SLU name is a real name • The Network-Qualified Address Pair control vector always accompanies this control vector
Note:		When this control vector is contained in CINIT or BFCINIT: • The PLU name is a real name or an uninterpreted name • The SLU name is an alias name or a real name

SSCP Identifier (X′17′) Control Vector

SSCP Identifier (X′17′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′17′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data

SSCP Identifier (X′17′) Control Vector

Byte	Bit	Content
2		SSCP visit count (set by the first gateway SSCP and then decremented at each gateway SSCP on the path) *Note:* This field is ignored by the receiver if byte 3, bit 1 is 0.
3		Usage indicators:
	0	Reserved
	1	Target resource indicator: 0 the resource named in this vector is not the target resource 1 the resource named in this vector is the target resource
	2–7	Reserved
4		Length of network ID
5 – m		Network ID of the subnetwork containing the SSCP
m + 1		Length of SSCP name
m + 2 – n		Name of the SSCP

SSCP Name (X′18′) Control Vector

SSCP Name (X′18′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′18′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2–9		Name of SSCP: symbolic name in EBCDIC characters
10–17 (= n)		Network ID of the subnetwork containing the SSCP

Resource Identifier (X′19′) Control Vector

The Resource Identifier (X′19′) control vector is used to identify specific characteristics of the resource described within the control vector.

Resource Identifier (X′19′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′19′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – s		Vector Data

Resource Identifier (X′19′) Control Vector

Byte	Bit	Content
2		SSCP visit count (set by the first gateway SSCP and then decremented at each gateway SSCP on the path) *Note:* This field is ignored by the receiver if byte 3, bit 1 is 0.
3		Usage indicators:
	0	Name translation: 0 translation has not occurred for this name 1 translation has occurred for this name
	1	Target resource indicator: 0 the resource named in this vector is not the target resource 1 the resource named in this vector is the target resource
	2	Real resource owner indicator: 0 The owner indicated may not be the real owner of the resource 1 The owner indicated is the real owner of the resource
	3–4	Reserved
	5	Name resolution (reserved in CDTERM and RSP(DSRLST)): 0 generic name resolution should not be performed 1 generic name resolution should be performed
	6–7	Characteristic of the resource named in this control vector (reserved except in CDTERM and RSP(DSRLST)): 00 the characteristic is undetermined 01 the LU name will not change 10 the LU name may change due to a failure; it should be validated if an abnormal session termination occurs 11 the LU name may change frequently; it should be validated for every session initiation
4		Length of following SSCP name
5 – m		Network name of SSCP that controls the LU (see Note 2)
m + 1		Length of following network identifier
m + 2 – n		Network identifier of the subnetwork containing the LU (see Note 2)
n + 1		Length of following LU name
n + 2 – p		Real network name of the LU (see Note 2)
p + 1		Length of following network identifier
p + 2 – q		Network identifier of the subnetwork in which the LU name alias is known (see Note 2)
q + 1		Length of following LU name
q + 2 – r		Network name of the LU (see Note 2)
r – s		Control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note: 1* The following control vectors may be included; they are parsed according to parsing rule KL. X′0E′ Network Name control vector: (present when the owning CP information is known and the sending node supports inclusion of this vector) specifies the network-qualified name of the SSCP or CP that controls the real LU referenced in bytes n+2 – p and will always be of type X′F4′ (CP name). *Note: 2* The network names and network identifiers above are encoded using coded graphic character set 01134–00500 with the first character being alphabetic.

NAU Address (X′1A′) Control Vector

NAU Address (X′1A′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′1A′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2– 7		Network address of the NAU
8		ENA support:
	0	0 NAU does not support ENA
		1 NAU supports ENA
	1– 7	Reserved
9(=n)		Extended subarea address support:
	0	0 Extended subarea address not supported
		1 Extended subarea address supported
	1– 3	Reserved
	4– 7	Extended subarea address address limit
		0000 Subarea address limit = 255
		0001 Subarea address limit = 511
		0010 Subarea address limit = 1023
		0011 Subarea address limit = 2047
		0100 Subarea address limit = 4095
		0101 Subarea address limit = 8191
		0110 Subarea address limit = 16383
		0111 Subarea address limit = 32767
		1000 Subarea address limit = 65535

VRID List (X′1B′) Control Vector

VRID List (X′1B′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′1B′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2– 9		Network ID
10 – n		Virtual Route Information Field
10		Format of virtual route list:
		X′00′ format 0 (only value defined)
11		Type of virtual route required:
		X′00′ Only virtual routes mapping to ER0 from the subarea of the SLU to the subarea of the PLU may be used.
		X′01′ Virtual routes mapping to any ERN may be used.

VRID List (X′1B′) Control Vector

Byte	Bit	Content
12		Number of entries in the Virtual Route Information field:
13 – n		Virtual route list: 2-byte (VRN, TPF) entries, where VRN is one byte and TPF is one byte

Network-Qualified Name Pair (X′1C′) Control Vector

Network-Qualified Name Pair (X′1C′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′1C′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Session type: X′00′ SSCP-SSCP X′01′ LU-LU
3 – n		Control vectors *Note:* See “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7 X′0E′ Network Name control vector X′0E′ Network Name control vector *Note:* The names belong to the two session partners.

VR-ER Mapping Data (X′1E′) Control Vector

VR-ER Mapping Data (X′1E′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′1E′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		VRN and TPF data:
	0– 3	Virtual route number (VRN) assigned to the session indicated in the containing RU
	4– 5	Reserved
	6– 7	Transmission Priority field (TPF) assigned to the session indicated in the containing RU
3		Explicit route data:
	0– 3	Reserved
	4– 7	Outbound ERN for the VRN specified in byte 2, bits 0– 3

VR-ER Mapping Data (X′1E′) Control Vector

Byte	Bit	Content
4(=n)		Reverse explicit route data:
	0– 3	Reserved
	4– 7	RERN corresponding to the ERN in byte 3

ER Configuration (X′1F′) Control Vector

ER Configuration (X′1F′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′1F′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – p		Vector Data
2		Outbound TG number (reserved in last vector)
3		Inbound TG number (reserved in first vector)
4– 7		Subarea address of the PU that has appended this control vector, in the network in which the containing RU is flowing
8		Number of SSCP Address fields in bytes 9 – m
9 – m		SSCP address fields: list of 8-byte fields, one for each SSCP that currently controls at least one active link in any TG underlying the tested ER or has an active SSCP-PU session
Note: *The format of each 8-byte field is:*		
0		Reserved
1	0– 5	Reserved
	6	Set to 1 if this SSCP has at least one link active in the TG over which the ER test will be sent, as specified in the Outbound TG Number field (byte 2 of this control vector)
	7	Set to 1 if this SSCP has at least one link active in the TG over which the ER test was received, as specified in the Inbound TG Number field (byte 3 of this control vector) *Note:* If bits 6 and 7 are both 0, the address in bytes 2 through 7 is the address of an SSCP that did not issue an ACTLINK for any link in either the inbound or outbound TG. In this case, it is an SSCP that has an SSCP-PU session with this node.
2– 7		Address of the SSCP
Note: *End of the 8-byte field format*		
m + 1		Length, in binary, of network ID *Note:* When the length is 0, the network ID is the same as that of the subnetwork containing the ER, and the fields defined in bytes m+2 – p are not included.
m + 2 – n		Network ID of the network in which the SSCP addresses are known
n + 1 – p		4-byte subarea address of the PU that has appended this control vector, as known in the network defined in bytes m+2 – n

ER Congestion Data (X′20′) Control Vector

(Retired) This control vector has been replaced by ER Congestion Data control vector (X′6A′)

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 9-5 and Figure 9-1 on page 9-7)
2 – n(=4|8)		Vector Data
2– 3		Error byte offset: the binary offset (0-origin in the XID information field) of the first byte of the field in error
4		Error bit offset: the binary offset (0-origin in the byte pointed to in the Error Byte Offset field) of the first bit of the field in error
5– 8		Optional sense data

Local-Form Session Identifier (X′23′) Control Vector

Local-Form Session Identifier (X′23′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′23′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Format: X′02′ Format 2: FID2 session identifier X′03′ Format 3: FID3 session identifier (reserved for T2.1 nodes)
For format 2—FID2		
3 – n		Session identifier (SID) for Format 2—FID2
3		High-order byte of session identifier (SIDH): OAF′ in the TH of the BIND
4		Low-order byte of session identifier (SIDL): DAF′ in the TH of the BIND
5(=n)		Flags:
	0– 5	Reserved
	6	ODAI field from the TH of the BIND
	7	Reserved
For format 3—FID3		

Local-Form Session Identifier (X′23′) Control Vector

Byte	Bit	Content
3 – n		Session identifier for Format 3—FID3
3(=n)		LSID from TH of the BIND

IPL Load Module Request (X′24′) Control Vector

IPL Load Module Request (X′24′) Control Vector

Byte	Bit	Content
0		Vector header; Key=X′24′
		Vector Data
1– 8 (= n)		Load module ID: a 1- to 8-character type-A symbol string identifying the requested IPL load module: X′4040...40′ any load module will be accepted ¬ X′4040...40′ identifies specific load module name

Security ID Control (X′25′) Control Vector

The Security ID control vector is available for sending identifier information for a switched call to the SSCP. The information may be provided by the X.25 or switched telephone network.

Security ID Control (X′25′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′25′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – q		Vector Data
2		Reserved
3 – q		Security ID
3		Length (q–4, in binary, of security ID)
4 – q		Security ID: a string of EBCDIC characters *Note:* In the X.25 environment, the security ID is the calling DTE address provided by the network, once converted from packed digits to EBCDIC characters.

NCE Identifier (X′26′) Control Vector

The NCE Identifier control vector contains a 1- to 8-byte identifier of the network connection endpoint of an RTP connection.

NCE Identifier (X′26′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′26′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data Network connection endpoint identifier (NCE ID): a 1- to 8-byte identifier, the structure of which is determined by the implementing product, except that the high-order bit is always set to 1, and X′FF′ is not a valid byte in the string (since it is used as an ANR routing field ending delimiter)

XRF Session Activation (X′27′) Control Vector

XRF Session Activation (X′27′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′27′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Usage indicators:
	0	Session type: 0 XRF-active: The BIND request is for an XRF-active session. 1 XRF-backup: The BIND request is for an XRF-backup session. The boundary function will relate this session to the previously activated XRF-active session.
	1	Length-Checked Compression Support Indicator: 0 not supported 1 supported
	2– 7	Reserved
3 – n		Session correlation
3		Length of session correlator
4 – n		Session correlator: a unique binary value used as a related session identifier

Related Session Identifier (X′28′) Control Vector

Note: An HPR-only control vector also uses this control vector key.

Related Session Identifier (X′28′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′28′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – p		Vector Data
2		Session status:
	0	XRF capability: 1 This session is XRF-capable (only value defined)
	1	XRF session status (reserved if bit 0 = 0): 0 This session is XRF-active. 1 This session is XRF-backup.
	2– 7	Reserved
3 – n		A session key, as described in the "Session Keys" discussion in Chapter 9, "Common Fields" *Note:* The following session keys are used: X′15′ Network-Qualified Address Pair session key: PLU and SLU respectively
n + 1 – p		If more than 1 related session is being reported on, a session status byte and a session key are included as defined above (bytes 2 – n) for each additional XRF related session.

Topic Identifier (X′28′) Control Vector

The Topic Identifier control vector identifies the session or protocol traffic to be carried on an RTP connection being activated.

Note: Another control vector also uses this control vector key in a subarea network context.

Topic Identifier (X′28′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′28′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2	0	User-defined indicator: 0 Topic identifier is globally unique and defined as a character string. 1 Topic identifier is user-defined in a COS/TPF (X′2C′) control vector.
	1– 7	Reserved

Topic Identifier (X′28′) Control Vector

Byte	Bit	Content
3 – n		Topic identifier: • For a globally unique topic identifier (byte 2, bit 0 = 0), one of the following character strings: – CPSVCMG (for CP-CP session traffic) – RSETUP (for route-setup protocol traffic) • For a user-defined topic identifier (byte 2, bit 0 = 1), assigned for connections carrying LU-LU traffic: COS/TPF (X′2C′) control vector, identifying the appropriate COS and transmission priority for all LU-LU sessions using the RTP connection

Session State Data (X′29′) Control Vector

Session State Data (X′29′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′29′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – p		Vector Data
2		SWITCH type: same value received on the corresponding SWITCH request unless promoted from conditional to forced by the receiving LU or boundary function
	0– 3	Type: X′0′ reserved X′1′ conditional X′2′ forced: the state change is immediate X′3′ XRF-active session error: the boundary function promoted the conditional switch request to forced because of an error on the XRF-active session.
	4– 7	State change (same value received on the corresponding SWITCH request): X′0′ reserved X′1′ XRF-backup: the current XRF-active PLU is ready to become a XRF-backup PLU. X′2′ XRF-active: the current XRF-backup PLU is ready to become XRF-active PLU. *Note:* The value X′21′ in byte 2 is reserved. X′22′ The session was switched from XRF-backup to XRF-active (only value defined) X′23′ The session which was suspended due to an application failure has been resumed per the request of the application.

Session State Data (X′29′) Control Vector

Byte	Bit	Content
3		Data flow indicators:
	0	The last request or response was: 0 sent PLU-to-SLU 1 sent SLU-to-PLU
	1	The last request or response was: 0 normal-flow 1 expedited-flow
	2	The last PIU was: 0 a request 1 a response
	3	Expedited response required from the SLU: 0 if an expedited request has been sent to the SLU, then a subsequent expedited response has been sent to the PLU 1 an expedited request has been sent to the SLU and the subsequent expedited response has not been sent to the PLU
	4	Expedited response required from the PLU: 0 if an expedited request has been sent to the PLU, then a subsequent expedited response has been sent to the SLU 1 an expedited request has been sent to the PLU and the subsequent expedited response has not been sent to the SLU
	5	Pacing response status: 0 if a pacing request has been sent to the PLU, then a subsequent pacing response has been sent to the SLU by the PLU 1 a pacing request was sent to the PLU and the subsequent pacing response was sent to the SLU by the boundary function
	6–7	Reserved
4		
	0	Extended last normal request flow indicator 0 extended fields are not present 1 extended PLU-to-SLU and SLU-to-PLU last normal request flow fields are present in bytes 91-98 and 99-106 respectively
	1–7	Reserved
Information about PLU-to-SLU Normal-Flow:		
5–9		Last normal-flow (BC,•EC) request sent PLU-to-SLU
5–6		Sequence number (from the TH) associated with the following RH
7–9		RH of the last first-in-chain or only-in-chain (BC,•EC) request sent PLU-to-SLU
10–19		Last normal-flow request sent PLU-to-SLU
10–11		Last normal-flow request sequence number (from the TH) sent PLU-to-SLU
12–14		RH associated with the following RU
15–19		First 5 bytes of the last normal-flow request RU sent PLU-to-SLU
20–28		Last normal-flow response sent PLU-to-SLU
20–21		Sequence number (from the TH) associated with the following RH
22–23		Bytes 0 and 1 of the RH associated with the following response
24–28		First 5 bytes of the last normal-flow response RU sent PLU-to-SLU
Information about PLU-to-SLU Expedited-Flow:		
29–38		Last expedited-flow request sent PLU-to-SLU

Session State Data (X′29′) Control Vector

Byte	Bit	Content
29–30		Last expedited-flow request sequence number (from the TH) sent PLU-to-SLU
31–33		RH associated with the following RU
34–38		First 5 bytes of the last-expedited request RU sent PLU-to-SLU
39–47		Last expedited-flow response sent PLU-to-SLU
39–40		Last expedited-flow response sequence number (from the TH) sent PLU-to-SLU
41–42		Bytes 0 and 1 of the RH associated with the following response
43–47		First 5 bytes of the last-expedited response RU sent PLU-to-SLU
Information about SLU-to-PLU Normal-Flow:		
48–52		Last normal-flow (BC,•EC) request sent SLU-to-PLU
48–49		Sequence number (from the TH) associated with the following RH
50–52		RH of the last first-in-chain or only-in-chain (BC,•EC) request sent SLU-to-PLU
53–62		Last normal-flow request sent SLU-to-PLU
53–54		Last normal-flow request sequence number (from the TH) sent SLU-to-PLU
55–57		RH associated with the following RU
58–62		First 5 bytes of the last normal-flow request RU sent SLU-to-PLU
63–71		Last normal-flow response sent SLU-to-PLU
63–64		Sequence number (from the TH) associated with the following RH
65–66		Bytes 0 and 1 of the RH associated with the following response
67–71		First 5 bytes of the last normal-flow response RU sent PLU-to-SLU
Information about SLU-to-PLU Expedited Flow:		
72–81		Last expedited flow request sent SLU-to-PLU
72–73		Last expedited-flow request sequence number (from the TH) sent SLU-to-PLU
74–76		RH associated with the following RU
77–81		First 5 bytes of the last expedited-flow request RU sent SLU-to-PLU
82–90		Last expedited-flow response sent SLU-to-PLU
82–83		Last expedited-flow response sequence number (from the TH) sent SLU-to-PLU)
84–85		Bytes 0 and 1 of the RH associated with the following response
86–90		First 5 bytes of the last expedited-flow response RU sent SLU-to-PLU
91–98		First 8 bytes of the last normal flow request RU sent PLU-to-SLU (conditionally present)
99–106		First 8 bytes of the last normal flow request RU sent SLU-to-PLU (conditionally present)
		Note: A value of X′FF′ or X′00′ in the first byte of the above RH fields indicates that a request or response, respectively, has not yet been received by the boundary function for that flow (normal or expedited) and the related fields are to be ignored (i.e., if byte 74 contains X′FF′, the contents of bytes 72–81 are to be ignored). Also, bytes 10–14 and 53–57 may be set/reset as a result of the boundary function detecting a STSN or CLEAR being sent to the SLU. Bytes 10–11 and 53–54 are set to the value carried in the STSN, or reset to X′00′ by the CLEAR. Bytes 12–14 and 55–57 are loaded with the RH of the STSN or CLEAR.

Session Information (X′2A′) Control Vector

A Session Information control vector is included, once for each session, on BFSESSINFO to report the existence of an independent LU′s sessions to an SSCP taking over support services for the LU. It is also included on RSP(ACTLU) and SESSST to report the existence of a dependent LU′s session to an SSCP taking over control of the LU.

Session Information (X′2A′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′2A′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2	0	LU role: 0 The subject LU is the SLU in this session. 1 The subject LU is the PLU in this session.
	1– 7	Reserved
3 – n		Control vectors, as described in detail elsewhere in this chapter. *Note:* The following control vectors may be included; they are parsed according to parsing rule KL (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′0E′ Network Name control vector: the network-qualified name of the partner LU for the reported session (always present) X′15′ Fully Qualified Network Address Pair control vector: PLU and SLU, respectively (present except for an LU attached to a dependent LU requester) X′1E′ VR-ER Mapping Data control vector (present except for an LU attached to a dependent LU requester) X′23′ Local-Form Session Identifier control vector X′60′ Fully Qualified PCID control vector (present when present in the corresponding BIND)

Route Selection (X′2B′) Control Vector

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

Route Selection (X′2B′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′2B′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data

Route Selection (X′2B′) Control Vector

Byte	Bit	Content
2		Maximum hop count: the number, in binary, of TG Descriptor or Network Name control vectors in the Route Selection control vector
3		Current hop count: the index, in binary, of the last TG Descriptor control vector that was processed; the index divides the traversed from the to-be-traversed portions of the path and, thus, points to the next hop to be traversed *Note:* When the values of the Maximum Hop Count field and the Current Hop Count field are equal, all nodes specified in the control vectors have processed the RU
4 – n		Control vectors, the details of which are described elsewhere in this chapter or immediately following this control vector *Note:* The following control vectors may be included; they are parsed according to parsing rule LT. X′0E′ Network Name control vector: one for each control point (Type = X′F4′) on the procedure path (present when the RSCV is carried on Locate) X′46′ TG Descriptor control vector: one for each TG on the session path (present when the RSCV is carried on BIND or RSP(BIND), or on a reply CD-Initiate from a network node server to its client end node) X′80′ RSCV Descriptor control vector (present when the RSCV is carried on BIND, RSP(BIND), or CD-Initiate and a T5 node, a branch network node, or an extended border node in the session path recognizes that the RSCV does not describe the complete session path) *Note:* When present, control vector X′80′ follows immediately after the last control vector X′46′ in the RSCV.

RSCV Descriptor (X′80′) Route Selection Control Vector

The RSCV Descriptor control vector provides additional information regarding the route described in the RSCV.

RSCV Descriptor (X′80′) Route Selection Control Vector

Byte	Bit	Content
0– 1		Control vector header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Control Vector Data

RSCV Descriptor (X′80′) Route Selection Control Vector

Byte	Bit	Content
2		Route Characteristics
2	0	LEN connection indicator: 0 This route does not traverse a LEN connection. 1 This route traverses a LEN connection not described in the RSCV.
	1	Partial RSCV indicator (set only by an interchange node; otherwise, reserved): 0 The portion of this RSCV for which the NNS(PLU) is responsible describes the complete session path. 1 This RSCV describes an incomplete portion of a session path through one or more connected contiguous APPN subnetworks (i.e., no LEN nodes have been traversed).
	2	Border node indicator: 0 The node identified in the field indicating the origin node′s network-qualified CP name is not a border node. 1 The node identified in the field indicating the origin node′s network-qualified CP name is a border node.
	3–7	Reserved
3		Length (0 or 3–17), in binary, of the origin node′s network-qualified CP name
4 – n		Origin node′s network-qualified CP name — identifies the CP at the origin of this RSCV (present only when different from the CP of the PLU identified in the type-X′F3′ Network Name [X′0E′] control vector carried in the BIND): a 3- to 17-byte name consisting of a 1- to 8-byte type-1134 symbol-string network identifier followed by a period and a 1- to 8-byte type-1134 symbol-string name. The network-qualified name appears, for example, as follows: NETID.NAME, with optional (but not significant) trailing, but no imbedded, space (X′40′) characters. As noted in Appendix A, “SNA Character Sets and Symbol-String Types,” implementation usage constrains the leading character of the identifier and of the name to be alphabetic.

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 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2	0–4	Reserved
	5	Network priority indicator: 0 PIUs for this session flow at the priority specified in the Transmission Priority field (bits 6–7). 1 PIUs for this session flow at network priority, which is the highest transmission priority.
	6–7	Transmission priority (reserved if byte 2, bit 5 = 1): 00 low priority 01 medium priority 10 high priority 11 reserved
3		Length, in binary, of COS Name field

COS/TPF (X′2C′) Control Vector

Byte	Bit	Content
4 – n		COS name: 0 to 8 type-1134 symbol-string characters with optional (but not significant) trailing space (X′40′) characters (*Note*: Some SNA-defined COS names use the prefix X′7B′, which is not in the type-1134 character set but does appear in this field as the first character of the SNA-defined COS name.)

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 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Length, in binary, of Mode Name field
3 – n		Mode name: 0 to 8 type-1134 symbol-string characters with optional (but not significant) trailing space (X′40′) characters

LU Definition (X′2F′) Control Vector

The LU Definition control vector contains data that the receiving LU can use to determine the relevant characteristics of the sending LU.

LU Definition (X′2F′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′2F′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data *Note:* At least one of the following subfields (described following this control vector) is always present; they are parsed according to parsing rule LT. X′80′ Additional Mode Characteristics subfield (only one may be present) X′81′ Model Name subfield (only one may be present) X′82′ Associated LU subfield (more than one may be present)

Additional Mode Characteristics (X′80′) LU Definition Subfield

Additional Mode Characteristics (X′80′) LU Definition Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′80′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2		Characteristics
2	0	Type of terminal: 0 keyboard and printer 1 keyboard and display
	1–7	Reserved

Model Name (X′81′) LU Definition Subfield

This subfield carries the name of a model LU definition known to the PLU on a session to be activated. The PLU uses the named model to dynamically build a control-block representation of the SLU during logon processing.

Model Name (X′81′) LU Definition Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′81′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – m		Model name: a 1- to 8-byte type-A symbol-string giving the name of a model LU definition

Associated LU (X′82′) LU Definition Subfield

This subfield carries data about an LU that is not involved in a session to be activated but that the PLU on that session should consider logically related to the SLU.

Associated LU (X′82′) LU Definition Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′82′ Length (m+1), in binary, of Associated LU subfield (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – m		Subfield Data
2		Type of associated LU: X′01′ primary printer X′02′ alternate printer

Associated LU (X′82′) LU Definition Subfield

Byte	Bit	Content
3	0	Name translation: 0 Translation has not occurred for this name. 1 Translation has occurred for this name.
	1– 7	Reserved
4		Length (3–17), in binary, of network-qualified name of associated LU
5 – m		Network-qualified name of associated LU: a maximum-17-byte name consisting of a 1- to 8-byte qualifier concatenated with a period to a 1- to 8-byte name, both type-1134 symbol-strings. The network-qualified name appears, for example, as: NETID.NAME, with no imbedded space (X′40′) characters and with optional (but not significant) trailing space characters

Assign LU Characteristics (X′30′) Control Vector

Assign LU Characteristics (X′30′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′30′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2– 3		Reserved session resources: the number, in binary, of session resources the boundary function is to reserve for the use of the LU identified in the RU
4	0	Route extension (REX) stage pacing parameters: 0 Adaptive session-level pacing is allowed on the REX stage. 1 Fixed session-level pacing is to be used on the REX stage.
	1	Reserved
	2– 7	Pacing window size: the pacing window size for fixed pacing if bit 0 of this byte is *on*, or the batch-mode pacing window size if bit 0 of this byte is *off*.
5	0	Subarea stage pacing parameters: 0 Adaptive session-level pacing is allowed on the subarea stage. 1 Fixed session-level pacing is to be used on the subarea stage.
	1	Reserved
	2– 7	Pacing window size: the pacing window size for fixed pacing if bit 0 of this byte is *on*, or the batch-mode pacing window size if bit 0 of this byte is *off*.
6– 7 (= n)		Maximum number of simultaneous LU-LU sessions in which this LU can participate. If this value is 0, the node receiving this control vector uses its locally defined default value.

BIND Image (X′31′) Control Vector

BIND Image (X′31′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′31′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2 – n		Bytes 1 – r (i.e., through the SLU Name field) of the BIND for the session being activated *Note:* This control vector normally contains bytes 1 – r of BIND, but the number of bytes present may be qualified in the specific descriptions of the RUs or GDS variables that carry it.

Short-Hold Mode (X′32′) Control Vector

Short-Hold Mode (X′32′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′32′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – q		Vector Data
2		Reserved
3 – n		Dial Digits of XID Sender
3		Number, in binary, of dial digits
4 – n		Dial digits: a string of packed (2 hex-digits per byte) digits *Note:* For an odd number of dial digits, the low-order half-byte of the low-order byte contains X′C′.
n + 1 – p		Connection Identifier
n + 1		Length, in binary, of connection identifier (values 0 to 8 are valid)
n + 2 – p		Connection identifier: an implementation-defined value used to quickly associate an incoming call for a short-hold mode reconnection with the proper link station. During the link activation process, each node sets this field to a value of its own choosing and sends it to the adjacent node on the link activation XID. During a short-hold mode call reconnection, each node (e.g., in a nonactivation XID) sets this field to the connection identifier it received from its partner during the link activation process.
p + 1 – q		Short-Hold Mode (SHM) Additional Information
p + 1		Length, in binary, of SHM additional information (values 0 to 8 are valid; currently set to 0)
p + 2 – q		SHM additional information (reserved for future use; currently omitted)

ENCP Search Control (X′33′) Control Vector

The ENCP Search Control control vector is sent in the CP Capabilities GDS variable that is sent from an end node to its network node server. It specifies whether the network node should search the end node or not for the specified resource type during a distributed network search.

ENCP Search Control (X′33′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X′33′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2	0	Search status indicator: 0 Do not search the sending ENCP for resources of the type indicated in ENCP Resource Type field (bytes 4–5) unless there is a directory entry for the resource in the network directory database. 1 Search the sending ENCP when needed for resources of the type indicated in ENCP Resource Type field (bytes 4–5).
	1–7	Reserved
3		Reserved
4–5 (=n)		ENCP resource type: X′00F3′ logical unit (only value defined)

LU Definition Override (X′34′) Control Vector

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

LU Definition Override (X′34′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′34′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Length, in binary, of model name (0 if none)
3 – f		Model name (omitted if Length of Model Name = 0, in which case the next field, Length of Primary Printer Name, immediately follows Length of Model Name): a 1- to 8-byte type-A symbol-string giving the name of the model definition that the PLU should use for the SLU that is logging on; or, alternatively, an all-space (X′40′) string representing a null name
f+1		Length, in binary, of primary printer name (0 if none)

LU Definition Override (X′34′) Control Vector

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

Extended Sense Data (X′35′) Control Vector

Extended Sense Data (X′35′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′35′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – p		Vector Data
2– 5		Sense data
Note: *The shorter abbreviated form (now retired) of the control vector ends here.*		
6 – p		Extended Sense Information
6	0	RU information included: 0 RU information not included (bits 1– 2 set to 00 and bytes 8 – m are not included) 1 RU information included (see bytes 8 – m below)
	1– 2	RU category of the RU in error (reserved when bit 0 = 0): 00 FMD 01 NC 10 DFC 11 SC
	3	FMD message-unit type (reserved when RU category is not FMD): 0 FMD message unit is not a GDS variable. 1 FMD message unit is a GDS variable (only value used on Locate).
	4	Generator of Extended Sense Data control vector (reserved when Termination Procedure Origin Name field not present): 0 the termination procedure origin (only value used on Locate) 1 a node other than the termination procedure origin
	5	Contents of Termination Procedure Origin Name field (reserved when Termination Procedure Origin Name field not present): 0 termination procedure origin name (only value used on Locate) 1 name of node other than termination procedure origin, as described below; termination procedure origin name not known
	6– 7	Reserved

Extended Sense Data (X′35′) Control Vector

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

Directory Error (X′36′) Control Vector

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

Directory Error (X′36′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′36′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2– 5 (= n)		Sense data

Directory Entry Correlator (X′37′) Control Vector

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

Directory Entry Correlator (X′37′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′37′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Implementation-defined correlator value

Short-Hold Mode Emulation (X′38′) Control Vector

The Short-Hold Mode Emulation control vector carries information to establish short-hold mode connections through non-X.21 networks.

Short-Hold Mode Emulation (X′38′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′38′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – q		Vector Data
2		Reserved
3 – n		Dial number
3		Length, in binary, of dial number
4 – n		Dial number: string of EBCDIC characters
n + 1 – p		link Connection identifier
n + 1		Length, in binary, of link connection identifier (values 0 to 8 are valid)
n + 2 – p		Link Connection identifier used to associate a short-hold mode call with the proper logical connection
p + 1 – q		Short-Hold Mode Token
p + 1		Length, in binary, of short-hold mode (SHM) token (values 0 to 8 are valid)
p + 2 – q		SHM token: user-given value *Note:* One possible use of this token is to convey information required by the call-clearing algorithm for some tariff structures, such as a timer value.

NCE Instance Identifier (X′39′) Control Vector

The NCE Instance Identifier control vector carries an implementation-defined 4-byte value identifying a particular instance (or activation) of an NCE.

NCE Instance Identifier (X′39′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′39′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2– 5		NCE instance identifier: a 4-byte implementation-defined value

Route Status Data (X′3A′) Control Vector

Route Status Data (X′3A′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′3A′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2	0– 3	Reserved
	4– 7	VRN for the VR tested
3	0– 5	Reserved
	6– 7	Transmission priority field for the VR tested
4		VR status: X′00′ VR is not defined X′01′ VR is in reset state X′02′ activation of the VR is pending notification of the activation of the underlying ER X′03′ an NC-ACTVR was sent to activate the VR, but no RSP(NC-ACTVR) has been received X′04′ an NC-ACTVR was received to activate the VR, but no RSP(NC-ACTVR) has been sent X′05′ an NC-DACTVR(Orderly) has been sent, but no RSP(NC-DACTVR) has been received X′06′ an NC-DACTVR(Orderly) was received, but no RSP(NC-DACTVR) has been sent X′07′ an NC-DACTVR(Forced) was received, but no RSP(NC-DACTVR) has been sent X′08′ an NC-DACTVR(Forced) was sent but no RSP(NC-DACTVR) has been received X′09′ VR is active
5– 7		Reserved
8	0– 3	Reserved
	4– 7	ERN of the ER tested

Route Status Data (X′3A′) Control Vector

Byte	Bit	Content
9		ER status: X′00′ ER is not defined and not currently operative X′01′ ER is defined but not currently operative X′02′ ER is defined and operative, but not currently active X′03′ an NC-ER-ACT was sent, but no NC-ER-ACT-REPLY has been received X′04′ an NC-ER-ACT was received, but no NC-ER-ACT-REPLY has been sent X′05′ an NC-ER-ACT was received and an NC-ER-ACT-REPLY was sent; an NC-ER-ACT was sent, but no NC-ER-ACT-REPLY has been received X′06′ an NC-ER-ACT was received but no ER is defined; should the ER subsequently become defined, an NC-ER-ACT will be sent X′07′ an NC-ER-ACT was received and an NC-ER-ACT-REPLY was sent (no NC-ER-ACT has been sent from this end) X′08′ ER is active and each node on the ER supports ER-VR protocols X′09′ ER is operative but not currently defined X′0A′ ER is active and traverses a node that does not support ER-VR protocols
10–13		Subarea address of the adjacent node through which the ER being tested flows from this node
14		Transmission group number of the TG (to the node identified in bytes 10–13) over which the ER being tested flows from this node

VR Congestion Data (X′3B′) Control Vector

VR Congestion Data (X′3B′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′3B′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2	0–3	Reserved
	4–7	VRN for the VR tested
3	0–5	Reserved
	6–7	Transmission priority field value for the VR tested
4		Reserved
5		Maximum pacing window size
6		Reserved
7		Minimum pacing window size
8		Reserved
9		Current pacing window size
10–11	0–3	Reserved
	4–15	Next VR sequence number to be sent
12–13	0–3	Reserved
	4–15	Next VR sequence number to be received

VR Congestion Data (X′3B′) Control Vector

Byte	Bit	Content
14	0	VR blockage indicator: 0 VR not blocked 1 VR blocked
	1	Extended data indicator: 0 no extended data follows 1 extended data follows *Note:* If bit 1 is 0, bits 2 and 3 are reserved and bytes 15 through 20 are not included.
	2	Withholding VRPRS indicator: 0 VRPRSs are not being withheld 1 VRPRSs are being withheld
	3	Discarding PIU indicator: 0 PIUs are arriving in sequence and not being discarded 1 PIUs are arriving out of sequence and being discarded
	4–7	Reserved
15–16		Contents of an implementation-defined VR control block
17–18		Inbound VR PIU buffer pool threshhold (the number of buffered inbound PIUs at which VRPRS will be withheld)
19–20		Inbound VR PIU buffer pool count (the number of currently buffered inbound PIUs)

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

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

Associated Resource Entry Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X′3C′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2–3		Resource type: X′00F4′ ENCP X′00F6′ NNCP
4 – n		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted) *Note:* The network ID is always present when different from the network ID of the receiver.

Directory Entry (X′3D′) Control Vector

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

Directory Entry (X′3D′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′3D′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2– 3		Resource type: X′00F3′ logical unit X′00F4′ ENCP X′00F6′ NNCP *Note:* The hierarchically highest type may be specified in the case of a merged CP/LU (CP=LU) — i.e., where the identified resource is both a control point and an LU serving end-user sessions. In this case, the duplicative Associative Resource Entry (X′3C′) control vector is omitted. So for NNCP=LU for example, X′00F6′ may be specified here, and no X′3C′ control vector precedes this X′3D′ control vector. For ENCP=LU, X′00F4′ may be specified here, and the X′3C′ control vector that would otherwise (i.e., in the LU≠CP case) precede this control vector to specify the ENCP is omitted.
4 – n		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted) *Note:* The network ID, if omitted, is assumed to be the same as that of the hierarchically related X′3C′ control vector, or, if that is absent, the same as that of the receiver.

Directory Entry Characteristic (X′3E′) Control Vector

The Directory Entry Characteristic (X′3E′) control vector specifies a characteristic of a resource identified in a preceding X′3D′ control vector (or X′3C′ control vector if LU=CP and the X′3D′ control vector is omitted).

Directory Entry Characteristic (X′3E′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′3E′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2 – n		Subfields, as described below. Note: At least one of the following subfields is always present; they are parsed according to the parsing rule LT. X′80′ Directory Entry Stability subfield (used on Find, Found, and Register) X′81′ LU Name Stability subfield (used on Found and Register) X′82′ Subarea Characteristics (used on Find and Found)

Directory Entry Stability (X′80′) Directory Entry Characteristic Subfield

The Directory Entry Stability subfield carries the directory entry stability information.

Directory Entry Stability (X′80′) Directory Entry Characteristic Subfield

Byte	Bit	Content
0–1		Subfield header; Key = X′80′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2		Characteristics
2	0	Availability indicator: 0 The associated resource is not available for the time interval indicated. 1 The associated resource is available for the time interval indicated.
	1	MNPS resource indicator (for special resource handling and recovery purposes, MNPS mobile LUs are identified during Locate operations; this MNPS attribute is retained by directory services for future reference): 0 not an MNPS resource 1 MNPS resource
	2	resource verification required indicator (for special resource handling purposes, some LUs are identified during resource registration for locate operations; this resource verification attribute is retained by directory services for future reference): 0 resource verification not required during RSCV precomputation 1 resource verification required during RSCV precomputation
	3–7	Reserved
3–6 (=m)		Delta time: a 32-bit unsigned binary integer specifying a time interval, in seconds, after the time received, where the maximum value X′FFFFFFFF′ indicates that the availability status indicated in byte 2, bit 0 will continue for approximately 136 years after the time received; X′00000000′ indicates an unstable resource; i.e., the location of the associated resource must be verified with a directed Locate search

LU Name Stability (X′81′) Directory Entry Characteristic Subfield

The LU Name Stability subfield carries the LU name stability information.

LU Name Stability (X′81′) Subfield.

Byte	Bit	Content
0–1		Subfield header; Key = X′81′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2(=m)		Characteristics

LU Name Stability (X′81′) Subfield.

Byte	Bit	Content
2	0–1	LU name stability indicator: 00 The stability is undetermined. 01 The LU name will not change. 10 The LU name may change as a result of a failure; it should be verified with a directed Locate search after an abnormal session termination occurs. 11 The LU name may change frequently; it should be verified with a directed Locate search for every session initiation.
	2–7	Reserved

Subarea Characteristics (X′82′) Directory Entry Characteristic Subfield

The Subarea Characteristics subfield carries subarea related information about the resource.

Subarea Characteristics (X′82′) Subfield.

Byte	Bit	Content
0–1		Subfield header; Key = X′82′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2(=m)		Characteristics
2–5		Subarea number - this information is used to compute more optimal routes through subarea networks.

SSCP(SLU) Capabilities (X′3F′) Control Vector

The SSCP(SLU) Capabilities (X′3F′) control vector is used to indicate the capabilities of an SSCP(SLU) to provide certain functions, such as notification to the SLU of specific error conditions.

SSCP(SLU) Capabilities (X′3F′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X′3F′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2	0	Error notification to SLU: 0 The SLU will be notified if the SSCP(SLU) receives a BINDF containing an indication that a session initiation request was transferred, or "passed," to another PLU. 1 The SLU will not be notified if the SSCP(SLU) receives a BINDF containing an indication that session initiation request was transferred to another PLU. *Note:* This condition is indicated by the inclusion of sense data X′0842FFFF′ in the BINDF.
	1–7	Reserved

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

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

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

Byte	Bit	Content
0– 1		Vector header; Key = X′40′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2– 3		Resource type: X′00F6′ NNCP
4 – n		Resource name: a 1- to 17-byte name consisting of a qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period.

Station Parameters (X′41′) Control Vector

Station Parameters (X′41′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′41′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)

Station Parameters (X′41′) Control Vector

Byte	Bit	Content
2 – n		Vector Data *Note:* The following subfields are optionally included; they are parsed according to parsing rule LT. X′81′ IP Address X′82′ Maximum Transmission Unit X′83′ COMRATE - first subparameter X′84′ COMRATE - second subparameter X′85′ Data Link Connection Identifier X′86′ T1TIMER - first subparameter X′87′ T1TIMER - second subparameter X′88′ T2TIMER - first subparameter X′89′ T2TIMER - second subparameter X′8A′ T2TIMER - third subparameter X′8B′ DYNWIND - first subparameter X′8C′ DYNWIND - second subparameter X′8D′ DYNWIND - third subparameter X′8E′ Virtual Path/Virtual Circuit X′8F′ HPR Queue Limit X′90′ IPQLIM Value X′91′ Delayed Disconnect Timer *Note:* If all the subfields do not fit into one control vector X′41′, that control vector is immediately followed by one or more additional control vector X′41′s that contain only those subfields that would not fit into the first control vector X′41′.

IP Address (X′81′) Station Parameters Subfield

The IP Address subfield is provided when an IP interface is being dynamically added to the system.

IP Address (X′81′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′81′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (IP address of interface being added)

Maximum Transmission Unit (X′82′) Station Parameters Subfield

The maximum transmission unit is optionally provided for each IP interface that is added to the system.

Maximum Transmission Unit (X′82′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′82′ (See “Substructure Encoding/Parsing Rules” on page 9-5)

Maximum Transmission Unit (X′82′) Station Parameters Subfield

Byte	Bit	Content
2 – n		Subfield Data (maximum transmission unit)

First COMRATE Subparameter (X′83′) Station Parameters Subfield

The first COMRATE subparameter provides the communications rate discard eligibility (DE) control information.

First COMRATE Subparameter (X′83′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′83′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (communications rate DE control)

Second COMRATE Subparameter (X′84′) Station Parameters Subfield

The second COMRATE subparameter provides the communications rate transmission priority information.

Second COMRATE Subparameter (X′84′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′84′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (communications rate transmission priority)

Data Link Connection Identifier (X′85′) Station Parameters Subfield

The data link connection identifier subfield provides the DLCI number.

Data Link Connection Identifier (X′85′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′85′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (DLCI)

First T1TIMER Subparameter (X′86′) Station Parameters Subfield

The first T1TIMER subparameter provides the T1 local timer value in tenths of a second.

First T1TIMER Subparameter (X′86′) Station Parameters Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′86′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (T1 local timer value)

Second T1TIMER Subparameter (X′87′) Station Parameters Subfield

The second T1TIMER subparameter provides the T1 remote timer value in tenths of a second.

Second T1TIMER Subparameter (X′87′) Station Parameters Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′87′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (T1 remote timer value)

First T2TIMER Subparameter (X′88′) Station Parameters Subfield

The first T2TIMER subparameter provides the T2 local timer value in tenths of a second.

First T2TIMER Subparameter (X′88′) Station Parameters Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′88′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (T2 local timer value)

Second T2TIMER Subparameter (X′89′) Station Parameters Subfield

The second T2TIMER subparameter provides the T2 remote timer value in tenths of a second.

Second T2TIMER Subparameter (X′89′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′89′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (T2 remote timer value)

Third T2TIMER Subparameter (X′8A′) Station Parameters Subfield

The third T2TIMER subparameter provides the T2 timer N3 value.

Third T2TIMER Subparameter (X′8A′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′8A′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (T2 timer N3 value)

First DYNWIND Subparameter (X′8B′) Station Parameters Subfield

The first DYNWIND subparameter provides the DYNWIND Nw value.

First DYNWIND Subparameter (X′8B′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′8B′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (DYNWIND Nw value)

Second DYNWIND Subparameter (X′8C′) Station Parameters Subfield

The second DYNWIND subparameter provides the DYNWIND Dw value.

Second DYNWIND Subparameter (X′8C′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′8C′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (DYNWIND Dw value)

Third DYNWIND Subparameter (X′8D′) Station Parameters Subfield

The third DYNWIND subparameter provides the DYNWIND Dwc value.

Third DYNWIND Subparameter (X′8D′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′8D′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (DYNWIND Dwc value)

Virtual Path/Virtual Circuit (X′8E′) Station Parameters Subfield

The virtual path/virtual circuit subfield provides the virtual path/virtual circuit (VPVC) value.

Virtual Path/Virtual Circuit (X′8E′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′8E′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (VPVC value)

HPR Queue Limit (X′8F′) Station Parameters Subfield

The HPR queue limit subfield provides the HPR queue limit (HPRQLIM) value.

HPR Queue Limit (X′8F′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′8F′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (HPRQLIM value)

IPQLIM Value (X′90′) Station Parameters Subfield

The IPQLIM subfield provides the IP queue limit (IPQLIM) value.

IPQLIM value (X′90′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′90′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (IPQLIM value)

Delayed Disconnect Time Value (X′91′) Station Parameters Subfield

The delayed disconnect time value subfield provides the delayed disconnect time value in seconds.

Delayed Disconnect Time Value (X′91′) Station Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′91′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (delayed disconnect time value)

Dynamic Path Update Data (X′42′) Control Vector

The Dynamic Path Update Data control vector carries data that is used by a T4|5 node to update its routing tables.

Dynamic Path Update Data (X′42′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′42′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data *Note:* The following subfields may be included; they are parsed according to parsing rule LT. X′80′ Node Identifier Data (single subfield always present) X′81′ Explicit Route Data (multiple subfields may be present—one for each ERN entry to be added, replaced, or deleted) X′82′ Virtual Route Data (present if the VR to ER mapping is being changed; multiple subfields may be present—one for each VRN entry to be added or replaced) X′83′ Virtual Route Window Size Data (present if the VR TPF window sizes are being changed; multiple subfields may be present—one for each VRN transmission priority entry being added or replaced). *Note:* The Node Identifier Data (X′80′) is always the first subfield. It is followed in order by all the Explicit Route Data (X′81′) subfields, all the Virtual Route Data (X′82′) subfields, and all the Virtual Route Window Size Data (X′83′) subfields. When multiple Dynamic Path Update Data (X′42′) control vectors are included in a SETCV RU, the second control vector will be a continuation of the first, and so forth. This means, for example, that only the first control vector will include a Node Identifier Data subfield.

Node Identifier Data (X′80′) Dynamic Path Update Data Subfield

The Node Identifier Data subfield identifies the destination node of the routes being changed. The information it carries is used in conjunction with the information in the Explicit Route Data, Virtual Route Data, and Virtual Route Window Size Data subfields to make changes to the routing tables.

Node Identifier Data (X′80′) Dynamic Path Update Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
2		Type field for the Dynamic Path Update Data (X′42′) control vector: X′00′ request X′20′ invalid network ID specified X′21′ control block allocation failed X′22′ invalid Explicit Route Data, Virtual Route Data, or Virtual Route Window Size Data subfield included in the Dynamic Path Update Data control vector X′23′ Destination Subarea Address (DSA) is greater than the defined subarea address limit *Note:* The type code X′00′ flows on a request. The type codes X′20′, X′21′, X′22′, and X′23′ flow on a response.
3– 6		Subarea address of the destination subarea (DSA) for which the routing table is being modified
7		Length, in binary, of Network Identifier field: values of 1 through 8 are valid
8 – n		Network identifier of the DSA

Explicit Route Data (X′81′) Dynamic Path Update Data Subfield

The Explicit Route Data subfield carries the information required to update or change the ERN-to-TG mapping for a given ERN. Information is optionally included to allow the TG block to be built.

Explicit Route Data (X′81′) Dynamic Path Update Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′81′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data

Explicit Route Data (X′81′) Dynamic Path Update Data Subfield

Byte	Bit	Content
2		Type: X′00′ add or replace the ERN definition X′01′ delete the ERN definition X′20′ the ER is operative X′21′ control block allocation failed X′22′ adjacent subarea address is greater than the defined subarea address limit X′23′ Explicit Route Number (ERN) is greater than the defined ERN limit *Note:* The type codes X′00′ and X′01′ flow on a request. The type codes X′20′, X′21′, X′22′, and X′23′ flow on a response.
3	0–3	Reserved
	4–7	ERN of the explicit route to be changed: a binary value 0–15
4		Length, in binary, of the following transmission group information: transmission group number, adjacent subarea address, and the optional TG block fields: values of 0, 5 and 17 are valid
5		Transmission group number (TGN): a binary value 1–255
6–9		Adjacent subarea address
10–21		Optional TG block fields (all present or none present)
10–12	0–4	Reserved
	5–23	Low-priority threshold: a binary value 0–524,287
13–15	0–4	Reserved
	5–23	Medium-priority threshold: a binary value 0–524,287
16–18	0–4	Reserved
	5–23	High-priority threshold: a binary value 0–524,287
19–21	0–4	Reserved
	5–23	Total threshold: a binary value 0–524,287

Virtual Route Data (X′82′) Dynamic Path Update Data Subfield

The Virtual Route Data subfield carries the information required to update the VR-to-ER mapping information for a given ER-VR pair.

Virtual Route Data (X′82′) Dynamic Path Update Data Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′82′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2–3		Subfield Data
2		Type: X′00′ add or replace the VRN mapping X′20′ no corresponding ER subfield (add or replace) successfully processed X′21′ the VRN is active on a different ER X′22′ Explicit Route Number (ERN) is greater than the defined ERN limit *Note:* The type code X′00′ flows on a request. The type codes X′20′, X′21′, and X′22′ flow on a response.

Virtual Route Data (X′82′) Dynamic Path Update Data Subfield

Byte	Bit	Content
3	0–3	Virtual route number (VRN): a binary value 0–7
	4–7	Explicit route number (ERN): a binary value 0–15

Virtual Route Window Size Data (X′83′) Dynamic Path Update Data Subfield

The Virtual Route Window Size Data subfield carries the window sizes of the VR for the specified transmission priority.

Virtual Route Window Size Data (X′83′) Dynamic Path Update Data Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′83′ (See “Substructure Encoding/Parsing Rules” on page 9-5)
2–5		Subfield Data
2		Type: X′00′ add or replace the VR TPF window sizes X′20′ no corresponding VR subfield successfully processed X′21′ the VRN/TPF is already active X′22′ control block allocation failed *Note:* The type code X′00′ flows on a request. The type codes X′20′, X′21′, and X′22′ flow on a response.
3	0–3	Virtual route number (VRN): a binary value 0–7
	4–7	Transmission priority field (TPF): a binary value 0–2
4		Minimum window size: a binary value 0–255
5		Maximum window size: a binary value 0–255 *Note:* A value of 0 is not a valid window size. If minimum window size and maximum window size are both coded 0, then default values for window size should be used.

Extended SDLC Station (X′43′) Control Vector

Extended SDLC Station (X′43′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′43′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2	0–4	Reserved
	5–7	Node type identifier: 100 T1 node 010 T2 node 001 T4|5 node

Extended SDLC Station (X′43′) Control Vector

Byte	Bit	Content
3		Type modifier:
	0	TS profile usage (reserved except when byte 2 indicates T1 node): 0 ¬ TS profile 2 1 TS profile 2
	1	0 Discontinue link-level contact with adjacent T1|2 node if the subarea node initiates an auto network shutdown procedure for the SSCP controlling that T1|2 node or, in the case of a switched subarea link, discontinue link-level contact with adjacent T4|5 node if auto network shutdown procedure is initiated. 1 Continue link-level contact with adjacent T1|2 node if the subarea node initiates an auto network shutdown procedure for the SSCP controlling that T1|2 node or, in the case of a switched subarea link, continue link-level contact with adjacent T4|5 node if an auto network shutdown procedure is initiated.
	2	0 Use SNRM polling for the secondary station. 1 Use null XID polling for the secondary station.
	3	0 Modem test support for the secondary station is as specified during system definition for the link. 1 Modem test is not supported for the secondary station.
	4	Data mode: 0 half-duplex 1 full-duplex
	5	LPDA-2 Information field (bytes 17–18) validity indicator: 0 Bytes 17–18 are valid. 1 Bytes 17–18 are to be ignored.
	6	Retry Control Information field (bytes 7–8) validity indicator: 0 Bytes 7–8 are to be ignored. 1 Bytes 7–8 are valid.
	7	Reserved
4		Frame send control value: • For SDLC, the maximum number of frames (or BTUs) that the node can send before it must receive an acknowledgment from its partner link station. This value has an implied modulus for the SDLC send and receive counts: less than 8 implies a modulus of 8, while 8 or greater implies a modulus of 128. • For frame-relay switching equipment (FRSE), the maximum send queue size that, when reached, causes the node to discard frames, without forwarding, until room once again exists in the queue. Reaching half this maximum queue size causes the node to set the appropriate congestion indicators in available frames to inform its partner FRSEs of the congestion condition.
5		Maximum consecutive BTUs sent from the primary station to the specified secondary station without another secondary station on the link being polled or being sent BTUs
6		Error retry indicator: X′00′ no immediate retry X′10′ immediate retry
7–8		Retry Control Information
7		Length of pause between retry sequences
8		Maximum number of retry sequences
9–10		Byte count of maximum BTU size permitted to be sent to the adjacent link station represented by the specified secondary station
11–14		Threshold Control Information
11–12		Total number of transmissions

Extended SDLC Station (X′43′) Control Vector

Byte	Bit	Content
13–14		Total number of error retries
15–16		Average number of bytes expected when the station is polled
17–18		LPDA-2 Information
17		Number (1 or 2), in binary, of link connection segments
18		Local Modem Addresses
	0–3	Local modem address of link segment 1: X′0′ – X′F′ are valid addresses.
	4–7	Local modem address of link segment 2: X′0′ – X′F′ are valid addresses; a value of 0 is used if only one link segment exists.
19		Group poll address (present when byte 5, bit 4 in control vector X′0B′ was set by the T4 node and the SSCP has APPN capability; otherwise, not present): nonzero value to identify a group poll address; 0 if no group poll address exists
Note:		When this control vector flows on an RNAA, it may be abbreviated (indicated by the length field); in this case, bytes 7–8 are reserved, and byte 19 or bytes 11–19 are omitted. When this control vector is sent on a SETCV, byte 3, bits 2–7, and bytes 5–8 are reserved.

Node Descriptor (X′44′) Control Vector

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

Node Descriptor (X′44′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′44′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – p		Vector Data: All implementations can accept and forward the Node Descriptor (X′44′) control vector in its entirety
2		Length (1–17), in binary, of the node′s network-qualified CP name (some back-level nodes omit the network ID qualifier)
3 – m		Network-qualified CP name: a 2-part name consisting of a qualifier concatenated by a CP name, each part being a 1- to 8-byte type-1134 symbol string
m + 1		Length (0 is the only value currently defined), in binary, of the following reserved field
m + 2 – n		Reserved (variable length)
n + 1 (= p)	0	Connection network indicator: 0 The Network-Qualified CP Name field does not identify a connection network. 1 The Network-Qualified CP Name field does identify a connection network.
	1–7	Reserved

Node Characteristics (X′45′) Control Vector

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

Node Characteristics (X′45′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′45′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data: All implementations can accept and forward the Node Characteristics (X′45′) control vector in its entirety. *Note:* The following subfields may be included. They are parsed according to subfield parsing rule LT. X′80′ Node Type and Status subfield (always present) X′81′ Extended Support Node Characteristics subfield (present when the sending node is a central directory server; otherwise, not currently used)

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

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

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

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		Subfield Data
2– 5		Resource sequence number: a 32-bit binary value used to time stamp each resource update in the topology update. It is always incremented by 2 by the node that creates the resource update. When the low-order bit of the sequence number is set to 1, a node has recognized receiving inconsistent data; that is, more than one update with the same TG descriptor and with the same resource sequence number but with different data has been received by that node. If the RSN for a resource reaches the maximum value, the owner of the resource should rename the resource and reintroduce it into the network. A node may generate a maximum RSN about its own resources of only 2^{32}-2, since it increases RSNs by 2 for these resources. The value 2^{32}-1 is used only when a node indicates that it found another node's resource update using an RSN of 2^{32}-2 to be inconsistent.
6		Route-addition resistance: a binary value in the range 0–255; the greater the value, the less the ability of the node to accept additional routes

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

Byte	Bit	Content
7		Node status (a value of 1 indicates that the condition is true):
	0	Node congested
	1	Intermediate routing resources depleted
	2	Retired
	3	Garbage collection indicator: 0 The node record represented by this control vector has not been marked to be garbage collected (deleted from the topology database) upon the next garbage collection cycle. 1 The node record has been marked to be garbage collected upon the next garbage collection cycle.
	4	Maximum Locate length. Together, bits 4, 6, and 7 (in byte 7) indicate, in 3-bit exponent form, the maximum length Locate message supported by this node on all CP-CP sessions. Bit 4 is the leftmost bit, bit 6 is the middle bit, and bit 7 is the rightmost bit. **3-bit exponent form** is a 3-bit field indicating the number of kilobytes (K bytes) with values 0-7 used as an exponent of 2. Value 0 (B′000′) indicates 1K bytes (default), 1 (B′001′) indicates 2K, 2 (B′010′) indicates 4K, 3 (B′011′) indicates 8K, 4 (B′100′) indicates 16K, 5 (B′101′) indicates 32K, 6 (B′110′) indicates 64K, and 7 (B′111′) indicates 128K.
	5	Quiescing
	6–7	The middle and rightmost bits of the maximum Locate length field (see complete description under byte 7, bit 4).
8		Node Type and Support
	0	Gateway services support: 0 gateway services not supported 1 gateway services supported
	1	Central directory services support: 0 central directory services not supported 1 central directory services supported
	2	Intermediate routing services support: 0 intermediate routing services not supported 1 intermediate routing services supported
	3	Retired (always set to 1)
	4	Branch awareness support (1120): 0 branch awareness not supported 1 branch awareness supported
	5	Reserved
	6–7	Retired (always set to 11)

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

Byte	Bit	Content
9(=n)		Additional Node Support
	0	Peripheral border node support: 0 The node lacks such support. 1 The node has such support.
	1	Interchange node support: 0 The node lacks such support. 1 The node has such support.
	2	Extended border node support: 0 The node lacks such support. 1 The node has such support.
	3–4	HPR support (used solely for network management purposes): 00 none (only value used for virtual routing nodes) 01 HPR base (1400) support 10 RTP (1401) option set support 11 Control Flows Over RTP (1402) option set support
	5	Interior border node support: 0 The node lacks such support. 1 The node has such support.
	6–7	Reserved

Extended Support (X′81′) Node Characteristics Subfield

The Node Characteristics Extended Support subfield carries the node extended support data that may be reported as part of a topology update.

Extended Support (X′81′) Node Characteristics Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′81′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data (Reserved bytes may be truncated and the length field adjusted accordingly)
2		Directory Services Extended Support
	0	Central directory server support: 0 Node is not a central directory server. 1 Node is a central directory server.
	1–7	Reserved
3(=n)		Reserved

TG Descriptor (X′46′) Control Vector

The TG Descriptor control vector identifies a transmission group (TG). All implementations, as a base requirement, always accept and forward the TG Descriptor (X′46′) control vector in its entirety.

TG Descriptor (X′46′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′46′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data *Note:* The following subfields (described following this control vector) may be included. They are parsed according to subfield parsing rule LT. X′80′ TG Identifier subfield (always present except when the control vector is included in a FID2 Encapsulation GDS variable or when a Node-Specific TG Identifier subfield is included) X′81′ Connection Network TG Numbers subfield (present on XID3 when the sending node is activating a TG through an ATM or frame-relay connection network; optionally present on XID3 when the sending node is activating a TG through an IP connection network) X′82′ DLC Signaling Information subfield (present only when the TG is connected to a token-ring or ethernet connection network, except not present on XID3) X′83′ Real Partner CP Name subfield (present in a CD-Initiate in a Locate reply [or request] when a border node modifies the associated resource hierarchy such that the CP(DLU) [or CP(OLU)] is not adjacent to the NNS(DLU) [or NNS(OLU)]; or in an RSCV when an NNS(OLU) calculates a route that includes a TG vector carrying it): when present, used in preference to the TG-partner node's CP name in the TG Identifier (X′80′) subfield X′85′ Composite Route Selection subfield (present in an RSCV when an intersubnet route has traversed an HPR border node and may be present in an RSCV or an endpoint TG vector to represent a branch downlink.) X′88′ TG Identifier Extension subfield (present on an endpoint TG vector or in a session RSCV, to identify a branch uplink) X′91′ DLC Signaling Type subfield (present in the FID2 Encapsulation GDS variable when DLC signaling information is to be conveyed for DLUR use; present for ATM, frame-relay, and IP connection network TGs in Route Selection [X′2B′] control vector, Cross-Domain Initiate GDS variable, and Topology Database Update GDS variable). When present, additional DLC Signaling Information subfields (in the range X′92′ through X′F0′) follow this subfield, depending upon the DLC type indicated in the X′91′ subfield, as described below. More than one such series of DLC signaling information, consisting of one DLC Signaling Type (X′91′) subfield and its following DLC Signaling Information (X′92′ – X′F0′) subfields, may be included in the TG Descriptor (X′46′) control vector. (Note: Each of the subfields following the X′91′ subfield is called a "DLC Signaling Information" subfield. However, when referred to in the X′91′ subfield, each subfield is identified by its content.)

TG Identifier (X′80′) TG Descriptor Subfield

TG Identifier (X′80′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
2		TG number: the binary integer negotiated during XID exchange to represent the TG to the partner node on the TG
3		Length, in binary, of TG-partner node′s network-qualified CP name; values 1 to 17 are valid, except set to 0 when carried on XID3 and byte m+1, bit 0 (below) is set to 0 (i.e., not a connection network) or bytes m+2 – m+5 are present having a nonzero value
4 – m		TG-partner node′s network-qualified CP name (i.e., the name of the CP in the node at the opposite end of the TG). When this subfield is carried in an RSCV or in a Composite Route Selection (X′85′) TG Descriptor subfield, the net ID qualifier may be omitted from this field if the partner node′s CP name is qualified by the same net ID as the TG origin node (to conserve on RSCV total length).

TG Identifier (X′80′) TG Descriptor Subfield

Byte	Bit	Content
m+1	0	Connection network indicator: 0 The TG-Partner Node′s Network-Qualified CP Name field does not identify a connection network (e.g., a local-area network). 1 The TG-Partner Node′s Network-Qualified CP Name field does identify a connection network; in this case, bytes 4 – m contain the CP name representing the virtual routing node.
	1	Additional configuration information indicator (used only when this TG Descriptor is carried in an RSCV or a TDU; otherwise, reserved) — indicates that the CP identified in the TG-Partner Node′s Network-Qualified CP Name field may have additional configuration information, used for network management, associated with this TG. • When this control vector is carried in an RSCV, the additional information (e.g., subarea routing information within a composite network node) pertains to the session path described in the RSCV. • When this control vector is carried in a TDU, the additional information (e.g., additional network topology beyond a branch TG) pertains to network topology. 0 Additional information is not associated with this TG. 1 Additional information may be associated with this TG.
	2	HPR supported indicator: 0 HPR not supported over this TG 1 HPR supported over this TG — meaning the TG connects two (real) nodes supporting HPR over this TG, or connects an HPR-supporting node to a virtual-routing node
	3– 4	TG type indicator: 00 APPN TG or boundary-function-based TG 01 interchange TG 10 virtual-route-based TG
	5	Intersubnetwork TG indicator: 0 This TG is not an intersubnetwork TG. 1 This TG is an intersubnetwork TG (defines a border between subnetworks).
	6	Extended border node indicator: 0 The local (sending) node is not an extended border node. 1 The local (sending) node is an extended border node.
	7	RTP (1401) option set supported indicator (reserved if byte m+1, bit 2 = 0): The meaning of this bit depends on the nodes connected by the TG: • The TG connects a real node (EN or NN) and a virtual routing node: In this case, the bit indicates whether the real node supports the RTP option set. • The TG connects two NNs: In this case, the bit indicates whether the NN the TG goes to supports the RTP option set. • The TG connects an EN to another real node, i.e., is reported in a TG vector: In this case, the bit indicates whether the EN reporting the TG vector supports the RTP option set. 0 RTP option set not supported over this TG 1 RTP option set supported over this TG
m+2 – m+5 (=n)		Subarea number (omitted if all 0′s): In TDU GDS variables, if the local node is a subarea-capable node, this field contains the subarea number of the local T4|5 node that owns the link station associated with this TG. If the local node is not a subarea-capable node but the partner node is, this field contains the subarea number of the T4|5 node that owns the link station in the TG-partner node. In the former case, the high-order bit of the subarea number is 1; in the latter, 0. If neither node is subarea-capable, this field is omitted. If carried on XID3, this field contains the subarea address of the sending node (always an interchange node) if it has one, and the high-order bit of the address is always 0.

Connection Network TG Numbers (X′81′) TG Descriptor Subfield

Connection Network TG Numbers (X′81′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′81′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
2		TG number of the TG to the virtual routing node
3		Partner node′s TG number for its TG to the virtual routing node
4– 5 (= n)		Reserved

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

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

Byte	Bit	Content
0– 1		Subfield header; Key=X′82′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		DLC-Specific Data Related to the Connection Network
		Note: The DLC addresses below, no matter which DLC type, are **here** represented most-significant byte first, and most-significant bit first within each byte. This is sometimes referred to as *noncanonical* (or *big-endian*) format.
For Token-Ring (see IBM Token-Ring Network Architecture Reference*):*		
2– 7		MAC address (see **Note** above)
8(=n)		LSAP address (see **Note** above)
For Ethernet (see ANSI/IEEE Standard 802.3 — 1990 Edition*):*		
2– 7		MAC address (see **Note** above)
8(=n)		LSAP address (see **Note** above)

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

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

Byte	Bit	Content
0–1		Subfield header; Key=X′83′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		Subfield Data
2		Length (1–17), in binary, of the TG-partner node's network-qualified CP name
3 – n		Network-qualified CP name of the real TG-partner node: the name of the CP in the node at the opposite end of the TG. This subfield indicates that the name specified in the X′80′ subfield does not reflect the true name of the CP located on the opposite end of the TG.
n + 1	0	Indicates whether an intersubnetwork TG specified in a precomputed RSCV is on the route to the DLUR of the dependent SLU: 0 The intersubnetwork TG specified in the precomputed RSCV may not be on the route to the DLUR. 1 The intersubnetwork TG specified in the precomputed RSCV is on the route to the DLUR.
	1–7	Reserved

Composite Route Selection (X′85′) TG Descriptor Subfield

The Composite Route Selection subfield may be carried in the CD-Initiate GDS variable and BIND to describe the path through multiple subnets or across one or more branch downlinks that a session is to take or has taken.

Composite Route Selection (X′85′) TG Descriptor Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′85′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2		The number, in binary, of TG Descriptor (X′46′) control vectors carried in this Composite Route Selection subfield
3		Reserved
4 – n		Control vectors, as described in the section "Control Vectors" in Chapter 9, "Common Fields" *Note:* The following control vectors may be included; they are parsed according to parsing rule LT. X′46′ TG Descriptor control vector (one or more), the series of which defines the path through one or more HPR nonnative subnets or across one or more branch downlinks toward the destination node; these X′46′ control vectors do not themselves contain X′85′ or X′83′ subfields.

TG Identifier Extension (X′88′) TG Descriptor Subfield

The TG Identifier Extension TG Descriptor subfield, when present, is included within a TG Descriptor (X′46′) control vector, which in turn may be present as a endpoint TG vector on the Cdinit GDS variable, or in a Route Selection (X′2B′) control vector on Cdinit, BIND, or Route Setup.

TG Identifier Extension (X′88′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′88′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
2	0	Branch uplink indicator: 0 This TG is not a branch uplink. 1 This TG is a branch uplink.
	1-7	Reserved

DLC Signaling Type (X′91′) TG Descriptor Subfield

DLC Signaling Type (X′91′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′91′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type of Internal PU (DLUR):		
2– 6 (= n)		DLC or physical interface type (EBCDIC characters): INTPU (for internal PU) *Note: For DLC Type = INTPU, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield; when both are present, each one identifies the same internal PU. Each internal PU to be identified requires its own X′91′ series.* X′92′ Internal PU Identifier: ID Block/ID Number (always present if identifying a different internal PU from that identified by the CP name in another X′91′ series present in the same X′46′ control vector; may be present if a X′93′ subfield is present in the same X′91′ series, in which case it identifies the same internal PU as the CP name in the accompanying X′93′ subfield) X′93′ Internal PU Identifier: CP Name (may be present to identify only one internal PU; any other internal PUs identified in other X′91′ series in the same X′46′ control vector are identified by ID block/ID number only — in other words, in a given X′46′ control vector, only one X′91′ subfield for DLC type = INTPU can have a following CP Name [X′93′] subfield)
For DLC Type of Token-Ring LAN:		
2– 3 (= n)		DLC or physical interface type (EBCDIC characters):

DLC Signaling Type (X′91′) TG Descriptor Subfield

Byte	Bit	Content
		TR (for token-ring — see *IBM Token-Ring Network Architecture Reference*) *Note: For DLC Type = TR, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′92′ Port Identifier (optionally present): product-dependent identifier X′93′ Destination Service Access Point Address (DSAP) (always present) X′94′ Destination MAC Address (always present) X′95′ Source Service Access Point Address (SSAP) (present if multiple parallel links are supported) X′96′ Source MAC Address (present if X′92′ subfield not present; optional otherwise)
For DLC Type of Ethernet (IEEE 802.3) LAN:		
2– 9 (= n)		DLC or physical interface type (EBCDIC characters): ETHERNET (for IEEE 802.3 LAN — see ISO/IEEE 8802.3 documentation) *Note: For DLC Type = ETHERNET, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′92′ Port Identifier (optionally present): product-dependent identifier X′93′ Destination Service Access Point Address (DSAP) (always present) X′94′ Destination MAC Address (always present) X′95′ Source Service Access Point Address (SSAP) (present if multiple parallel links are supported) X′96′ Source MAC Address (present if X′92′ subfield not present; optional otherwise)
For DLC Type of Frame-Relay Permanent Virtual Connection (PVC):		
2– 6 (= n)		DLC or physical interface type (EBCDIC characters): FRPVC (for frame-relay permanent virtual connection} — or FR, now retired *Note: For DLC Type = FRPVC, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′92′ Port Identifier (always present): product-dependent identifier X′93′ Destination Service Access Point Address (DSAP) (always present) X′94′ Data Link Connection Identifier (DLCI) (always present) X′95′ Source Service Access Point Address (SSAP) (present if multiple parallel logical links are supported) X′96′ Destination MAC Address (present if RFC 1490 bridged 802.5 frame format is used) **Note:** The frame-relay Boundary Access Node (BAN) function uses this RFC 1490 format.
For DLC Type of Frame-Relay Switched Virtual Connection (SVC):		
2– 6 (= n)		DLC or physical interface type (EBCDIC characters): FRS (for frame-relay switched virtual connection) *Note: For DLC Type = FRS, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′97′ Link Layer Core Parameters (always present) X′A5′ Called Party Number and Link Service Access Point (LSAP) Address (always present)

DLC Signaling Type (X′91′) TG Descriptor Subfield

Byte	Bit	Content
For DLC Type of Internet Protocol (IP) Network:		
2– 3 (= n)		DLC or physical interface type (EBCDIC characters): IP (for Internet Protocol network) *Note: For DLC Type = IP, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′A5′ IP Address and Link Service Access Point (LSAP) Address (always present)
For DLC Type of ATM Switched Virtual Channel (SVC):		
2– 5 (= n)		DLC or physical interface type (EBCDIC characters): ATMS (for ATM switched virtual channel) *Note: For DLC Type = ATMS, the following subfields (described following this subfield) may be included in a TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′97′ Traffic Descriptor and Quality of Service (always present) X′A5′ Link Service Access Point (LSAP) Address and ATM Address (always present)
For DLC Type of X.25 Permanent Virtual Circuit (PVC):		
2– 7 (= n)		DLC or physical interface type (EBCDIC characters): X25PVC (for X.25 Permanent Virtual Circuit) *Note: For DLC Type = X25PVC, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′92′ Port Identifier (always present): product-dependent identifier X′93′ Logical Channel Number (always present)
For DLC Type of X.25 Switched Virtual Circuit (SVC):		
2– 7 (= n)		DLC or physical interface type (EBCDIC characters): X25SVC (for X.25 Switched Virtual Circuit) — or X25, now retired *Note: For DLC Type = X25SVC, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′92′ Port Identifier (always present): product-dependent identifier X′94′ Protocol Identifier (present when a protocol other than the default is being selected) X′A5′ Called (destination) DTE Address (always present) X′A6′ Calling (origin) DTE Address (present when calling DTE address is required, for example, to convey a subaddress) X′AE′ Optional User Facilities (present when one or more of these optional facilities are being selected) X′CD′ Call User Data (present when call user data is to be included in a call request packet)
For DLC Type of SDLC Nonswitched:		
2– 7 (= n)		DLC or physical interface type (EBCDIC characters):

DLC Signaling Type (X′91′) TG Descriptor Subfield

Byte	Bit	Content
		SDLCNS (for SDLC nonswitched) *Note: For DLC Type = SDLCNS, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′92′ Port Identifier (always present): product-dependent identifier X′93′ Secondary Station Address (always present) X′94′ Group Address (optionally present if group addressing is used) X′95′ Station Role (retired) X′96′ Asynchronous SDLC Support Type (present if asynchronous SDLC operation is used)
For DLC Type of SDLC Switched:		
2– 7 (= n)		DLC or physical interface type (EBCDIC characters): SDLCSW (for SDLC switched) *Note: For DLC Type = SDLCSW, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the DLC Signaling Type (X′91′) subfield:* X′92′ Port Identifier (always present): product-dependent identifier X′93′ Secondary Station Address (always present) X′94′ Group Address (optionally present if group addressing is used) X′95′ Station Role (retired) X′96′ Asynchronous SDLC Support Type (present if asynchronous SDLC operation is used) *Note: For DLC Type = SDLCSW, a second DLC Signaling Type (X′91′) subfield specifying the switched physical interface type, along with its associated subfields, immediately follows the SDLCSW subfields. The Port Identifier subfields in both the SDLCSW X′91′ series and the corresponding physical interface X′91′ series contain the same port identifier.*
For Physical Interface Type of V.25:		
2– 4 (= n)		DLC or physical interface type (EBCDIC characters): V25 (for V.25 switched physical interface) *Note: For Physical Interface Type = V25, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the Physical Interface Signaling Type (X′91′) subfield:* X′92′ Port Identifier (always present): product-dependent identifier X′93′ Destination Dial Address (always present) *Note: For Physical Interface Type = V25, a DLC Signaling Type (X′91′) subfield specifying the associated DLC protocol, along with its associated subfields, immediately precedes the subfield X′91′ and associated subfields for V25. The Port Identifier subfields in both the V25 X′91′ series and the corresponding DLC X′91′ series contain the same port identifier.*
For Physical Interface Type of V.25bis:		
2– 7 (= n)		DLC or physical interface type (EBCDIC characters):

DLC Signaling Type (X′91′) TG Descriptor Subfield

Byte	Bit	Content
		V25BIS (for V.25bis switched physical interface)
		Note: For Physical Interface Type = V25BIS, the following subfields (described following this subfield) may be included in the TG Descriptor (X′46′) control vector following the Physical Interface Signaling Type (X′91′) subfield: X′92′ Port Identifier (always present): product-dependent identifier X′93′ Destination Dial Address (present when the DLC does not require a composite command and address string) X′94′ Calling Identification Number (present when the DLC does not require a composite command and address string) X′95′ V.25 bis composite command and address string (present when the DLC requires this composite string)
		Note: For Physical Interface Type = V25BIS, a DLC Signaling Type (X′91′) subfield specifying the associated DLC protocol, along with its associated subfields, immediately precedes the subfield X′91′ and associated subfields for V25BIS. The Port Identifier subfields in both the V25BIS X′91′ series and the corresponding DLC X′91′ series contain the same port identifier.

DLC Signaling Information (X′92′) TG Descriptor Subfield

DLC Signaling Information (X′92′) TG Descriptor Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′92′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type = INTPU:		
2–5 (=n)		Internal PU identifier: ID block/ID number
	0–11	Block number: product-specific binary number, as contained in bytes 2–5, bits 0–11 in XID3
	12–31	ID number: binary number that, together with the block number, identifies a specific link station uniquely within a network, as contained in bytes 2–5, bits 12–31 in XID3
For DLC Type = TR, ETHERNET, FRPVC, X25PVC, X25SVC, SDLCNS, or SDLCSW:		
2 – n		Port Identifier: a variable-length hex string containing product-specific information to identify the port to which this signaling information pertains
For Physical Interface Type = V25 or V25BIS:		
2 – n		Port Identifier: a variable-length hex string containing product-specific information to identify the port to which this signaling information pertains *Note: This subfield contains the same port identifier as that contained in the associated SDLCSW series Port Identifier (X′92′) subfield.*

DLC Signaling Information (X′93′) TG Descriptor Subfield

DLC Signaling Information (X′93′) TG Descriptor Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′93′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		Subfield Data
For DLC Type = INTPU:		
2 – n		Internal PU identifier: DLUR network-qualified CP name (up to 17 characters)
For DLC Type = TR or FRPVC:		
2(=n)		Destination service access point address (DSAP) **Note:** The DSAP here is encoded in *noncanonical* format.
For DLC Type = ETHERNET:		
2(=n)		Destination service access point address (DSAP) **Note:** The DSAP here is encoded in *canonical* (i.e., ISO/IEEE 8802) format.
For DLC Type = X25PVC		
2–3 (=n)	0–3	Reserved
	4–15	Logical channel number: a binary number
For DLC Type = SDLCNS or SDLCSW:		
2(=n)		Secondary station SDLC address **Note:** The DLUR link station is always primary to its downstream dependent-LU link stations.
For Physical Interface Type = V25 or V25BIS:		
2 – n		Destination dial address: Circuit-switched network address (as defined in CCITT Recommendation E.163) of the destination, starting with the country code, encoded in packed form two decimal digits to the byte. If the dial address contains an odd number of digits, the low-order half-byte of the byte containing the last (low-order) digit contains X′C′. Lengths up to 18 digits are accommodated, thereby allowing for extension.

DLC Signaling Information (X′94′) TG Descriptor Subfield

DLC Signaling Information (X′94′) TG Descriptor Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′94′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		Subfield Data
For DLC Type = TR:		

DLC Signaling Information (X′94′) TG Descriptor Subfield

Byte	Bit	Content
2– 7 (= n)		Destination MAC address **Note:** The MAC address here is encoded in *noncanonical* format.
For DLC Type = ETHERNET:		
2– 7 (= n)		Destination MAC address **Note:** The MAC address here is encoded in *canonical* (i.e., ISO/IEEE 8802) format.
For DLC Type = FRPVC:		
2– 3 (= n)	0– 5	Reserved
	6– 1 5	Data link connection identifier (DLCI)
For DLC Type = X25SVC		
2(=n)		Protocol identifier: X′C2′ PSH — Physical Services Header X′C3′ QLLC — Qualified Logical Link Control X′C6′ ELLC — Extended Logical Link Control X′CB′ QLLC — Qualified Logical Link Control with SNA-specific diagnostic codes X′CE′ ELLC — Extended Logical Link Control with SNA-specific diagnostic codes
For DLC Type = SDLCNS or SDLCSW:		
2(=n)		Group SDLC address
For Physical Interface Type = V25BIS:		
2 – n		Calling identification number: Circuit-switched network address (as defined in CCITT Recommendation E.163) of the calling data station, starting with the country code, encoded in packed form two decimal digits to the byte. If the calling identification number contains an odd number of digits, the low-order half-byte of the byte containing the last (low-order) digit contains X′C′. Lengths up to 18 digits are accommodated, thereby allowing for extension.

DLC Signaling Information (X′95′) TG Descriptor Subfield

DLC Signaling Information (X′95′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′95′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type = TR or FRPVC:		
2(=n)		Source service access point address (SSAP) **Note:** The SSAP here is encoded in *noncanonical* format.
For DLC Type = ETHERNET:		
2(=n)		Source service access point address (SSAP) **Note:** The SSAP here is encoded in *canonical* (i.e., ISO/IEEE 8802) format.
For Physical Interface Type = V25BIS:		

DLC Signaling Information (X′95′) TG Descriptor Subfield

Byte	Bit	Content
2 – n		Composite command and address string: A 3-character V.25 bis command followed by those addresses, separators, and other special characters that are appropriate for that command. A maximum of 64 EBCDIC characters may be contained in this subfield. Any characters that are acceptable to the specific V.25 bis modem being used may included in this subfield. The ordering within the subfield conforms to ITU-T Recommendation V.25 bis, so that no reordering is required before passing this signaling information to the modem.

DLC Signaling Information (X′96′) TG Descriptor Subfield

DLC Signaling Information (X′96′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′96′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type = TR:		
2– 7 (= n)		Source MAC address **Note:** The MAC address here is encoded in *noncanonical* format.
For DLC Type = ETHERNET:		
2– 7 (= n)		Source MAC address **Note:** The MAC address here is encoded in *canonical* (i.e., ISO/IEEE 8802) format.
For DLC Type = FRPVC:		
2– 7 (= n)		Destination MAC address **Note:** The MAC address here is encoded in *noncanonical* format.
For DLC Type = SDLCNS or SDLCSW:		
2(=n)		Asynchronous SDLC Support Type
	0	Data transparency type: 0 8-bit data transparency not selected 1 8-bit data transparency selected (allows 8-bit data to be transmitted on 7-bit networks)
	1– 2	Character transparency type: 00 basic transparency selected 01 flow-control transparency selected (includes basic transparency) 10 control-character transparency selected (includes flow-control transparency) 11 Reserved
	3– 7	Reserved

DLC Signaling Information (X′97′) TG Descriptor Subfield

DLC Signaling Information (X′97′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′97′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type = FRS:		
2 – n		Link-layer core parameters
2		Link-layer-core-parameters content options:
	0	Outgoing best-effort indicator: 0 best-effort service not requested 1 best-effort service requested (i.e., throughput, minimum acceptable throughput, and committed burst size should all be set to 0)
	1– 2	Reserved
	3	Dedicated SVC support indicator: 0 dedicated SVCs not supported 1 dedicated SVCs supported
	4	Dedicated SVC request indicator (reserved except when this subfield is included in an HPR Route Setup): 0 dedicated SVC not requested 1 dedicated SVC requested
	5– 7	Reserved
3 – n		Outgoing parameter block
3		Outgoing throughput (present only when the Outgoing Best-Effort indicator is set to 0): a floating-point number, in units of 4 bits per second, representing the desired average outgoing bandwidth for the connection (see **Note** at the end of this subfield for the encoding of this field)
4		Outgoing minimum acceptable throughput (present only when the Outgoing Best-Effort indicator is set to 0): a floating-point number, in units of 4 bits per second, representing the minimum desired average outgoing bandwidth for the connection (see **Note** at the end of this subfield for the encoding of this field)
5		Outgoing committed burst size (present only when the Outgoing Best-Effort indicator is set to 0): a floating-point number, in units of 8 bits, representing the maximum amount of committed data a user may offer to the network during the computed measurement interval (see **Note** at the end of this subfield for the encoding of this field)
n		Outgoing excess burst size: a floating-point number, in units of 8 bits, representing the maximum amount of data by which a user can exceed the committed burst size during the computed measurement interval (see **Note** at the end of this subfield for the encoding of this field)
For DLC Type = ATMS:		
2		Traffic descriptor and quality of service (QoS) format: X′00′ Format for ATM Forum UNI 3.1
3 – n		Traffic Descriptor and QoS Format X′00′ Continuation

DLC Signaling Information (X′97′) TG Descriptor Subfield

Byte	Bit	Content
3		Traffic descriptor content options:
	0	Best-effort indicator: 0 best-effort service not requested 1 best-effort service requested
	1	Reserved
	2	Dedicated SVC support indicator: 0 dedicated SVCs not supported 1 dedicated SVCs supported
	3	Dedicated SVC request indicator (reserved except when this subfield is included in an HPR Route Setup): 0 dedicated SVC not requested 1 dedicated SVC requested
	4–7	Reserved
4 – n		Traffic parameters and QoS class
For Best-Effort indicator = 0:		
4–7 (= n)		Outgoing traffic and QoS parameters
4		Outgoing peak bit rate (cell loss priority [CLP]=0|1): a floating-point number, in units of 300 bits per second, representing the product of the peak cell rate multiplied by 384 bits per cell (see **Note** at the end of this subfield for the encoding of this field)
5		Outgoing sustainable bit rate (CLP=0): a floating-point number, in units of 300 bits per second, representing the product of the sustainable cell rate multiplied by 384 bits per cell (see **Note** at the end of this subfield for the encoding of this field)
6		Outgoing maximum burst size (CLP=0): a floating-point number, in units of 300 bits, representing the product of the maximum number of cells that can be transmitted at the peak cell rate multiplied by 384 bits per cell (see **Note** at the end of this subfield for the encoding of this field)
7(=n)		Outgoing traffic management options and QoS class
	0–1	Reserved
	2	Tagging requested indicator: 0 tagging not requested 1 tagging requested
	3	Reserved
	4–7	Outgoing quality-of-service (QoS) class: X′0′ QoS class 0 (unspecified) X′1′ QoS class 1 X′2′ QoS class 2 X′3′ QoS class 3 X′4′ QoS class 4
For Best-Effort indicator = 1:		
4		Outgoing peak bit rate (cell loss priority [CLP]=0|1): a floating-point number, in units of 300 bits per second, representing the product of the peak cell rate multiplied by 384 bits per cell (see **Note** at the end of this subfield for the encoding of this field)
5(=n)	0–3	Reserved
	4–7	Outgoing quality-of-service (QoS) class: X′0′ QoS class 0 (unspecified) X′1′ QoS class 1 X′2′ QoS class 2 X′3′ QoS class 3 X′4′ QoS class 4

DLC Signaling Information (X′97′) TG Descriptor Subfield

Byte	Bit	Content
		Note: Floating-point numbers represented above include (effectively) 4 bits of mantissa and 5 bits of exponent, encoded as follows. The binary value is first normalized and the exponent adjusted appropriately. The mantissa to be encoded consists of the four bits to the right of the binary point. The remaining bits are truncated. The high-order bit of the mantissa is not included in the encoding, since by the normalization it is 1. The encoded value in the corresponding byte fields above consists of the remaining three bits of the mantissa as the low-order three bits and the exponent as the high-order five bits. As an example, consider the value 23, binary 10111, which normalized is $.10111_2 \times 2^5$. Truncating all but the most significant four bits, leaves $.1011_2 \times 2^5$ (equals 22). The high-order 1 of the mantissa is assumed, so the encoded value is 00101 011.

DLC Signaling Information (X′A5′) TG Descriptor Subfield

DLC Signaling Information (X′A5′) TG Descriptor Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′A5′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		Subfield Data
For DLC Type = FRS:		
		Note: When this subfield is included in a FID2 Encapsulation GDS variable, the LSAP is the destination service access point address (DSAP), and the called party number is the destination frame relay address. For a frame-relay connection network TG, the LSAP and the called party number are associated with the node attached to the frame-relay connection network.
2		Link service access point address (LSAP) **Note:** The LSAP here is encoded in *noncanonical* format.
3 – n		Called party number (based on ITU-T Recommendation Q.933 and Frame Relay Forum Implementation Agreement FRF.4)
3	0	Pad bit: always set to 1
	1–3	Type of number: 000 unknown (This type of number is used when prefix or escape digits are included.) 001 international number (Prefix or escape digits are not to be used with this type of number.)
	4–7	Numbering plan identification: 0001 ISDN/telephony numbering plan (CCITT Recommendation E.164/E.163) 0011 Data numbering plan (ITU-T Recommendation X.121)
4 – n		Called party number: number digits encoded in packed form, two decimal digits to the byte. If the called party number contains an odd number of digits, bits 4–7 of the last byte contain a X′F′ pad character.
For DLC Type = IP:		

DLC Signaling Information (X′A5′) TG Descriptor Subfield

Byte	Bit	Content
2		Link service access point address (LSAP) **Note:** For an IP connection network TG, the LSAP is associated with the node attached to the IP connection network. The LSAP here is encoded in *noncanonical* format.
3	0–3	IP version (indicates the format of the IP header including the IP address) 0100 IP version 4 (only value defined)
	4–7	Reserved
4–7 (= n)		IP address
For DLC Type = ATMS:		
2		Link service access point address (LSAP) **Note:** For an ATM connection network TG, the LSAP is associated with the node attached to the ATM connection network. The LSAP here is encoded in *noncanonical* format.
3		ATM address type: X′02′ Private ATM endsystem address in OSI NSAP (ISO 8348) format X′11′ Native E.164 address: A native E.164 address is an international ISDN number composed of a variable length of decimal digits arranged in specific code fields. The code fields are the country code used to select the destination country, the national destination code which may be used to select a destination network, and the subscriber number. See CCITT Recommendation E.164, ISDN/Telephony Numbering Plan.
4 – n		ATM address
For ATM Address Type = X′02′		
4–23		Private ATM endsystem address coded as described in ISO 8348, Addendum 2, using the preferred binary encoding
24 – n		Native E.164 address of a local gateway to a public ATM network (optionally present if there is such a gateway): calling number digits, starting with the country code, encoded in packed form two decimal digits to the byte. If the native E.164 address contains an odd number of digits, bits 4–7 of the last byte contain a X′F′ pad character. If the Authority and Format indicator of the private ATM endsystem address specifies E.164 ATM format, no gateway address is required; a native E.164 address can be derived from an E.164 private ATM endsystem address.
For ATM Address Type = X′11′		
4 – n		Native E.164 address: calling number digits, starting with the country code, encoded in packed form two decimal digits to the byte. If the native E.164 address contains an odd number of digits, bits 4–7 of the last byte contain a X′F′ pad character.
For DLC Type = X25SVC:		
2 – n		X.25 Called (destination) DTE Address (based on CCITT Recommendation X.25 1992)
2	0	Address format type indicator (X.25 A bit): 0 non-TOA/NPI (type of address/numbering plan identification) address 1 TOA/NPI address
	1–7	Reserved
3 – n		Address Information for Address Format Type Indicator = 0
3	0–3	Reserved
	4–7	Length, in binary, of called DTE address in half-bytes (number of BCD digits — maximum of 15)

DLC Signaling Information (X′A5′) TG Descriptor Subfield

Byte	Bit	Content
4 – n		Called DTE address: Packet-switched network address (as defined in CCITT Recommendation X.25 1992), starting with the country code, encoded in packed form two decimal digits to the byte. If the DTE address contains an odd number of digits, the low-order half-byte of the byte containing the last (low-order) digit contains X′0′.
3 – n		Address Information for Address Format Type Indicator = 1
3		Length, in binary, of called DTE address field in half-bytes (The TOA/NPI indicators are included in this length, which has a maximum value of 17 for X.121 addresses.)
4		TOA/NPI indicators
	0– 3	Type of address: 0000 network-dependent number 0001 international number 0010 national number 0101 alternative address
	4– 7	Numbering plan identification: 0001 address as defined in CCITT Recommendation E.164. 0011 address as defined in CCITT Recommendation X.121
5 – n		Called DTE address: Packet-switched network address encoded in packed form, two decimal digits to the byte. If the DTE address contains an odd number of digits, the low-order half-byte of the byte containing the last (low-order) digit contains X′0′.

DLC Signaling Information (X′A6′) TG Descriptor Subfield

DLC Signaling Information (X′A6′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′A6′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type = X25SVC:		
2 – n		X.25 Calling (origin) DTE Address (based on CCITT Recommendation X.25 1992)
2	0	Address format type indicator (X.25 A bit): 0 non-TOA/NPI (type of address/numbering plan identification) address 1 TOA/NPI address
	1– 7	Reserved
3 – n		Address Information for Address Format Type Indicator = 0
3	0– 3	Reserved
	4– 7	Length, in binary, of calling DTE address in half-bytes (number of BCD digits — maximum of 15)
4 – n		Calling DTE address: Packet-switched network address (as defined in CCITT Recommendation X.25 1992), starting with the country code, encoded in packed form two decimal digits to the byte. If the DTE address contains an odd number of digits, the low-order half-byte of the byte containing the last (low-order) digit contains X′0′.

DLC Signaling Information (X′A6′) TG Descriptor Subfield

Byte	Bit	Content
3 – n		Address Information for Address Format Type Indicator = 1
3		Length, in binary, of calling DTE address field in half-bytes (The TOA/NPI indicators are included in this length, which has a maximum value of 17 for X.121 addresses.)
4		TOA/NPI indicators
	0– 3	Type of address: 0000 network-dependent number 0001 international number 0010 national number 0101 alternative address
	4– 7	Numbering plan identification: 0001 address as defined in CCITT Recommendation E.164. 0011 address as defined in CCITT Recommendation X.121
5 – n		Calling DTE address: Packet-switched network address encoded in packed form, two decimal digits to the byte. If the DTE address contains an odd number of digits, the low-order half-byte of the byte containing the last (low-order) digit contains X′0′.

DLC Signaling Information (X′AE′) TG Descriptor Subfield

DLC Signaling Information (X′AE′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′AE′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type = X25SVC:		
2 – n		X.25 Optional User Facilities This subfield is used to convey the information needed to invoke one or more optional user facilities on a call basis. The subfield data is coded exactly as it is to be placed in a call request packet. (See Recommendation X.25 for a complete listing of the optional user facilities that may be included in a call request.) If the inclusion of different categories of facilities requires the use of CCITT-specified facility markers to separate them, these markers are always included in the subfield. The coding of the two facilities most commonly associated with basic APPN connectivity, closed user groups and reverse charging, is given below. These are standard X.25 facilities, which may be included in either order and do not require facility markers separating them. In each case, the facility request is shown starting at byte 0.
0		Closed user group facility code: X′03′ basic closed user group X′47′ extended closed user group
1		For basic or extended format index (binary coded decimal)
	0– 3	First digit of closed user group index
	4– 7	Second digit of closed user group index

DLC Signaling Information (X′AE′) TG Descriptor Subfield

Byte	Bit	Content
2		For extended format index (binary coded decimal) — not present for basic closed user group
	0– 3	Third digit of closed user group index
	4– 7	Fourth digit of closed user group index
0		Facility code: X′01′ RC/ICRD/FS (Reverse Charging/ICRD Status Selection/Fast Select) facility
1		RC/ICRD/FS parameter field
	0– 1	Fast-select parameter: 00 fast-select not requested 01 fast-select not requested 10 fast-select requested with no restriction 11 fast-select requested with restriction
	2– 3	ICRD parameter: 00 ICRD status not selected 01 ICRD allowance requested 10 ICRD prevention requested 11 reserved
	4– 6	Reserved
	7	Reverse charging parameter: 0 reverse charging not requested 1 reverse charging requested **Note:** Only the reverse charging parameter is expected to be used in APPN.

DLC Signaling Information (X′CD′) TG Descriptor Subfield

DLC Signaling Information (X′CD′) TG Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′CD′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
For DLC Type = X25SVC:		
2 – n		X.25 Call User Data Data to be transferred from DTE to DTE in the call user data field of a call request packet immediately following the one-byte protocol identifier defined in subfield X′94′. A maximum of 15 bytes may be defined (n = 16) unless the fast-select facility is used. In that case, 127 bytes may be defined (n = 128). One use of this subfield is to convey a logon identifier.

TG Characteristics (X′47′) Control Vector

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

Note: As a base requirement, all implementations can accept and forward the TG Characteristics control vector in its entirety.

TG Characteristics (X′47′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′47′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data:
2– 5		Resource sequence number: a 32-bit binary value used to time-stamp each resource update in the topology database update. It is always incremented by 2 by the node that creates the resource update. When the low-order bit of the sequence number is set to 1, a node has recognized inconsistent received data; that is, more than one update with the same TG descriptor and with the same resource sequence number but with different data has been received by that node. If the RSN for a resource reaches the maximum value, the owner of the resource should rename the resource and reintroduce it into the network. A node may generate a maximum RSN about its own resources of only 2^{32}-2, since it increases RSNs by 2 for these resources. The value 2^{32}-1 is used only when a node indicates that it found another node's resource update using an RSN of 2^{32}-2 to be inconsistent.
6		Status:
	0	Operational status: 0 The TG is not operational. 1 The TG is operational (only value sent by an end node).
	1	Garbage collection indicator: 0 The TG record represented by this control vector is not marked to be garbage collected (deleted from the topology database) upon the next garbage collection cycle. 1 The TG record is marked to be garbage collected upon the next garbage collection cycle.
	2	Quiescing: 0 The TG is not quiescing. 1 The TG is quiescing.
	3– 4	CP-CP session support and status: 00 CP-CP sessions are supported on this TG (retired: using this setting, it is not indicated whether the sender's contention-winner CP-CP session is active or not active). 01 CP-CP sessions are supported and the sender's (i.e., the TG record owner's) contention-winner CP-CP session is active on this TG. 10 CP-CP sessions are not supported and the sender's contention-winner CP-CP session is not active on this TG. 11 CP-CP sessions are supported but the sender's contention-winner CP-CP session is not active on this TG.
	5	Multilink TG (MLTG) indicator: 0 This TG is not an MLTG. 1 This TG is an MLTG.
	6– 7	Reserved

TG Characteristics (X′47′) Control Vector

Byte	Bit	Content
7		Effective capacity: a floating-point number, in units of 300 bits per second, representing the product of a user-defined maximum load factor and the bit transmission rate of the link underlying the TG (see Note 1 at the end of this control vector for the encoding of this field)
8–12		Reserved
13		Cost per connect time: a value representing the relative cost per unit time of using the TG; permissible values are 0–255, where the value 0 means free
14		Cost per byte transmitted: a value representing the relative cost of transmitting a byte over the TG; permissible values are 0–255, where the value 0 means free
15		Reserved
16		Security: X′01′ not secure X′20′ public switched network used; secure in the sense that traffic takes no predetermined route X′40′ underground cable; located in a secure country (as determined by the network administrator) X′60′ secure conduit containing the transmission medium, not guarded (e.g., pressurized pipe) X′80′ guarded conduit containing the transmission medium, protected against physical tapping X′A0′ link-level encryption used X′C0′ guarded conduit containing the transmission medium, protected against physical and radiation tapping
17		Propagation delay: propagation delay of the TG; given as a floating-point number (see Note 1) specifying microsecond units. The following default values and ranges are defined.

default	meaning	range(decimal)	range(hex)
X′00′	"minimum"	–	–
X′4C′	negligible	<0.48 ms	<X′4F′
X′71′	terrestrial	0.48–49.152 ms	X′4F′–X′84′
X′91′	packet-switched	49.152–245.76 ms	X′84′–X′97′
X′99′	long	>245.76 ms	>X′97′
X′FF′	"maximum"	–	–

18		Reserved
19		User-defined parameter 1
20		User-defined parameter 2
21(=n)		User-defined parameter 3

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

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

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

Topology Resource Descriptor (X′48′) Control Vector

The Topology Resource Descriptor control vector carries information stored in the network topology database (TDB) related to either a transmission group (TG) or a network node. The Topology Resource Descriptor control vector (CV) for a particular resource may have differing values in the TDBs of separate network nodes within the network. Because these values may differ, the contents of this CV are not compared with the contents of the corresponding stored CV record if a resource update for that resource is received in a TDU by an NN, and the NN already has a record for that resource in its TDB, and the RSNs of both the received resource information and the stored resource information are identical.

Topology Resource Descriptor (X′48′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′48′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data *Note:* The following subfields (described following this control vector) may be included. They are parsed according to subfield parsing rule LT. X′80′ Resource Time Left subfield (always present)

Resource Time Left (X′80′) Topology Resource Descriptor Subfield

Resource Time Left (X′80′) Topology Resource Descriptor Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2		Number of days left, in binary, before resource is to be deleted from the topology database (**Note:** If the resource is a TG, its record must also be marked either *inconsistent* or *inoperative* before its record is deleted.)

Multinode Persistent Sessions (MNPS) LU Name (X′49′) Control Vector

The Multinode Persistent Sessions (MNPS) LU Name (X′49′) control vector specifies the MNPS LU name of the partner LU and is included in non-path-switch Route Setup requests and replies when the partner LU supports MNPS. If both LUs support MNPS, the MNPS LU name is used in subsequent Locates issued when doing a path switch. Normally, the search is done using the CP Name of the partner. However, with MNPS, it may be the case that the LU moved to another node. By searching with the LU name, the searching node will find the LU even if it moves.

Multinode Persistent Sessions (MNPS) LU Name (X′49′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′49′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2		Network qualified name type: X′FE′ MNPS LU name
3 – n		Network qualified name: a 1- to 17-byte string consisting of a 1- to 8-byte type-1134 symbol-string network ID (NETID) concatenated by a period to a 1- to 8-byte type-1134 symbol-string name.

Real Owning Control Point (X′4A′) Control Vector

The Real Owning Control Point (X′4A′) control vector specifies the true control point that is hierarchically superior to the resource named in the Directory Entry (X′3D′) control vector preceding it. It is used when the CP name in the Associated Resource Entry (X′3C′) control vector type X′00F4′ (ENCP) in the hierarchy specifies — or, when CV X′3C′ is absent, the sender is — a surrogate for the real owning control point.

Real Owning Control Point (X′4A′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′4A′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2– 3		Resource type: X′00F4′ ENCP

Real Owning Control Point (X′4A′) Control Vector

Byte	Bit	Content
4 – n		Resource name: a 1- to 17-byte string consisting of a 1- to 8-byte type-1134 symbol-string network ID (NETID) concatenated by a period to a 1- to 8-byte type-1134 symbol-string name. The node sending a Register GDS variable optionally omits the NETID from control vector (CV) X′4A′, if the NETID for the CV X′4A′ matches that in the nearest CV X′3C′ preceding the X′4A′ within the same Register GDS variable. If no CV X′3C′ precedes the CV X′4A′, the sender optionally omits the NETID from CV X′4A′ if the NETID matches the sender′s NETID. The node receiving Register does the following: • If a NETID is present on CV X′4A′, it is used. • If the NETID is omitted from CV X′4A′, the NETID qualifying the CP name in the X′4A′ comes from the nearest CV X′3C′ preceding the CV X′4A′ within that Register GDS variable. If no CV X′3C′ precedes the CV X′4A′, the NETID comes from the NETID of the node that sent the Register.

DLUR/S Capabilities (X′51′) Control Vector

The DLUR/S Capabilities control vector is exchanged by the dependent LU requester and the dependent LU server to describe their DLUR/S capabilities once a DLUS-DLUR session is activated.

DLUR/S Capabilities (X′51′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′51′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Dependent LU support level: X′01′ Release 1
3		Node type: X′00′ APPN network node X′01′ APPN end node
4		DLUR/S node type: X′00′ DLUR X′01′ DLUS
5– 8		Flow-reduction sequence number: a value that identifies the latest Topology Database Update GDS variable received by the sender of the DLUR/S Capabilities control vector from the receiver of the DLUR/S Capabilities control vector

DLUR/S Capabilities (X′51′) Control Vector

Byte	Bit	Content
9		Support indicators (bit is set to 1 if the sender supports the function):
	0	RECEIVE_TDU_TP (X′22F0F0F4′) service transaction program supported: The sending CP supports receipt of TDUs on this session.
	1	Limited DLUR auto network shutdown (ANS) support: The DLUR terminates all LU-LU sessions immediately when connectivity to the DLUS is lost.
	2	Limited DLUR multisubnet support: The DLUR will not respond to dependent-LU searches when the Owning CP Respond (OCR) bit in the Find is set to 1.
	3	Network name forwarding supported: The sender supports DLUR forwarding of Network Name control vectors on ACTPUs and ACTLUs to DLUR-served PUs and their associated LUs.
	4	Nondisruptive DLUS-DLUR session deactivation type X′08A0 000B′ supported: The sender supports the use of UNBIND with sense data X′08A0 000B′ for CP-SVR pipe deactivation.
	5	FID2 Encapsulation (X′1500′) GDS variable blocking supported: The sender supports sending and receiving more than one FID2 Encapsulation (X′1500′) GDS variable in a DLUS-DLUR session BIU.
	6	CP-SVR pipe persistence supported: The sender supports CP-SVR pipes which are active even without any asociated pending or active SSCP-PU sessions; in addition, if the sender is a DLUR, the sender requests persistence for this particular CP-SVR pipe.
	7	Reserved
10–13		Reserved

Primary Send Pacing Window Size (X′52′) Control Vector

Primary Send Pacing Window Size (X′52′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′52′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Vector Data
2		Primary send pacing window size
	0–1	Reserved
	2–7	Primary send window size, in binary, for session-level pacing *Note:* A value of 0 indicates that no pacing is to be performed.

Call Security Verification (X′56′) Control Vector

The Call Security Verification control vector is available for sending security verification information for a switched call to the SSCP.

Call Security Verification (X′56′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′56′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – q		Vector Data
2		Reserved
3 – q		Security IDs
3		Length, in binary, of Security IDs field; including this length field
4– 1 1		First 8-byte security ID (random data or enciphered random data)
12– 1 9		Second 8-byte security ID (random data or enciphered random data or space characters)

DLC Connection Data (X′57′) Control Vector

The DLC Connection Data control vector carries the characteristics of a DLC-level connection used to establish an SNA connection.

DLC Connection Data (X′57′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′57′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2		DLC type: X′00′ reserved X′01′ token-ring X′02′ token-bus X′03′ Ethernet X′04′ FDDI X′05′ ISDN X′06′ T1 TDM X′07′ X.25 (NPSI) X′08′ channel
3 – n		Vector Data *Note:* The following subfields (described following this control vector) may be included. X′01′ LAN MAC and SAP Data X′02′ Related Resource Network Name X′03′ LAN Routing Information X′04′ Port Number Name X′05′ ISDN Call Connection ID X′06′ T1 TDM Port Name X′07′ Frame-Relay DLCI X′08′ IP Address

LAN MAC and SAP Data (X′01′) DLC Connection Data Subfield

The LAN MAC and SAP Data subfield carries information that describes the LAN LLC connection used to establish an SNA session or connection. It always includes the local MAC and SAP addresses and may also include the remote MAC and SAP addresses.

LAN MAC and SAP Data (X′01′) DLC Connection Data Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′01′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Flag byte:
	0	0 only local MAC and SAP included 1 both local and remote pairs included
	1–7	Reserved
3–8		Local MAC address
9		Local SAP address
10–15		Remote MAC address (present only when byte 2, bit 0 = 1)
16		Remote SAP address (present only when byte 2, bit 0 = 1)

Related Resource Network Name (X′02′) DLC Connection Data Subfield

The Related Resource Network Name subfield carries the name of a related SNA resource.

Related Resource Network Name (X′02′) DLC Connection Data Subfield

Byte	Bit	Content
0–1		Subfield header; Key=X′02′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Reserved
3–10		Network Name of the related resource

LAN Routing Information. (X′03′) DLC Connection Data Subfield

The LAN Routing Information subfield carries information that describes the route over a series of bridged LANs that the SNA session or connection will follow. It includes the bridge numbers and the segment (ring) numbers.

LAN Routing Information (X′03′) DLC Connection Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′03′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Reserved
3		Length of Routing Information field (maximum of 18 bytes)
4 – n		Routing Information

Port Number Name (X′04′) DLC Connection Data Subfield

The Port Number Name subfield carries the port number, the value for PortNumberName, the naming attribute of the ISDN Layer1 object and of the DS1TDM Adapter object.

Port Number Name (X′04′) DLC Connection Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′04′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Reserved
3		Length of Port Number Name field
4 – n		Port number name

ISDN Call Connection ID (X′05′) DLC Connection Data Subfield

The ISDN Call Connection ID subfield carries the call connection identifier, the value for ISDNLayer3Name, and the naming attribute of the ISDN Layer 3 Object.

ISDN Call Connection ID (X′05′) DLC Connection Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′05′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Reserved
3		Length of Call Connection Identifier field (maximum of 50, or X′32′, bytes)
4 – n		Call connection identifier

T1 TDM Port Name (X′06′) DLC Connection Data Subfield

The T1 TDM Port Name subfield carries the port name and the naming attribute of the T1 TDM Layer 1 Object.

T1 TDM Port Name (X′06′) DLC Connection Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′06′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Reserved
3– 1 0		T1 TDM port name

Frame-Relay DLCI (X′07′) DLC Connection Data Subfield

The Frame-Relay DLCI subfield carries the data link connection identifier (DLCI) for logical frame-relay stations only.

Frame relay DLCI (X′07′) DLC Connection Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′07′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Reserved
3		Local SAP
4		Remote SAP
5		Length of DLCI field
6 – n		DLCI: hex characters as assigned by public carriers

IP Address (X′08′) DLC Connection Data Subfield

The IP Address subfield carries the IP address.

IP address (X′08′) DLC Connection Data Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′08′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Reserved
3– 6		IP address

Installation-Defined CDINIT Data (X′59′) Control Vector

Installation-Defined CDINIT Data (X′59′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′59′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Vector Data
2 – n		Installation-defined data that is passed between SSCPs on CDINIT and RSP(CDINIT) and passed to the application program on CINIT

Session Services Extensions Support (X′5A′) Control Vector

Session Services Extensions Support (X′5A′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′5A′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Flags:
	0	LU status indicator (reserved if not DSRLST): 0 not requested 1 requested
	1	Mode name translated indicator (reserved if not DSRLST): 0 not yet translated 1 mode name already translated
	2	Subarea-only function indicator: 0 Function requested is not subarea-only. 1 Function requested is a subarea-only function.
	3– 7	Reserved

Interchange Node Parameters (X′5B′) Control Vector

The Interchange Node Parameters control vector contains APPN related information that one interchange node passes through a subarea portion of the network to another interchange node.

Interchange Node Parameters (X′5B′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′5B′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)

Interchange Node Parameters (X′5B′) Interchange Node Parameters Subfield

Byte	Bit	Content
2 – n		Vector Data The following subfields may be included (no more than one of each); they are parsed according to parsing rule LT X′80′ Additional Interchange Node Support (always present) X′81′ Entry Interchange Node Name (always present on a request, invalid otherwise)

Interchange Node Session Initiation Indicators (X′80′) Interchange Node Parameters Subfield

The Interchange Node Session Initiation Indicators subfield contains APPN related session initiation information that is passed across a subarea portion of a network from one interchange node to another.

Interchange Node Session Initiation Indicators (X′80′) Interchange Node Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Vector Data
2		Flags
	0	Verification not required (only valid on requests): 0 required 1 not required
	1	Verification not performed (only valid on responses): 0 performed 1 not performed
	2	Original request indicator (only valid on requests): 0 original subarea request was not Init Other CD 1 original subarea request was Init Other CD
	3– 7	Reserved

Entry Interchange Node Name (X′81′) Interchange Node Parameters Subfield

The Entry Interchange Node Name subfield carries the network qualified name of the interchange node control point that originated the Interchange Node Parameters Subfield control vector.

Entry Interchange Node Name (X′81′) Interchange Node Parameters Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Vector Data

Entry Interchange Node Name (X′81′) Interchange Node Parameters Subfield

Byte	Bit	Content
2 – n		Entry Interchange Node Name (network qualified CP name of the interchange node that originated this control vector; see "Network Name (X′0E′) Control Vector" on page 9-26 for details on length and content restrictions)

APPN Message Transport (X′5C′) Control Vector

The APPN Message Transport control vector carries APPN-related control information across a subarea portion of the network. Because this information may exceed the 255-byte control vector length limitation, multiple X′5C′ control vectors may be used to carry segments of APPN information in an order-dependent fashion.

APPN Message Transport (X′5C′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′1E′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		Vector Data The following subfield may be included; it is parsed according to subfield parsing rule LT. X′80′ GDS Variable Transport subfield (always present)

GDS Variable Transport (X′80′) APPN Message Transport Subfield

The GDS Variable Transport subfield of the APPN Message Transport control vector carries all or some portion of an APPN GDS variable across a subarea portion of the network. In the case where the GDS variable to be transported exceeds the maximum allowable control vector length, multiple order-dependent X′5C′ control vectors are used to carry the GDS variable in segments.

GDS Variable Transport (X′80′) APPN Message Transport Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2 – n		GDS variable data (i.e., all or some portion of the GDS variable to be transported across the subarea portion of a network)

Subarea Message Transport (X′5D′) Control Vector

The Subarea Message Transport control vector carries subarea-specific information transparently across an APPN network.

Subarea Message Transport (X′5D′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′5D′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Vector Data *Note:* The following subfields (described following this control vector) may be included. They are parsed according to subfield parsing rule LT. X′81′ Disjoint Network subfield (present when searching APPN networks from an interchange node)

Disjoint Network (X′81′) Subarea Message Transport Subfield

The Disjoint Network subfield carries information regarding the contiguity of the network described by the subject network identifier.

Disjoint Network (X′81′) Subarea Message Transport Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′81′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Flags
2	0	Disjoint network indicator: 0 The subarea network indicated in bytes 4 – n is not disjoint (i.e., has fully meshed SSCP-to-SSCP sessions). 1 The subarea network identified in bytes 4 – n is disjoint.
	1– 7	Reserved
3		Length (1–8), in binary, of the network identifier
4 – n		Network identifier of the subject subarea network: a 1- to 8-byte type-1134 symbol string with optional (but not significant) trailing, but no imbedded, space (X′40′) characters. As noted in Appendix A, “SNA Character Sets and Symbol-String Types,” implementation usage constrains the leading character of the identifier to be alphabetic.

Related Request (X′5E′) Control Vector

Related Request (X′5E′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′5E′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Vector Data
2	0– 1	Request category: 00 FMD 01 NC 10 DFC 11 SC
	2	FMD message unit type (reserved when RU category is not FMD): 0 RU 1 GDS variable
	3– 7	Reserved
3		Length of Request Identifier field
4 – n		Request identifier: request code, NS header, or GDS variable identifier

Extended Fully Qualified PCID (X′5F′) Control Vector

Extended Fully Qualified PCID (X′5F′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′5F′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2	0– 3	Purpose of using this control vector: 0000 session correlation (only value defined)
	4– 7	Reserved
3– 10		PCID
11		Length (3–17), in binary, of network-qualified CP name
12 – n		Network-qualified CP name

Fully Qualified PCID (X′60′) Control Vector

The fully qualified procedure correlation identifier (FQPCID) is a unique value throughout an entire network.

Fully Qualified PCID (X′60′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′60′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – p		Vector Data
2– 9		PCID: a unique value used as a procedure identifier
10		Length (3–17), in binary, of Network-Qualified CP Name field
11 – n		Network-qualified CP name (network identifier always present)
n + 1 – p		Subfields as described below; they are parsed according to parsing rule KL (since they are carried only on subarea-only RUs): X′81′ PCID Modifier subfield (present only when the FQPCID control vector is included on a CDINIT or DSRLST request or response; or on a CDSESSST, CDTERM, INIT-OTHER-CD, or NOTIFY request) — Notice that on a Locate GDS variable, the PCID Modifier is carried as a control vector in conjunction with, rather than as a subfield of, the X′60′ control vector; there it is parsed LT.

PCID Modifier (X′81′) Fully Qualified PCID Subfield

PCID Modifier (X′81′) Fully Qualified PCID Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′81′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Subfield Data
2– 3		Procedure resubmit number
4		Last significant half-byte in the PCID Modifier List field (zero-origin index to the last half-byte in the list that has been claimed by a node on the procedure path)
5 – n		PCID modifier list: up to 20 half-bytes of list entries, with each list entry containing a binary count of the number of new search subprocedures created by a particular node. The Last Significant Half-byte field is used to determine the next available half-byte in the PCID Modifier List. The PCID modifier list has a minimum length of 1 byte, and must always contain an integral number of bytes. Thus, if byte 4 specifies an odd number of half-bytes (i.e., byte 4 is even), then the last byte in the list is padded with X′0′ to form a full byte. A node must be able to receive at least twenty entries in the list.

HPR Capabilities (X′61′) Control Vector

The presence of the HPR Capabilities control vector (CV) in XID3 indicates that the sender desires that the link run HPR protocols; it is used on negotiation-proceeding and nonactivation XID3s. On a prenegotiation XID3, it is not sent and is simply ignored by the receiver if received. Further details on usage appear in *HPR Architecture Reference*.

When used on nonactivation XID3 only the RTP Tower Supported indicator (byte 2, bit 2) can change; all other fields remain the same as on the last-sent CV X′61′ (which could have been on a negotiation-proceeding or nonactivation XID3). Receivers of CV X′61′ on a nonactivation XID3 may assume that other fields have not changed and thus need not check them.

HPR Capabilities (X′61′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′61′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2	0–1	Error recovery mode, indicating whether or not error recovery is required or preferred on this link (e.g., error recovery that is done by the LLC layer) for NLPs (Note: FID2 packets always require error recovery; FID2 packets are supported when this field is set to 00, 01, or 10.): 00 error recovery required for NLPs (ERP) 01 error recovery not allowed for NLPs (¬ ERP) 10 no error recovery preferred for NLPs but will be done if partner wants it (*ERP) 11 error recovery not available for NLPs or FID2 packets (¬ FID2) (if the receiver does not support this value, the XID is rejected by sending a CV X′22′ with sense data X′10160028′— invalid error recovery mode specified) Notes: • Condition 1: If one node specifies ERP and the other node specifies ¬ ERP, the link will run as a base-APPN link (i.e., HPR protocols will not be used on this link). Also, since a conflict has occurred, an Alert will be sent with the sense data X′10160021′. • Condition 2: If both nodes specify ERP, or if one node specifies ERP and the other specifies *ERP, error recovery is done. • If neither Condition 1 nor Condition 2 is true, error recovery is not done. • The ERP value should be chosen when the link is deemed unreliable (i.e., has a high error rate). If the link is reliable (low error rate), then either the *ERP or ¬ ERP values should be used. • Nodes that support only LDLC specify ¬ FID2.
	2	RTP tower (1401 option set) supported indicator: 0 This node does not support the RTP tower. 1 This node does support the RTP tower.
	3	Control Flows Over RTP tower (1402 option set) support indicator—this field is meaningful only when the sender supports the RTP tower (i.e., byte 2, bit 2 is 1); otherwise, it is reserved: 0 This node does not support the Control Flows Over RTP tower. (Note: If the X′81′ subfield is received, it is ignored.) 1 This node does support the Control Flows Over RTP tower. (Note: If the X′81′ subfield is not present, the XID3 fails with sense data X′10160027′.)

HPR Capabilities (X′61′) Control Vector

Byte	Bit	Content
	4	Logical Data Link Control (LDLC) supported indicator: 0 This node does not support LDLC. (Note: If this value is received by a node that supports only LDLC, the XID3 fails with sense data X′10160034′.) 1 This node does support LDLC.
	5–7	Reserved
3		Reserved
4		ANR Label length: a binary value in the range 1 to 8 indicating the length of the following field
5 – j		The ANR label, assigned by the XID sender, for this link (TG) in the direction that this XID is flowing. This label is used in the ANR Path (X′67′) control vector of the HPR Switching Information (X′83′) control vector in the Switching Information (X′14′) segment when activating CP-CP or route-setup RTP connections.
j + 1 – n		The following subfields (SFs) may be present. They are parsed according to the subfield parsing rule LT and may appear in any order. X′80′ IEEE 802.2 LLC subfield (SF X′80′)—optionally present in the following contexts: • LAN (token-ring, Ethernet): If not sent, the sender′s SAP value is defaulted to X′C8′. • Frame relay: Sent when the sender supports SAP multiplexing (encapsulation of multiple HPR link connections within a single frame-relay virtual connection by multiplexing on the SAP fields in an IEEE 802.2 header). If both sides send SF X′80′, the IEEE 802.2 header in the frame-relay packet contains the SAP values from the SF X′80′s. In the HPR base support for frame relay, this subfield is not sent. SAP multiplexing support over frame relay is optional. • ATM: SF X′80′ is always sent. If SF X′80′ is received, then the IEEE 802.2 header contains the SAP values from the SF X′80′s; otherwise, no IEEE 802.2 header is used. • X.25: SF X′80′ is never sent. If SF X′80′ is received, it is ignored. The IEEE 802.2 header is not used. SF X′80′ SAP values are used only when NLPs are being carried over a link that is not performing link-level error recovery (see byte 2, bits 0–1). If link-level error recovery is being done, the system-defined APPN SAPs (i.e., X′04′ or an installation-defined value) are used and SF X′80′s are ignored. X′81′ Control Flows Over RTP Tower subfield (present only when the node sending the XID supports the Control Flows Over RTP option set)

Note: Any length errors found in this control vector causes a rejection to occur via a CV X′22′ with sense data X′086F0000′ and the Error Byte/Bit Offset fields set to point to the field in error.

IEEE 802.2 LLC (X′80′) HPR Capabilities Subfield

This subfield is optionally included in control vector X′61′ when XID3 is being exchanged over a link that uses IEEE 802.2 LLC (e.g., token-ring, Ethernet, ATM, or frame relay).

IEEE 802.2 LLC (X′80′) HPR Capabilities Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′80′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2		The LLC SAP that is to be used by the adjacent node when sending NLPs that do not require link-level error recovery to this node (i.e., the adjacent node uses this field′s contents as the destination LLC SAP). Senders may set this field to the default value (X′C8′ the IBM reserved value assigned for HPR non-ERP traffic) or a value specified by the node operator. The value of this SAP may be the same as the one being used for APPN. Products should allow the default value to be overridden so that conflicting SAP values, which may arise in some network configurations, can be avoided.

Control Flows Over RTP Tower (X′81′) HPR Capabilities Subfield

This subfield is included in control vector X′61′ (on XID3) by nodes that support the Control Flows Over RTP (1402) option set. This subfield provides information that would normally be obtained by the route-setup protocol. The route-setup protocol is not done prior to activating CP-CP session and route-setup RTP connections.

Nodes that do not support the Control Flows Over RTP option set ignore this control vector when received (i.e., treat it as an unknown control vector).

Control Flows Over RTP Tower (X′81′) HPR Capabilities Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′81′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2– 3		Maximum send packet size (in bytes) that can be sent by this node over this link: a value greater than or equal to 768, since 768 is the minimum "maximum packet size" supported by HPR. The value in this field is used to compute the maximum packet size on the return path in the HPR Switching Information CV segment. This field is required, since the link transmit and receive buffer sizes might be different; in the case where the transmit buffer size of the adjacent partner is smaller than the receiver buffer size of the local node, the transmit size will be used by the partner RTP to segment messages sent over the link. The size does not include the link header (e.g., the 802.2 link header).
4– 7		Path switch time: the time that the sender′s CP requires for a path switch (in milliseconds). This field is used along with the receiver CP′s path switch time to determine how much time to allow for a path switch.
8	0	Mobility indicator: 0 The sender is not mobile (i.e., the sender is stationary). 1 The sender is mobile.
	1	Multilink TG (MLTG) supported indicator: 0 MLTG is not supported. 1 MLTG is supported and it is desired that this link become part of the MLTG indicated in the XID TG number field (byte 16).

Control Flows Over RTP Tower (X′81′) HPR Capabilities Subfield

Byte	Bit	Content
	2–3	ARB mode - indicates the mode of ARB protocol supported by the XID (CV X′61′) sender. 00 Base mode ARB 01 Responsive mode ARB
	4–7	Reserved
9–12		Control point NCE instance identifier: an implementation-specified value
13–16		Route-setup NCE instance identifier: an implementation-specified value
17		Length of the following Control Point NCE Identifier field: a value in the range 1–8
18 – k		Control point NCE identifier: an identifier of the NCE that represents the CP in the node sending this CV and that is used by the adjacent node when sending NLPs to this CP over a CP-CP session on an RTP connection. It is the only ANR label in the ANR Routing field of the NHDR.
k+1		Length of the following Route Setup NCE Identifier field: a value in the range 1–8
k+2 – n		Route-setup NCE identifier: an identifier of the route-setup component associated with this link for the sender of this XID. There may be one route-setup component associated with each link, group of links, or all links in the node (i.e., one route-setup component per node). The number of route-setup components is determined by the implementation. The route-setup NCE identifier is used by the adjacent node when sending NLPs to the route-setup component in this node over the route-setup RTP connection. It is the only ANR label in the ANR Routing field of the NHDR

Session Address (X′62′) Control Vector

The Session Address control vector is used on positive RSP(BIND)s to convey the FID5 session address to be used by the primary LU when sending data to the secondary LU. It is assigned by the secondary LU. It is included on RSP(BIND)s that are carried in FID5 PIUs over RTP connections, but not on RSP(BIND)s carried in FID2 PIUs.

Session Address (X′62′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′62′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2–9		A FID5 session address, used to distinguish a session that is multiplexed with others on a single RTP connection, and therefore, unique per RTP connection.
	0	Session address assignor indicator: 1 Tthe session address was assigned by the sender of this control vector (only value allowed; if a value other than 1 is specified, the session is rejected with sense data X′08956202′).
	1	Reserved
	2–63	Session address

Cryptography Key Distribution (X′63′) Control Vector

The Cryptography Key Distribution control vector distributes cryptographic keys between nodes to establish cryptographic sessions.

Cryptography Key Distribution (X′63′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′63′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data *Note:* The following subfields may be included. They are parsed according to subfield parsing rule LT. X′80′ Cryptography Capabilities subfield (always present) X′81′ BIND Receiver Enciphered Session Key subfield (present when an enciphered session key is transmitted from one CP to another) X′82′ Cross-Domain Enciphered Session Key subfield (present when an enciphered session key is transmitted from one CP to another)

Cryptography Capabilities (X′80′) Cryptography Key Distribution Subfield

This subfield carries the cryptography capabilities of the sending control point to the receiving control point.

Cryptography Capabilities (X′80′) Cryptography Key Distribution Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2	0	Reserved
	1– 2	Cryptography support at sender: 00 no session-level cryptography supported 01 session-level cryptography is optional (this value carried on requests only); on the reply, the partner may specify no session-level cryptography, session-level selective cryptography, or session-level mandatory cryptography 10 session-level selective cryptography supported; all cryptography key management is supported by the CP and LUs; exchange (via +RSP(BIND)) and verification (via CRV) of the cryptography session-seed value is supported by the LUs; each sender selectively enciphers FMD RUs and sets the Enciphered Data indication in the RH accordingly 11 session-level mandatory cryptography supported; all cryptography key management is supported by the CP and LUs; exchange (via +RSP(BIND)) and verification (via CRV) of the cryptography session-seed value is supported by the LUs; each sender enciphers all FMD requests
	3	Session-key enciphering-key selection indicator: 0 Use EN(PLU) as target for enciphering final session key. 1 Use NNS(PLU) as target for enciphering final session key.
	4– 7	Reserved
3– 5		Reserved

BIND Receiver Session Key (X′81′) Cryptography Key Distribution Subfield

This subfield contains the session key enciphered under the SLU master key. It will eventually flow in a BIND.

BIND Receiver Session Key (X′81′) Cryptography Key Distribution Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′81′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2		Length, in binary, of BIND receiver session key
3 – n		BIND receiver session key enciphered under the SLU′s master cryptography key (when Triple DES is being used this a 24-byte key; otherwise it′s an 8-byte key)

Cross-Domain Enciphered Session Key (X′82′) Cryptography Key Distribution Subfield

This subfield contains the session key enciphered under a cross-domain key. It will eventually be deciphered and stored at the PLU as a session key.

Cross-Domain Enciphered Session Key (X′82′) Cryptography Key Distribution Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′82′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2		Length, in binary, of following enciphered session key
3 – i		Session key enciphered under the cross-domain key used between the CPs defined below (when Triple DES is being used this is a 24-byte key; otherwise it is an 8-byte key)
i+1 – m		Source CP Name
i+1		Length, in binary, of network-qualified name of the *source* enciphering control point, i.e., the control point that last enciphered the session key in bytes 3 – i
i+2 – m		Network-qualified name of the source control point: a 3- to 17-byte name consisting of a 1- to 8-byte network qualifier concatenated with a period and a 1- to 8-byte CP name, both variable-length components being type-1134 symbol-strings
m+1 – n		Target CP Name
m+1		Length, in binary, of the network-qualified name of the *target* processing control point, i.e., the control point that is to decipher, or forward under a different cross-domain key, the key received in bytes 3 – i
m+2 – n		Network-qualifed name of the target control point: a 3- to 17-byte name consisting of a 1- to 8-byte network qualifier concatenated with a period and a 1- to 8-byte CP name, both variable-length components being type-1134 symbol-strings

TCP/IP Information (X′64′) Control Vector

The TCP/IP Information control vector is used in the SNA network to associate TCP/IP information with SNA resources. One specific use of this control vector is to associate connecting Telnet 3270 client IP addresses with their LU resources.

TCP/IP Information (X′64′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′64′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data *Note:* The following subfield (described following this control vector) is included. Additional subfields (also described following this control vector) follow this subfield, depending upon the TCP/IP information type indicated in the X′91′ subfield, as indicated below. All subfields are parsed according to subfield parsing rule KL. X′91′ TCP/IP information type subfield (always present, always first)

TCP/IP Information Type (X′91′) TCP/IP Information Subfield

This subfield is always included in control vector X′64′ to identify the type of TCP/IP information provided by the control vector.

TCP/IP Information Type (X′91′) TCP/IP Information Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′91′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2	0– 3	TCP/IP information type: X′0′ Telnet 3270 client (only value defined)
	4– 7	Reserved
		Note: For TCP/IP information type = Telnet 3270 client, the following subfields (described following this subfield) may be included in the TCP/IP Information (X′64′) control vector following the TCP/IP Information Type (X′91′) subfield: X′81′ IP address subfield (always present) X′82′ Application port number subfield (always present) X′85′ IP host name subfield (optionally present)

IP Address (X′81′) TCP/IP Information Subfield

This subfield is always included in control vector X′64′ to identify the IP address associated with the LU resource.

IP Address (X′81′) TCP/IP Information Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′81′ (see "Substructure Encoding/Parsing Rules" on page 9-5 and Figure 9-1 on page 9-7)
2		IP version X′04′ IP version 4 (only value defined)
3	0– 5	Reserved
	6– 7	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
4 – n		IP address (of the IP version specified in byte 2; e.g., a 32-bit binary IP version 4 address)

Application Port Number (X′82′) TCP/IP Information Subfield

This subfield is always included in control vector X′64′ to identify the port number associated with the LU resource.

Application Port Number (X′82′) TCP/IP Information Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′82′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2– 3		Application port number: a 16-bit binary value

IP Host Name (X′85′) TCP/IP Information Subfield

This subfield may be included in control vector X′64′ to identify the IP host name associated with the LU resource.

IP Host Name (X′85′) TCP/IP Information Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′85′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2	0	IP host name truncation indicator: 0 full IP host name 1 truncated IP host name (beginning with the most-specific left-side label)
	1– 7	Reserved
3 – n		IP host name (full or truncated): domain name hierarchy in EBCDIC characters
		Note 1: Rules for encoding the IP host name can be found in *RFC 1034*.
		Note 2: The maximum count of the bytes in the full or truncated IP host name is 128 when this subfield is included in NOTIFY. When this subfield is included in RSP(ACTLU), the maximum count of bytes in the IP host name must be small enough that 145 bytes may be added to the RSP(ACTLU) by a T4 boundary node without the length of the RSP(ACTLU) exceeding 256 bytes; with all other control vectors and subfields at their maximum sizes, the maximum count of bytes in the IP host name is 66.

Device Characteristics (X′65′) Control Vector

The Device Characteristics (X′65′) control vector is provided in response to a request for session characteristics.

Device Characteristics (X′65′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′65′, (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data
2		Device data stream compatibility characteristics: X′00′ no data stream characteristics defined (the normal setting) X′04′ 2741 X′08′ WTTY X′10′ 115A X′20′ TWX (33–35) X′30′ 83B3 X′40′ 2740 X′80′ 1050 X′90′ 2780 X′91′ 3780 X′A0′ – X′FF′ available for installation-defined use
3		Device Language Support for 3270 Data Streams
3	0	Query indicator: 0 Identify language characteristics of the device from the code specified in bits 1–7. 1 Send query command to the device to determine the single-byte character-set language and double-byte character-set capability. (If the language cannot be determined from the input received from the query, the code specified in bits 1–7 will be used as the default.)
	1–7	Language supported: 0000000 US English (retired) 0000001 US English 0010001 Katakana
4(=n)		Device Flags
4	0	Buffer/screen size capability (reserved except for 3270 devices; 1 is the normal setting for 3270 devices): 0 Device cannot support a 1920-byte buffer or screen size. 1 Device can support a 1920-byte buffer or screen size.
	1	Inhibit-display and enable-display control characters supported (reserved except for TWX and WTTY devices): 0 characters not supported (printer device) 1 characters supported (display device)
	2	Switched connection to a boundary function (this indicator is reserved except in the device characteristics information created by an SSCP; the connection refers to the link connection between the node of the device and the subarea node containing the boundary function): 0 Device does not have a switched connection to a BF. 1 Device has a switched connection to a BF.
	3–7	Reserved

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

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

- For CINIT and CDCINIT, it specifies that the default compression level is to be overridden.
- For BIND and RSP(BIND), it carries the information necessary to negotiate compression.

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

Byte	Bit	Content
0– 1		Vector header; Key=X′66′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data *Note:* One of the following subfields (described following this control vector) is included, depending on the RU; they are parsed according to parsing rule LT. X′80′ Compression Override subfield (present on CDCINIT and CINIT if default compression is to be overridden) X′81′ RLE/LZ Compression Bid subfield (present on BIND if session is to use length-checked compression) X′82′ RLE/LZ Compression Result subfield (present on RSP(BIND) if session is to use length-checked compression)

Compression Override (X′80′) Length-Checked Compression Subfield

The Compression Override subfield specifies that the default level of compression on the BIND and RSP(BIND) is to be overridden. This subfield is used only when control vector X′66′ is carried on CDCINIT or CINIT.

Compression Override (X′80′) Length-Checked Compression Subfield

Byte	Bit	Content
0– 1		Subfield header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2– 3		Subfield Data
2		Override Flags
2	0– 5	Reserved
	6	Length-checked compression prohibited indicator: 0 not prohibited 1 prohibited; PLU must not include length-checked compression information on the BIND *Note:* See control vector X′66′, subfield X′81′ and the BIND RU, byte 25.

Compression Override (X′80′) Length-Checked Compression Subfield

Byte	Bit	Content
	7	Length-checked compression required indicator: 0 not required 1 required; in control vector X′66′ of the BIND, the PLU sets the Compression Level Needed By Links So Far field of subfield X′81′ to the PLU's compression capability and sets bits 6 and 7 of byte 25 in the BIND both to 1. *Note:* Bits 6 and 7 above cannot both be set to 1.
3		Length-Checked Compression Flags
3	0–6	Reserved
	7	BF.LU XRF-session compression support indicator: 0 not supported 1 supported

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

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

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

Byte	Bit	Content
0–1		Subfield header; Key=X′81′ (see "Substructure Encoding/Parsing Rules" on page 9-5)
2–4		Subfield Data
2		Length-Checked Compression Flags
	0	Compression and decompression involvement: 0 Length-checked compression and decompression may be performed by intermediate nodes. 1 Length-checked compression and decompression are done only by the session endpoints.
	1	RLE usage: 0 LZ and RLE compression in series is not allowed. 1 LZ and RLE compression in series is allowed.
	2–7	Reserved
3		Negotiable Compression Fields
	0–3	Compression capability of nodes so far: X′0′ no compression X′1′ RLE compression only X′2′ RLE and small-table LZ compression X′3′ RLE and medium-table LZ compression X′4′ RLE and large-table LZ compression
	4–7	Compression level needed by links so far: X′0′ no compression X′1′ RLE compression only X′2′ small-table LZ compression, optionally preceded by RLE X′3′ medium-table LZ compression, optionally preceded by RLE X′4′ large-table LZ compression, optionally preceded by RLE

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

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

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

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

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

Byte	Bit	Content
0–1		Subfield header; Key=X′82′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2–3		Subfield Data
2		Length-Checked Compression Flags
	0	Compression and decompression involvement: 0 Length-checked compression and decompression may be performed by intermediate nodes. 1 Length-checked compression and decompression are done only by the session endpoints.
	1	RLE usage: 0 LZ and RLE compression in series is not allowed. 1 LZ and RLE compression in series is allowed.
	2–7	Reserved
3		Negotiable Compression Fields
	0–3	Actual compression level of RUs going from PLU to SLU: X′0′ no compression X′1′ RLE compression only X′2′ small-table LZ compression, optionally preceded by RLE X′3′ medium-table LZ compression, optionally preceded by RLE X′4′ large-table LZ compression, optionally preceded by RLE

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

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

ANR Path (X′67′) Control Vector

The ANR Path control vector describes a path using a series of ANR label entries.

ANR Path (X′67′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′67′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		A series of ANR label entries (one for each hop), where each ANR label entry has the following format, shown in zero-origin format. (The ANR label entries are contiguous with one another; i.e., they are not word aligned.)
Format of an ANR label entry		
0		Length (k+1), in binary, of this length field (byte 0), the flag byte (byte 1), and the ANR label (bytes 2 – k). Since the length of an ANR varies from 1 to 8 bytes, this length field can have values ranging from 3 to 10.
1		Flag Byte
	0	Subarea route indicator: 0 The ANR label does not represent a route through a subarea network. 1 The ANR label represents a route through a subarea network (i.e., represents the route between a subarea network entry node and a subarea network exit node).
	1– 7	Reserved
2 – k		ANR label: a 1- to 8-byte implementation-defined string, the high-order bit of which is set to 1, and no byte of which is X′FF′ (see the ANR Routing field in the NHDR for further discussion)

XRF/Session Cryptography (X′68′) Control Vector

XRF/Session Cryptography (X′68′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′68′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)

XRF/Session Cryptography (X′68′) Control Vector

Byte	Bit	Content
2 – n		Vector Data
2		Indicator Bits
2	0	Session key indicator: 0 session key not present 1 session key present
	1	Seed indicator: 0 seed value not present 1 seed value present
	2–7	Reserved
3–4		Cryptography Options
3	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:* Values 00, 01, 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 4–20) omitted X′9′ session-level cryptography specified; additional options follow in next nine bytes
4	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 001 block chaining with seed and cipher text feedback, using the Triple Data Encryption Standard (Triple DES) algorithm
5–12		Session cryptography key enciphered under the NCP master cryptography key; an 8-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FMD requests. When the cryptography cipher method is Triple DES (byte 4, bits 5–7 set to 001) this field contains the first 8 bytes of the Triple DES session cryptography key. When concatenated to the last 16 bytes of the key (contained in bytes 21–36) it forms the complete 24-byte Triple DES session cryptography key enciphered under the NCP master cryptography key. This value, when deciphered, is used by the Triple DES algorithm for enciphering and deciphering FMD requests.
13–20 (=n)		An 8-byte implementation-chosen, nonzero, pseudo-random session-seed cryptography value enciphered under the session cryptography key

XRF/Session Cryptography (X′68′) Control Vector

Byte	Bit	Content
21–36 (=n)		This field is optional and is present only when the cryptography cipher method is Triple DES (byte 4, bits 5–7 set to 001) and contains the last 16 bytes of the Triple DES session cryptography key. See description under bytes 5-12 for how this field is used with Triple DES.

Switched Parameters (X′69′) Control Vector

Switched Parameters (X′69′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′69′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2	0	Check node identification 0 node identification should not be checked 1 node identification should be checked *Note:* Node identification refers to the IDBLOCK/IDNUM subfield and/or the Network Name (X′0E′) subfield shown below.
	1–7	Reserved
3 – n		Vector Data *Note:* No more than one of each of the following subfields may be included and each is parsed KL: X′0E′ Network Name (invalid unless the Check Node Identification bit = 1) X′80′ Dial Number X′81′ Direct Call Line Name X′82′ IDBLOCK/IDNUM (invalid unless the Check Node Identification bit = 1) *Note:* Multiple subfields are arranged in ascending numerical order of the subfield key.

Network Name (X′0E′) Switched Parameters Subfield

Network Name (X′0E′) Switched Parameters Subfield

Byte	Bit	Content
0–1		Vector header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Bytes 2 – n of the Network Name control vector

Dial Number (X′80′) Switched Parameters Subfield

Dial Number (X′80′) Switched Parameters Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Dial Number

Direct Call Line Name (X′81′) Switched Parameters Subfield

Direct Call Line Name (X′81′) Switched Parameters Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′81′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Direct Call Line Name

IDBLOCK/IDNUM (X′82′) Switched Parameters Subfield

IDBLOCK/IDNUM (X′82′) Switched Parameters Subfield

Byte	Bit	Content
0– 1		Vector header; Key=X′82′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		IDBLOCK/IDNUM (see bytes 2– 5 of the XID)

ER Congestion Data (X′6A′) Control Vector

ER Congestion Data (X′6A′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′6A′ (see “Substructure Encoding/Parsing Rules” on page 9-5)
2 – n		Vector Data
2– 5		Subarea address, in the network in which the containing RU is flowing, of the PU that has appended this control vector

ER Congestion Data (X′6A′) Control Vector

Byte	Bit	Content
6	0	Pseudo-slowdown indicator: 0 not in pseudo-slowdown 1 in pseudo-slowdown
	1	Slowdown indicator: 0 not in slowdown 1 in slowdown *Note:* The pseudo-slowdown state exists when the number of buffers in the free buffer pool minus the number of committed buffers is less than the slowdown state entry threshold. There are three valid combinations of bits 0 and 1. These combinations occur when both Slowdown and Pseudo-Slowdown indicators are set to 1, both indicators are set to 0, and, lastly, when the Slowdown indicator is set to 0 and the Pseudo-Slowdown indicator is set to 1.
	2–7	Reserved
7–10		Total number of buffers in the buffer pool created at system initialization time
11–14		Number of committed buffers: number of buffers in the free buffer pool expected to be used by link I/O processes that are currently executing
15–18		Number of free buffers: number of buffers in the free buffer pool currently available for use, including those already committed
19–22		Slowdown state entry threshold: number of buffers that must be available in the free buffer pool in order to satisfy all types of buffer requests and to return pacing responses; when the number of buffers in the free buffer pool minus the number of committed buffers falls below this number, polling (input) ceases on peripheral and non-SNA links; when the total number of buffers in the free buffer pool falls below this number, VR and session pacing responses are withheld.
23–26		Slowdown state exit threshold: the number of buffers that must be available in the buffer pool to leave the slowdown state
27–30		Number of reserved buffers: the number of buffers reserved at all times for system use to avoid deadlock; when the number of buffers in the free buffer pool minus the number of committed buffers falls below this number, polling (input) ceases on subarea links *Note:* Slowdown state exit threshold = 1.5 x slowdown entry state threshold > number of reserved buffers > 0.

Triple DES Cryptography Key Continuation (X′71.′) Control Vector

Triple DES Cryptography Key Continuation (X′71.′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′71′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2–17		Last 16 bytes of the 24-byte enciphered Triple DES session cryptography key.

Control Vector Keys Not Recognized (X′FE′) Control Vector

Control Vector Keys Not Recognized (X′FE′) Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′FE′ (see “Substructure Encoding/Parsing Rules” on page 9-5 and Figure 9-1 on page 9-7)
2 – n		Vector Data: one or more one-byte control vector key values that were not recognized in the corresponding request

Control Lists

Introduction

The following table shows, by list type, the control lists and the message structures that can carry the control list.

Type	Control List	Message-Unit Structures
X′01′	LU Status	+RSP(DSRLST)

Control List Formats

The control lists are defined, by type, as follows (with 0-origin indexing of the list bytes; see the individual RU description for the actual displacement within the RU):

Note: Control List data is requested by type; however, the type is *not* included in the reply RU that carries the request control list.

LU Status (X′01′) Control List

LU Status (X′01′) Control List

Byte	Bit	Content
Type: X′01′		
0		LU Status
	0	Session availability indicator: 0 SSCP-SSCP session available 1 SSCP-SSCP session unavailable, NOTIFY forthcoming *Note:* If the SSCP-SSCP session is unavailable, the remaining information in the LU Status is not applicable.
	1	LU availability indicator: 0 LU is unavailable 1 LU is available *Note:* If the LU is unavailable, but an SSCP-LU path exists (i.e., byte 0, bit 4 of LU Status is 0), then the remaining information in bits 2–3 in the LU Status is applicable. If no SSCP-LU path exists, bits 2–3 are not meaningful.
	2–3	LU unavailability reason (reserved unless LU is unavailable): 00 LU session count exceeded 01 LU being taken down (not accepting new sessions) 10 LU not currently able to comply with the PLU specification 11 reserved
	4–5	SSCP-to-LU path availability and notification indicator: 00 existing SSCP-to-LU path 01 no existing SSCP-to-LU path; Notify forthcoming 10 no existing SSCP-to-LU path 11 reserved *Note:* Value 01 is sent by an SSCP that supports DSRLST X′04′ as specified in CDRM(X′06′) control vector, to an SSCP that has the same capability.
	6–7	Reserved

LU Status (X′01′) Control List

Byte	Bit	Content
1		LU information
	0	LU residence indicator: 0 LU does not reside in a T5 node. 1 LU resides in a T5 node.
	1	PLU capability: 0 LU capability of acting as PLU not determined 1 LU capable of acting as PLU (capability may be disabled now and thus the session will be queued)
	2–6	Reserved
	7	LU INITIATE/logon acceptance indicator: 0 LU is accepting INITIATEs/logons. 1 LU is temporarily not accepting INITIATEs/logons.
2–3		Session count (range: 0–65535)

Session Keys

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

Key	Session Key	Applicable Message-Unit Structures
X′01′	Network or Uninterpreted Name	Resource Available (retired) NOTIFY vector, TERM-SELF Format 1
X′05′	PCID	CDSESSEND, CDSESSSF, CDSESSTF, CDTERM, Resource Available (retired) NOTIFY vector, TERM-OTHER
X′06′	Network Name Pair or Uninterpreted Name Pair	CDSESSEND, CDSESSSF, CDSESSST, CDSESSTF, CDTERM, CLEANUUP, ILU/TLU or Third-party SSCP Notification NOTIFY vector, Resource Available (retired) NOTIFY vector, NSPE, TERM-OTHER
X′07′	Network Address Pair	BINDF, CDSESSEND, CDSESSSF, CDSESSST, CDSESSTF, CDTERM, CINIT, CLEANUP, CTERM, LU Deactivation NOTIFY vector, SESSEND, SESSST, TERM-OTHER, TERM-SELF Format 1, UNBINDF
X′0A′	URC	BFCINIT, BFCLEANUP, BFINIT, BFTERM, Resource Available (retired) NOTIFY vector, TERM-OTHER, TERM-SELF Format 1
X′15′	Network-Qualified Address Pair	BFCINIT, BFCLEANUP, BFINIT, BFSESSEND, BFSESSST, BFTERM, BINDF, CDCINIT, CDINIT, CDSESSEND, CDSESSSF, CDSESSST, CDSESSTF, CDTERM, CLEANUP, CTERM, ILU/TLU or Third-party SSCP Notification NOTIFY vector, LU Deactivation NOTIFY vector, LU-LU Session Status NOTIFY vector, REQACTCDRM, SESSEND, SESSST, TERM-OTHER, TERM-SELF Format 1, UNBINDF, Control Vector X′13′, Control Vector X′14′, Control Vector X′28′
X′1C′	Network-Qualified Name Pair	Resource Available (retired) NOTIFY vector, TERM-OTHER, NSPE

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 or Uninterpreted Name (X′01′) Session Key
(retired)

Network Name or Uninterpreted Name (X′01′) Session Key

Byte	Bit	Content
0		Key: X′01′
1		Type: X′F3′ logical unit (only value defined)
2		Length, in binary, of LU name
3 – m		LU name: For a Network Name session key, the name consists of type-1134 symbol-string characters; for an Uninterpreted Name session key, the name can be any EBCDIC string.

PCID (X′05′) Session Key

PCID (X′05′) Session Key

Byte	Bit	Content
0		Key: X′05′
1–8		PCID: A unique value, generated by the SSCP initiating the cross-domain procedure, that is retained and used in all cross-domain requests dealing with the same procedure until it is completed

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.

Network Address Pair (X′07′) Session Key

Network Address Pair (X′07′) Session Key

Byte	Bit	Content
0		Key: X′07′
1–2		Network address of NAU1
3–4		Network address of NAU2 *Note:* See the RUs that carry this session key for NAU1/NAU2 definitions and order requirements.

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

Network-Qualified Address Pair (X′15′) Session Key

Network-Qualified Address Pair (X′15′) Session Key

Byte	Bit	Content
0		Key: X′15′
1		Length, in binary, of Key Data field
2– 21		KEY Data field
2– 7		NAU1 network address
8– 13		NAU2 network address *Note:* See the RUs that carry this session key for NAU1/NAU2 definitions and order requirements.
14– 21		Network ID of the subnetwork in which the above addresses are valid *Note:* The Length byte is set to 12 when network ID is *not* included and to 20 when network ID is included. If the Network ID contains all space (X′40...40′) characters, the network addresses are in the sender′s network.

Network-Qualified Name Pair (X′1C′) Session Key

Network-Qualified Name Pair (X′1C′) Session Key

Byte	Bit	Content
0		Key: X′1C′
1		Length, in binary, of Key Data field
2		Session type: X′00′ SSCP-SSCP session X′01′ LU-LU session

Network-Qualified Name Pair (X′1C′) Session Key

Byte	Bit	Content
3 – n		Two control vectors, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields” *Note 1:* The following control vectors are used; they are parsed according to parsing rule LT. X′0E′ Network Name control vector X′0E′ Network Name control vector *Note 2:* The names belong to the two session partners. See the RUs that carry this session key for NAU1/NAU2 definitions and order requirements. *Note 3:* If either name is omitted, a null X′0E′ control vector is included, as described in the “Control Vectors” discussion in Chapter 9, “Common Fields.” Two X′0E′ control vectors, null or otherwise, are always present.

End of Chapter 9

Chapter 10. Sense Data

Introduction

The sense data included as a request reject (or outcome) code across an intra-nodal interface or carried in 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 (X′35′) control vector, an RTP Connection Fault (X′12′) segment, or an SNA report code is a 4-byte field (see Figure 10-1) that includes a 1-byte category value, a 1-byte modifier value, and two bytes of sense code specific information, whose format is defined along with the sense code definition, below.

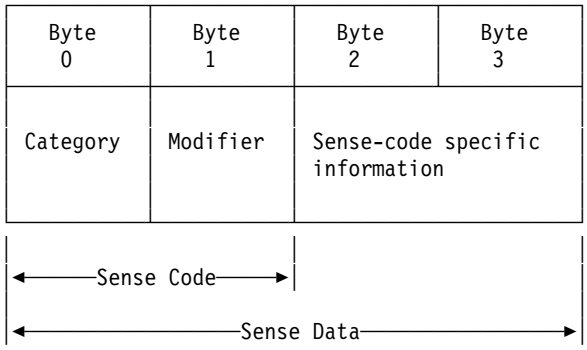


Figure 10-1. Sense Data Format

Together, the category byte 0, the modifier byte 1, and the sense code specific bytes 2 and 3 hold the sense data defined for the exception condition that has occurred.

The following categories are defined; all others are reserved:

VALUE	CATEGORY
X′00′	User Sense Data Only
X′08′	Request Reject
X′10′	Request Error
X′20′	State Error
X′40′	Request Header (RH) Usage Error
X′80′	Path Error
X′A0′	Rapid Transport Protocol (RTP) Error
X′FF′	Set aside for implementation-specific use

The category User Sense Data Only (X′00′) allows the end users to exchange sense data in bytes 2–3 for conditions not defined by SNA within the other categories (and perhaps unique to the end users involved). The modifier value is also X′00′. User Sense Data may not be sent on LU 6.2 sessions.

In earlier versions of SNA, user data (as well as implementation-specific data) generally could be carried in bytes 2–3 for all categories. This is no longer the case. Bytes 2–3 are used generally only for SNA-defined conditions for nonzero categories; exceptions for implementation-specific use are documented in the appropriate product publications.

The sense codes for the other categories are discussed below.

Request Reject (Category Code = X′08′)

This category indicates that the request was delivered to the intended component and was understood and supported, but not executed.

Category and modifier (in hexadecimal):

0801 Resource Not Available: The LU, PU, link station, or link specified in an RU is not available.

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

- 0000 No specific code applies.
- 0001 Independent LU Does Not Receive ACTLU: An ACTLU has been sent by the SSCP to an independent LU (sent by BF).
- 0002 Reserved Resources Requested for Sessions Exceed Allowable Maximum: The resource reservation request in RNAA exceeds the maximum allowed by system definition. The address was not assigned and no change was made to the current reservation of resources for the LU.
- 0003 Name aliasing cannot be performed because the name alias function is not available.
- 0004 A switched connection currently exists for the link being activated, and the SSCP or the subarea PU does not support the protocols necessary to allow take-over of such a link.
- 0005 A SETCV has been received for a resource that is still represented in the pool of available control blocks.
- 0006 The line is not associated with a line adapter, or is associated with a line adapter that is not valid for the usage tier specified in the receiving node′s system definition.
- 0007 The line is associated with a line adapter that is not installed or not attached to a communications processor.
- 0008 The line is associated with a line adapter that is inoperative.
- 0009 The LU is not available because it is not ready to accept sessions.
- 000A The PLU is not available because it is being taken down, and is therefore not accepting new sessions. The initiation request should not be retried.
- 000B The PLU is not available because it is unable to comply with the PLU-SLU role specification.

000C	The SLU is not available because it is unable to comply with the PLU-SLU role specification.
000D	The LU is not available because its SSCP is in the process of being taken down, and is therefore not allowing new sessions to be started. The initiation request should not be retried.
000E	The LU is not available because an intermediate gateway SSCP is in the process of being taken down, and is therefore not allowing new sessions to be started.
000F	The SLU is not available because it is being taken down, and is therefore not accepting new sessions. The initiation request should not be retried.
0010	Switched subarea connection cannot be established because no switched subarea links have been defined.
0011	Switched subarea connection to another network cannot be established because no switched subarea links have been defined within the gateway PU.
0012	An APPN connection cannot be established because this node has no available integers to represent a new TG.
0013	A switched connection cannot be established because no short-hold mode capable link is defined.
0014	A switched connection cannot be established. Call Request Verification was requested, but is not supported for this configuration. This condition will result from conflicting system definition.
0015	The link connection is unavailable as a result of a hardware failure within the line adapter.
0016	A link resource is not available as a result of maintenance occurring on the supporting hardware.
0017	A link resource is not available because a mismatch exists within the microcode of the supporting hardware.
0018	Activation of the channel link failed because the supporting hardware is undergoing error recovery.
0019	Activation of the channel link failed because the supporting hardware is undergoing concurrent maintenance.
001A-001E	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
001F	A link connection is associated with a protocol that is inoperative.
0020	A link resource is not available because of a mismatch between current operational parameters and the values specified at system generation.

0021 Resource Not Available: The link is associated with a connectivity subsystem that is not installed, not powered on, not initialized, or not operational.

0022 FRTE logical link activation failure caused by DLCI being not present. An ACTLINK was received for a FRTE resource but the last status message from the adjacent frame-relay node did not contain the DLCI.

0024 Link-level 2 test initiation failure caused by DLCI being inactive. An initiate link-level 2 test was received for a FRTE resource but the last status message from the adjacent frame-relay node indicated that the DLCI was inactive.

0025 Remote load failure caused by DLCI being inactive. An IPLINIT was received for a FRTE resource but the last status message from the adjacent frame-relay node indicated that the DLCI was inactive.

0026 The PU is not available because the connection between the dependent LU server and the dependent LU requester could not be established.

0027 A switched connection cannot be established because no switched link has been defined.

0028 REQDACTPU was received for a PU that is known but whose SSCP-PU session is currently inactive.

002A An ACTLINK has been received for a resource that is still represented in the pool of available control blocks.

002B The resource is unavailable as a result of a program or operator action.

002C The required Extended Coupling Facility is not available.

002D The ISDN B-channel specified in the dial digits of a CONNOUT for an ISDN logical link connection is currently being used for backing up a frame relay link connection, and is thus not available for the requested dial-on-demand function; the request has been rejected.

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, or an intermediate session routing (ISR) component in the path has reached its

maximum active intermediate session count. This sense code applies to ACTCDRM, INIT, BIND, and CINIT requests.

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

0000 No specific code applies.

0001 If accepted, the BIND request would prevent either the receiving LU or the sending LU from activating the number of contention winner sessions to the partner LU that were agreed upon during a change-number-of-sessions procedure.

0002 If accepted, the BIND request would cause the XRF-backup session limit to be exceeded.

0003 If accepted, the BIND request would cause the XRF-active session limit to be exceeded.

Note: The session limit for XRF-active sessions is 1. An XRF-active BIND is valid only if there are no XRF-active or XRF-backup sessions with the receiving SLU.

0005 The intermediate session router is unable to create a session connector control block. The pool of session connectors is saturated with active sessions and with pending-active sessions for which the Queue bit was set in the BIND; the BIND should not be retried.

0006 The intermediate session router is unable to create a session connector control block. The pool of session connectors is saturated with active sessions and with pending-active sessions for which the Queue bit was not set in the BIND; the BIND should be retried.

0008 For a dependent LU, if accepted, the BIND request would cause the session limit to be exceeded.

0009 If accepted, the request would cause the PLU session limit to be exceeded.

000A If accepted, the request would cause the SLU session limit to be exceeded.

000B The request was rejected because a session already exists between the same LU pair, and at least one of the LUs does not support parallel sessions.

000C An LU-LU session was not established because a session already exists between the SLU and the session-controller PLU.

000D The ISR component was unable to create a session connector because it had reached its maximum active intermediate session count. The BIND is rejected.

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.

0005 The physical unit is currently in the physical unit dynamic reconfiguration pool.

0006 For a dynamic reconfiguration DELETE, MOVE, or REPLACE operation, the resource to be dynamically reconfigured could not be found.

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

0008 For a dynamic reconfiguration DELETE, MOVE, or REPLACE operation, the NAU name in RNAA does not correspond to the resource identified by the element address in the RNAA.

0009 Reserved

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.
002B	The TG being activated is unknown.
002C	The identification supplied by the adjacent node in its XID3 differed from the identification that the receiving node was configured to expect.
002D,002E	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
002F	The destination resource was not found on this node during a session activation attempt.
0030	The adjacent node was not identified during CP-CP session activation or deactivation.
0031	Upon receiving a route request from another component in the node, TRS has been unable to locate in its topology database the destination network node or any network node specified in the TG vectors for the destination end node; the request is rejected.
0032	A SETCV defining an intra-FRSE PCV segment subport set was received containing an element address unknown to the receiver.
0033	A network resource needed for session establishment has become unavailable resulting in the termination of the pending session establishment procedure.
0034	REQDACTPU was received for an unknown PU.
0035	The local node has received an unknown adjacent CP name in a request to activate or deactivate a CP-CP session.

0036 No SSCP-SSCP session exists between the VRTG endpoints.

0807 Resource Not Available—LUSTAT Forthcoming: A subsidiary device will be unavailable for an indeterminate period of time. LUSTAT will be sent when the device becomes available.

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

0000 No specific code applies.

0001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0808 Invalid Contents ID: The contents ID contained on the ACTCDRM request was found to be invalid.

0809 Mode Inconsistency: The requested function cannot be performed in the present state of the receiver.

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

0000 No specific code applies.

0001–000D Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

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

000F An RNAA received from an SSCP is rejected because it specifies a resource (adjacent link station or LU) that currently has an address assigned as a result of another SSCP′s RNAA; or an ACTLU, FNA, or SETCV received from an SSCP is rejected because it specifies a resource address that is not assigned to an existing resource or is assigned as a result of another SSCP′s RNAA.

0010–0013 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0014 ANS mismatch discovered.

0015 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0016 The PU type on SETCV does not match the actual PU type.

0017,0018 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0019	A SETCV was received containing a value for the SDLC BTU send limit that conflicts with the previous value received.
001A,001B	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
001C	The RNAA request contains a network ID that is not known to the gateway PU.
001D	An address pair session key in a Network-Qualified Address Pair (X′15′) control vector is not known to the gateway PU.
001E	A gateway PU received an RNAA request for a cross-network session and all possible address transforms for the named resource are allocated.
001F	Retired
0020	The gateway node receiving an RNAA request cannot support another session between the named resource pair.
0021–0023	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0024	A PU received an ACTPU request with the SSCP-PU Session Capabilities control vector (X′0B′) indicating that the sending SSCP does not support ENA, but the PU does not know the SSCP′s maximum subarea address value.
0025	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0026	A SETCV was received containing an SDLC BTU send limit of 0.
0027	A request for a function was received by a component but the function was not enabled or activated.
0028	Cleanup termination of an LU-LU session has been converted to a forced termination by the LU. The SSCP must wait for session ended signals before deleting its session awareness records of the session.
0029	Second-level application state change has occurred. An application program served by an MS application program has changed the state of a node that may result in the rejection or failure of a current request or of a future request that would have previously been honored.
002A	The route setup procedure identified in a session services request was not in the expected state.

0030	An FNA was received for an LU that has an active LU-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 subject LU was not pending BFSESSINFO; the reported sessions will be terminated, and the associated network addresses will be freed. This sense data is also included in the BFCLEANUP when the sessions are terminated.
0033	A BIND with the same LFSID used for an existing pending-reset session has been received by a boundary function from a peripheral PLU.
0034	A termination request has been received for a resource that has been taken over by an SSCP. The termination type is not strong enough to apply to the resources. The termination type needs to be Forced or Cleanup.
0035	An other-domain resource that was expected to be active is inactive.
0036	The short-hold mode logical connection selected does not exist.
0037	A non-short-hold mode connection was attempted on a port that is dedicated to short-hold mode operation.
0038	There is an inconsistency of mode between the XID sender and receiver. The XID receiver is operating in short-hold mode. Examples include inconsistent settings of the Short-Hold indicator (SHI) and the Short-Hold Status indicator (SHSI).
0039	CP capabilities exchange protocol error: CP Capabilities (X′12C1′) GDS variable request sent indicating conversation complete or without change direction (i.e., CEB or ¬ CD) or CP Capabilities reply sent indicating conversation not yet complete (i.e., ¬ CEB).
003A	A null XID was received when an XID3 with its Exchange State indicators set to “prenegotiation,” “negotiation proceeding,”or “Exchange State indicators not supported” was expected.
003B	A null XID was received when a nonactivation XID3 was expected.
003C	An XID3 with the Exchange State indicators set to “pre-negotiation”was received when either of the values “negotiation proceeding” or “Exchange State indicators not supported” was expected.
003D	A nonactivation XID3 was received when a null XID or link-activation XID3 was expected.

003E A link activation XID3 was received when a null XID or nonactivation XID3 was expected.

003F The node with a secondary link station attempted to initiate a nonactivation exchange when secondary-initiated nonactivation exchanges are not supported on the connection.

0040 A mode-setting command was received and was either not expected or invalid for the receiving node; e.g., SNRME was received when SNRM was expected.

0041 An XID3 with the Exchange State indicators specifying a negotiation-proceeding exchange was received when an XID3 indicating a prenegotiation exchange was expected. If prenegotiation XID3s are used in a link activation XID exchange, each node must send and receive one.

0042 On an ABM TG on which secondary-initiated nonactivation XID exchanges are supported, the adjacent link station has initiated a nonactivation exchange by sending a nonactivation XID3 in which the ABM Nonactivation XID Exchange Initiator indicator specifies that the sending node is not initiating a nonactivation exchange. On such TGs, the initiator of a nonactivation exchange always explicitly indicates that it is initiating a nonactivation exchange.

0045 An XID3 indicating that the sender supports the Exchange State indicators was received when the sender had previously indicated that it does not support this field in XID3.

0046 An XID3 indicating that the sender does not support the Exchange State indicators was received when the sender had previously indicated that it does support this field in XID3.

0047 An XID has been received after receipt of a mode-setting command but before the completion of the mode-setting sequence, i.e., before RR, RNR, or an I-frame with the Poll bit set has been sent by the node with the primary link station after it has received UA in response to its mode-setting command.

0048 A node with an NRM primary link station has received an XID3 when it has no outstanding commands. NRM secondary link stations send XIDs only in response to XID commands.

0049 The XID3 received from the adjacent node had an XID Negotiation Error (X′22′) control vector appended. The XID exchange will therefore terminate unsuccessfully.

004A The request cannot be accepted because DR (dynamic reconfiguration) is in process for the target resource.

004B-004D	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
004E	A node with a secondary NRM link station has attempted to initiate a nonactivation XID exchange with an XID3. Nodes with secondary NRM link stations may solicit a nonactivation XID3 exchange only by means of sending a null XID at a response opportunity.
0050	An UNBIND request was received on behalf of a resource for which a previous UNBIND is in progress. The second UNBIND does not indicate an override of the first, and is therefore a duplicate request.
0051,0052	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0053	An activation request was received for an SSCP whose subarea address is already associated with another SSCP name.
0054	The adjacent node is not the node type that the receiving node was configured to expect. The received negotiation-proceeding XID3 indicated that the adjacent node is an end node while this node expected the adjacent node to be a network node, or *vice versa*.
0055	The adjacent node is attempting to establish a connection through a connection network but the virtual routing node CP named in the TG Descriptor (X′46′) control vector appended on the received negotiation-proceeding XID3 is not valid.
0057	The received NOTIFY type is not supported in the current state of the receiver.
0058	An intra-FRSE PVC segment subport received an RNAA(Assignment Type X′5′) with a DLC Header Link Station Address field containing a value outside the valid range. The RNAA is rejected.
0059	An intra-FRSE PVC segment subport received an RNAA with a control vector X′43′ specifying discontinue link-level contact if an auto network shutdown procedure is initiated. The RNAA is rejected.
005A	RNAA was received to add an intra-FRSE PVC segment subport to a hierarchical physical resource that is not active.
0060	An intra-FRSE PVC segment subport received an RNAA with a control vector X′43′ specifying that modem test support is permitted. The RNAA is rejected.

0061 An intra-FRSE PVC segment subport received an RNAA with a control vector X′43′ Frame send control value field containing a value outside the valid range. The RNAA is rejected.

0062 An intra-FRSE PVC segment subport received an RNAA with a control vector X′43′ Node Type Identifier field specifying a node type other than T1. The RNAA is rejected.

0063 An intra-FRSE PVC segment subport received an RNAA with a control vector X′43′ specifying that null XID polling for the secondary station should be used. The RNAA is rejected.

0064 A SETCV defining an intra-FRSE PVC segment subport set was received that contained an element address in the DR pool.

0065 A SETCV defining an intra-FRSE PVC segment subport set was received from an SSCP that did not originally add all the subarea element addresses listed in the SETCV to the DR pool.

0066 An ACTTRACE was received for a link connection trace while a trace for a logical link using that link connection was active, or ACTTRACE was received for a logical link trace while a trace for its underlying physical link connection was active.

0067 An intra-FRSE PVC segment subport received an RNAA5 containing a DLC Header Link Station Address that is being used by an existing Frame Relay Terminating Equipment subport.

0068 An XID was received with a Networking Capabilities indicator (specifying whether the sender is an APPN network node) that is inconsistent with the receiver's definition for the connection. The connection is rejected.

0069 An XID was received with CP Services and CP-CP Session Support indicators that are inconsistent with the receiver's definition for the connection. The connection is rejected.

006A A node type mismatch exists between the two SSCPs setting up a VRTG. An SSCP learns when the VRTG is established that the partner SSCP's APPN node type (NN or EN) is different from the type locally defined for that partner SSCP.

006B The IP address specified in an RNAA(Type=X′05′) for a new Internet Protocol (IP) PU is a duplicate of an existing IP address.

080A Permission Rejected: The receiver has denied an implicit or explicit request of the sender.

When sent in response to BIND, it implies either that the secondary LU will not notify the SSCP when a BIND can be accepted, or that the SSCP does not recognize the NOTIFY vector key X′0C′. (See the X′0845′ sense code for a contrasting response.)

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

0000 No specific code applies.

0001 An SSCP has denied permission to establish a session through its resources; the receiving SSCP should not attempt to reroute the request to another SSCP.

0002 An SSCP has denied permission to establish a session through its resources; the receiving SSCP should attempt to reroute the request to another SSCP.

0005 An SSCP has denied permission to continue session setup. A DLU name was presented to an alias selection function (authorization application) that rejected it. The receiving SSCP should not attempt to reroute the request to another SSCP.

0006 An SSCP has denied permission to continue session setup. A DLU name was presented to an alias selection function (authorization application) that rejected it. The receiving SSCP should attempt to reroute the request to another SSCP.

0007 An activation request was received for a resource that has a NETID different from that of the requesting SSCP, and the requesting SSCP indicated previously that it does not support this configuration.

0008 The request specified in the Request Change Control MS major vector was rejected because it did not originate from a valid focal point.

0009 The request specified in the Request Change Control MS major vector was rejected because the ability to support it has been disabled at the receiver.

000A The request was rejected because it would prohibit compliance with the status-reporting requirements specified in the Reporting Level MS Common subvector.

000B The request was rejected because the second-level application, though recognized, operates under the control of a program other than that which has received and is to forward the request to that second-level application program.

000C The request was rejected because the timer/clock at the receiver is protected and cannot be set by the request sender.

000D An SSCP or CP has denied a Locate search request. The receiving SSCP or CP should attempt to reroute the request.

000F The request was rejected because of constraints or policies specific to the receiving DLUS. The request should not be retried.

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.

0001–0004 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 a Name List MS subvector or an SNA Address List MS subvector with multiple entries, but the receiver supports only a single entry in such a subvector.

000B An MS Request Change Control major vector was received requesting automatic delayed acceptance, but the receiver does not support that function.

000C Set aside for implementation specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

000D An MS Request Change Control major vector was received requesting post-test, but the receiver does not support that function.

000E An MS Request Change Control major vector was received prohibiting automatic removal of a change, but the receiver does not support that function.

000F	An Activate MS major vector was received specifying use of changes installed in production only, but the receiver supports such a request only when it is received locally.
0010	Reserved
0011	Set aside for implementation specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.
0012	Application GDS variable in an MDS-MU not supported.
0013	MDS message type not supported. Receiving node does not support the MDS message type in this MDS-MU.
0014	An MS major vector was received requesting execution window timing, but the receiver does not support that function.
0015	An MS Activate major vector was received specifying change management activation use, but the receiver does not support that function.
0016	An MS Request Change Control major vector was received requesting Activate with Force Delay, but the receiver does not support that function.
0017	The changes referred to in a Request Change Control MS major vector are already installed on trial and the receiver does not support the transfer from trial to production with removability=yes.
0018	An MS Request Change Control major vector was received requesting pre-test, but the receiver does not support that function.
0019	A link trace requested in ACTTRACE is not supported for frame-relay logical links.
001A	A CONNOUT has been received for an ISDN B-channel link connection by a receiver that does not support this function; the request has been rejected.
01nn	A subvector in an MS major vector is identified as one inside which the receiver requires one of a number of supported subfields, but none of these subfields is present. Byte 3 contains the key (nn) of the subvector.
4001, 4003	Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

080D NAU Contention: A request to activate a session was received while the receiving half-session was awaiting a response to a previously sent activation request for the same session; for example, the SSCP receives an ACTCDRM from the other SSCP before it receives the response for an ACTCDRM that it sent to the other SSCP and the

SSCP ID in the received ACTCDRM was less than or equal to the SSCP ID in the ACTCDRM previously sent.

080E NAU Not Authorized: The requesting NAU does not have access to the requested resource.

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

0000 No specific code applies.

0001 The PU, according to its system definition, does not accept an ACTPU from any SSCP having the network ID of the sending SSCP.

0002 A gateway T4 node received a dynamic dump request from an SSCP that is not in the native network of the gateway T4 node.

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.

0006 A network node received a Register from an unauthorized APPN end node.

0007 A network node received a Register from another network node.

0008 A network node received a Delete from another network node.

0009 A network node received a Delete from an unauthorized APPN end node.

000A Retired

000B A Locate/CD-Initiate was received from a node that is not defined as a client end node. This can be detected by either DS or SS.

000C A gateway T4 node received a dynamic dump request from an SSCP that is not in the native network of the gateway T4 node.

080F End User or LU Not Authorized: The requesting end user or LU does not have the proper security authorization to access the requested resource.

0000 No specific code applies.

0001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0002 An LU that supports only the enhanced LU-LU verification protocol received a BIND or RSP(BIND) that specified the basic LU-LU verification protocol.

0003 An attempt was made to enter a remote subnetwork to which the origin subnetwork was not authorized.

0004 An attempt was made to install or remove a database table before issuing a logon to the database.

0983 Access Denied: The request specifies a resource that the requester is not permitted to access. Access to a resource is controlled by evaluation of the requester′s identity, profile, or location. This sense data is sent in FMH-7.

6051 Access Security Information Invalid: The request specifies an Access Security Information field that is unacceptable to the receiver; for security reasons, no further detail on the error is provided. This sense data is sent in FMH-7 or UNBIND.

6058 Access Failure: The request specifies an Access Security Information field that is unacceptable because of a processing failure.

8000 Access Failure: GSS-API Unexpected Status Value — A GSS-API request returned an unrecognized status value.

8001 Access Failure: GSS-API GSS_BAD_MECH — unsupported mechanism requested

8002 Access Failure: GSS-API GSS_BAD_NAMETYPE — name of unsupported type provided

8003 Access Failure: GSS-API GSS_BAD_BINDINGS — channel binding mismatch

8004 Access Failure: GSS-API GSS_BAD_NAME — invalid name provided

8005 Access Failure: GSS-API GSS_BAD_STATUS — invalid input status selector

8006 Access Failure: GSS-API GSS_BAD_SIG — token had invalid signature

8007 Access Failure: GSS-API GSS_DEFECTIVE_CREDENTIAL — defective credential detected

8008 Access Failure: GSS-API GSS_DEFECTIVE_TOKEN — defective token detected

8009 Access Failure: GSS-API GSS_FAILURE — failure, unspecified at GSS-API level

800A Access Failure: GSS-API GSS_NO_CONTEXT — no valid security context specified

800B Access Failure: GSS-API GSS_NO_CRED — no valid credentials provided

8101 Retryable Access Failure: GSS-API GSS_CONTEXT_EXPIRED — specified security context expired

8102 Retryable Access Failure: GSS-API GSS_CREDENTIALS_EXPIRED — expired credentials detected

8103 Retryable Access Failure: Deferred Authentication processing was requested, but communications failures occurred while attempting to communicate with the Distributed Authentication Service TP.

8104 Retryable Access Failure: Deferred Authentication processing was requested, but the origin could not locate the conversation′s security context.

FF00 Access Failure: The request specifies a password that is expired.

FF01 Access Failure: The request specifies a password that is invalid.

FF02 Access Failure: The request specifies a user ID that is revoked.

FF03 Access Failure: The request specifies a user ID that is invalid.

FF04 Access Failure: The request is missing a user ID.

FF05 Access Failure: The request is missing a password.

FF06 Access Failure: The request specifies a group that is invalid.

FF07 Access Failure: The request specifies a user ID that is revoked in the specified group.

FF08 Access Failure: The request specifies a user ID that is not defined in the specified group.

FF09 Access Failure: The request specifies a user ID that is not authorized to access the remote LU.

FF0A Access Failure: The request specifies a user ID that is not authorized to access the remote LU from the local LU.

FF0B Access Failure: The request specifies a user ID that is not authorized to access the transaction program at the remote LU.

FF0C Access Failure: The request failed as a result of installation exit processing at the remote LU.

FF0D Access Failure: The request failed because of a processing failure between the local LU and the remote LU. This is a correctable error, so subsequent requests might succeed.

FF0E – FFFF Reserved for future needs for security.

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.

0006 Unsuccessful Allocation: The intermediate session router or boundary function lacks resources to support a session connector. The RU being rejected is a BIND.

0007 Insufficient resources are available for LU address allocation.

0008 No Buffer Space: The session was deactivated because of a buffer shortage when extending a nonextended positive RSP(BIND). Insufficient resources exist to extend a BIND response.

0009 No unreserved session connectors are available to add an LU.

000A A network node does not have adequate resources to honor a Register request (the available directory capacity has already been reached).

000B A BFSESSINFO was received for an unknown LU.

000C Not enough buffer space exists to support a deadlock-free transmission group. The receiver does not have enough buffers to allocate a BIND receive buffer.

000D Insufficient buffers exist to activate a session.

000E A BIND or BFCINIT cannot be processed because of insufficient storage to keep the network-qualified name of the initiating control point.

000F Insufficient buffer space exists to build a BFINIT.

0010 The CP does not have adequate resources to process a GDS variable request; it will deactivate its CP-CP sessions with the partner CP.

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

0012 No memory is available to establish the timer queue specified in the request.

0014 This session has failed because of storage depletion at an intermediate node.

0015 Insufficient resources are available to initiate a short-hold mode connection.

0016 Unknown network identifier

0017 Insufficient buffer space exists to process a nonimmediate UNBIND.

0018 All LFSIDs this node is allowed to assign on the TG are in use at this time; the request is rejected.

0019 Insufficient storage is available to conduct an XID exchange.

001A Insufficient storage is available to activate a TG.

001B Insufficient resources to activate a token-ring connection.

001C Insufficient storage exists to respond precisely to an error condition.

Note: This sense data is returned when node buffer resources are in critical depletion and storage cannot be obtained to build a more specific error response.

001D The receiving T4 node does not have sufficient disk space to perform the requested dump.

001E A session has failed because depletion of pooled buffer storage has exceeded a critical threshold resulting from that session′s monopolizing usage.

001F Cannot create or access an internal space or object.

0021 A received XID3 cannot be fully processed because the receiver has insufficient storage to keep the network-qualified name of the sender′s control point.

0022 Resource control block allocation error: The limit specified in the pertinent NCP BUILD DYNPOOL suboperand has been reached.

0023 Resource control block allocation error: Buffer usage is too close to the NCP slowdown threshold.

0024 Resource control block allocation error: No network element addresses are available.

0025 Insufficient storage to keep the network-qualified name of the connection network.

0026 The specified TG was not activated because insufficient bandwidth was available on the local port.

0027 The specified TG was not activated because insufficient bandwidth was available on the destination port.

0028 The specified TG was not activated; a switched subnetwork had insufficient resources to accept a connection request.

nxxx Insufficient resources of type n (n = 1 or 2) exist to fully satisfy the received request. The value xxx is a byte offset into the received RU indexing the first specified network element (e.g., PU, LU, or link) for which resources could not be obtained. The type (n) of resource specified is product dependent.

0813 Bracket Bid Reject—No RTR Forthcoming: BID (or BB) was received while the first speaker was in the in-bracket state, or while the first speaker was in the between-brackets state and the first speaker denied permission. RTR will not be sent.

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

0000 No specific code applies.

Note: For LU 6.2, this is the only setting defined.

0001 Bracket Bid Reject: The component was in the in-bracket state when a bracket request was received.

0002 Bracket Bid Reject: The component was in the between-bracket state when a bracket request was received.

0814 Bracket Bid Reject—RTR Forthcoming: BID (or BB) was received while the first speaker was in the in-bracket state, or while the first speaker was in the between-brackets state and the first speaker denied permission. RTR will be sent.

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

0000 No specific code applies.

Note: For LU 6.2, this is the only setting defined.

0815 Function Active: A request to activate a network element or procedure was received, but the element or procedure was already active.

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

0000 No specific code applies.

0001 A session activation request was received by a boundary function to activate a session that was already active.

0002 A session activation request was received by a gateway function to activate a cross-network session that was already active.

0003 Processing for another management services request in progress. Sender should retry the request.

Note: This sense data is sent only by a type 2 node, which may lack sufficient queuing space.

0004 A BIND was received from a T2.1 node when the session is already active; i.e., the LFSID is in use. The receiver rejects the BIND.

0005 An IPL function (the loading or storing of a load module) is in progress.

0006 The short-hold mode connection selected has been recalled on another port.

0007 A session activation request was received by an APPN node to activate a CP-CP session that was already active.

0009 A session activation request was received by an APPN end node to activate a CP-CP session with a network node when a CP-CP session is already active with another network node.

0816 Function Inactive: A request to deactivate a network element or procedure was received, but the element or procedure was not active.

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

0000 No specific code applies.

0001 A session deactivation request was received by an APPN node to deactivate a CP-CP session that was not active.

0817 Link or Link Resource Inactive: A request requires the use of a link or link resource that is not active.

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

0000 No specific code applies.

0001 Link inactive.

0002 Link station inactive.

0003 Switched link connection inactive.

0004 The TG number of the desired link has been renegotiated to a new value; the route cannot be activated.

0005 Service link inactive.

0006 The link between an SNA node and an attached processor is inactive; for example, the connection between the main processor and its attached service processor goes down.

0007 The requested test was not initiated because the link to be tested was put into an inactive state.

0008 The requested test was interrupted because the link to be tested was put into an inactive state.

0009 Transport configuration table entry not active.

4001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0818 Link Procedure in Process: CONTACT, DISCONTACT, IPL, or other link procedure in progress when a conflicting request was received.

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

0000 No specific code applies.

0001,0002 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0003 CONTACT Not Serialized, Retry: An initial CONTACT procedure is in progress and a nonactivation CONTACT was received by the PU. The nonactivation CONTACT is rejected until the initial CONTACT procedure is completed.

0004 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0005 Link problem determination test for a modem in progress.

0006 On-line terminal test in progress.

0007 SDLC link test, level 2, in progress.

0009 The requested test was not initiated because another test was already in progress.

000A An on-line terminal test (OLTT) is active on the service link.

000B SDLC link test, level 2, in progress on the service link.

000C Link problem determination test for a modem on the service link 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, and 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.

0007 A CONTACT, BIND, or ACTLU has been received from an SSCP that has not established ownership of a permanent (system-defined) resource. The resource is not usable until RNAA(Move) has been received.

0008 A CONTACT, BIND, or ACTLU has been received from an SSCP that has not established ownership of a temporary (DR-added) resource. The resource is not usable until RNAA(Add) has been received.

0009–000F Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0010 The request is rejected or fails because the entry point or a target resource in the entry point is already in the state or condition that the request would have produced.

0011 A CONTACT was received specifying APPN in its Connection Support field, but was not preceded by a CONNOUT, a protocol violation.

0012 A CONTACT was received with a Connection Support field value that does not match that in the preceding CONNOUT.

0013 A CONNOUT, specifying LEN in its Connection Support field, was received for a nonswitched link station, a protocol violation.

0014 A CONTACT was received for an intra-FRSE PVC segment subport whose frame-relay port is in a discontacted state. The CONTACT is rejected.

0015 A CONTACT was received for an RNAA-added intra-FRSE PVC segment subport and a SETCV with a FRSE (X′80′) control vector has not been received. The CONTACT is rejected.

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.

0003 A hardware error was indicated for the link.

0005 A power-on request failed.

0006 A processor microcode load request failed.

0007 An operating system load request failed.

0008 Channel device name was not defined.

0009 A processor microcode quiesce request failed.

000A A power-off request failed.

000B The requested test was completed successfully, but the results of the test are not available due to a software failure.

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.

00B8 Hardware error indicated on the service link.

00B9 Service link not defined.

00BA The receiver has an error resulting from a microcode problem that prevents execution of the request.

0n0m An error was detected by the DLC manager of the receiving node during the execution of a management services request. If n=X′A′, the link connection status has not changed from the state previous to the execution; if n=X′B′, the link connection status was modified from the state existing previous to the execution. The error is specified as follows: m=X′1′ for volatile storage error, m=X′2′ for nonvolatile storage (e.g., file access error), m=X′3′ for link connection component (e.g., modem) interface error, and m=X′4′ for unspecified software error conditions.

Sense code specific information settings 0004, 0008, 000C, 0010, 0014, 0018, 0020, 0028, 0030, 0034, 0038, 003C, 0040, 0072, 0098, 00AB, 0100–0109, 0120–0126, 0149, 0189–0191, 0200–0209, 0210–0225, 290, 0291, 07**, and 08** are all set aside for implementation-specific

use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

081D Invalid Network Address or Name: A node, station, or CP identifier in the request was found to be invalid.

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

0000 The station ID or SSCP ID in the request was found to be invalid.

0001 The network ID, LU name pair in the request was found to be invalid.

0002 Invalid resource name found in the request.

0003 The network ID, SSCP name pair in the request was found to be invalid.

0004 A duplicate CP name has been detected, causing the links to one or both of the nodes having the same CP name to be deactivated.

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 Request Was Canceled or Not Allowed by an Operator.

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

0000 No specific code applies.

0001 The operator has either canceled the link activation request or deactivated the link.

0002 The operator at the node sending this sense data value has placed the link in a nonactivatable state so that it currently cannot be activated.

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

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

0000 No specific code applies.

0001 In a SETCV defining an intra-FRSE PVC subport set, one or both of the primary subport partners that define the subport set are not defined.

0002 In a SETCV defining an intra-FRSE PVC subport set, a specified element address does not define a subport within a subport set, or is defined more than once in a subport set.

0003 An element address of an intra-FRSE PVC subport set received in a SETCV was found to be already associated with another subport set.

0821 Invalid Session Parameters: Session parameters were not valid or not supported by the half-session whose activation was requested.

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

0000 No specific code applies.

0001 Invalid Mode Name at LU: The specified mode name was not recognized by the LU.

0002 Invalid Mode Name at CP: The specified mode name was not recognized by the CP.

0003 The primary half-session requires cryptography, but the secondary half-session does not support cryptography.

0004 The secondary half-session requires cryptography, but the primary half-session does not support cryptography.

0005 Selective or mandatory cryptography is specified, but no SLU cryptographic data key is provided.

0006 The BIND was rejected because it was nonnegotiable and specified a primary send pacing window size larger than the SSCP or BF can handle.

0007 The specified mode name was not recognized in a subarea network.

0008 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0009 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

000A Specified mode table name not found at receiving CP.

000B Sent on an UNBIND to signal that a RSP(BIND) was received indicating an insufficient level of compression support.

000C Sent on an UNBIND to signal that a BIND was received indicating an insufficient level of compression support.

000D PLU requires Triple DES level of encryption, but SLU has only provided DES level encryption keys or can only operate at DES level of encryption.

000E SLU requires Triple DES level of encryption, but PLU can only operate at DES level of encryption.

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

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

0000 No specific code applies.

0001 An IPLINIT request was received by a T4 node, resulting in a link disconnection. The sender should reinitialize the dial connection and resend the IPLINIT request.

0002 An ACTLINK for a token-ring logical connection was rejected because an active token-ring logical connection exists with the same MAC and SAP addresses.

80nn nn is product-specific and will not be otherwise defined in SNA.

0823 Unknown Control Vector: The control vector specified by a network address and key is not known to the receiver.

0824 Logical Unit of Work Aborted: The current unit of work has been aborted; when sync point protocols are in use, both sync point managers are to revert to the previously committed sync point.

For LU 6.2, this sense data is sent only in FMH-7.

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

0000 For LU 6.2, Backout Initiated—No Resync In Progress: A transaction program or its LU has initiated backout. The protected resources for the distributed logical unit of work are to be restored to the previously committed sync point.

When sent in reply to a PS header, no resync in progress means that all resources in the transaction subordinate to the backout sender have backed out.

For non-LU 6.2, no specific code applies.

0001 For LU 6.2, Backout Initiated — Resync In Progress: A transaction program or its LU has initiated backout. The protected resources for the distributed logical unit of work are to be restored to the previously committed sync point.

When sent in reply to a PS header, resync in progress means that one or more resources in the transaction subordinate to

the backout sender have experienced failure so it is not known whether they′ve backed out.

0825 Component Not Available: The LU component (a device indicated by an FM header) is not available.

0826 FM Function Not Supported: A function requested in an FMD RU is not supported by the receiver. (*Note*: X′1003′ has displaced this value for reporting such conditions.)

0827 Intermittent Error—Retry Requested: An error at the receiver caused an RU to be lost. The error is not permanent, and retry of the RU (or chain) is requested.

0828 Reply Not Allowed: A request requires a normal-flow reply, but the outbound data flow for this half-session is quiesced or shut down, and there is no delayed reply capability.

0829 Change Direction Required: A request requires a normal-flow reply, but the half-duplex flip-flop state (of the receiver of the request) is not-send, and CD was not set on the request. Therefore, there is no delayed reply capability.

082A Presentation Space Alteration: Presentation space altered by the end user while the half-duplex state was not-send, (¬ S,*R); request executed.

082B Presentation Space Integrity Lost: Presentation space integrity lost (for example, cleared or changed) because of a transient condition—for example, because of a transient hardware error or an end user action such as allowing presentation services to be used by the SSCP. (*Note*: The end-user action described under X′082A′ and X′084A′ is excluded here.)

082C Resource-Sharing Limit Reached: The request received from an SSCP was to activate a half-session, a link, or a procedure, when that resource was at its share limit.

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).

0002 Invalid Request: The specified PU has already received an ACTPU and is therefore under the control of another SSCP. This ACTPU 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.

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.

082E Intervention Required at LU Subsidiary Device: A condition requiring intervention, such as out-of-paper, power-off, or cover interlock open, exists at a subsidiary device.

082F Request Not Executable because of LU Subsidiary Device: The requested function cannot be executed, because of a permanent error condition in one or more of the receiver′s subsidiary devices.

0830 Session-Related Identifier Not Found: The receiver could not find a session-related identifier for a specified session.

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

0000 No specific code applies.

0001 PCID not found for the specified resources.

0002 LSID not found for the specified session.

0831 LU Component Disconnected: An LU component is not available because of power-off or some other disconnecting condition.

0832 Invalid Count Field: A count field contained in the request indicates a value too long or too short to be interpreted by the receiver, or the count field is inconsistent with the length of the remaining fields.

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

nnnn Bytes 2 and 3 contain a binary count that indexes (0-origin) the first byte of the invalid count field.

Note: This sense code is not used for a BIND error because the displacement of fields within the BIND may not be the same at both ends of a session when the BIND was affected by name transformations—for example, after the BIND has passed through a gateway. Sense code X′0835′ is used to specify a displacement for a BIND error.

0833 Invalid Parameter (with Pointer and Complemented Byte): One or more parameters contained in fixed- or variable-length fields of the request are invalid or not supported by the NAU that received the request.

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

nnmm Byte 2 contains a binary value that indexes (0-origin) the first byte that contained an invalid parameter.

Byte 3 contains a transform of the first byte that contained an invalid parameter: the bits that constitute the one or more invalid parameters are complemented, and all other bits are copied.

Note: This sense code is not used for a BIND error because the displacement of fields within the BIND may not be the same at both ends of a session when the BIND was affected by name transformations—for example, after the BIND has passed through a gateway. Sense code X′0835′ is used to specify a displacement for a BIND error.

0834 RPO Not Initiated: A power-off procedure for the specified node was not initiated because one or more other SSCPs have contacted the node, or because a CONTACT, DUMP, IPL, or DISCONTACT procedure is in progress for that node.

0835 Invalid Parameter (with Pointer Only): The request contained a fixed- or variable-length field whose contents are invalid or not supported by the NAU that received the request; or a +RSP(BIND) received at an intermediate node within a session path contained such an error (in which case, it was converted to a -RSP or UNBIND carrying this sense data value)

nnnn Bytes 2 and 3 contain a two-byte binary count that indexes (0-origin) the first byte of the fixed- or variable-length field having invalid contents.

Note: This sense code is not used to report an invalid value in an MS major vector. If the invalid value occurs in a formatted MS subvector, sense code X′086B′ is used. If it occurs in an unformatted subvector, sense code X′0870′ is used.

0836 PLU/SLU Specification Mismatch: For a specified LU-LU session, both the origin LU (OLU) and the destination LU (DLU) have only the primary capability or have only the secondary capability.

0837 Queuing Limit Exceeded: For an LU-LU session initiation request (INIT, CDINIT, or INIT-OTHER-CD) specifying (1) Initiate or Queue (if Initiate not possible) or (2) Queue Only, the queuing limit of either the OLU or the DLU, or both, was exceeded.

0838 Request Not Executable Because of Resource or Component State Incompatibility: The request is not executable because it is not compatible with the state of a resource or component in the receiver.

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

0000 No specific code applies.

0001 The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be deleted or replaced because it is installed marked removable.

0002 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed, removed, or accepted because they are in back-level state (see Note).

0003 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked on-trial because they are already installed marked on-trial (see Note).

0004 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked on-trial or in-production because they are already installed marked in-production removably (see Note).

0005 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked on-trial or in-production because they are already installed marked in-production and nonremovable (see Note).

0006 One or more of the changes referred to in a Request Change Control MS major vector cannot be removed or accepted because they are installed marked nonremovable (see Note).

0007 One or more of the changes referred to in a Request Change Control MS major vector cannot be removed or accepted because they are not installed (see Note).

0008 Pre-test is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

0009 Execution window timing is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000A Automatic removal is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000B Post-test is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000C Automatic delayed acceptance is not applicable to one or more of the changes referred to in a Request Change Control MS major vector (see Note).

000D One or more of the changes referred to in a Request Change Control MS major vector cannot be installed marked in-production because they are installed marked on-trial with a set of corequisites different from those requested on this install request.

One or more reported-on token strings are used to identify the corequisite changes currently installed when the report code is carried in an SNA condition report.

000E One or more of the changes referred to in a Request Change Control MS major vector cannot be accepted because they are installed marked on-trial (see Note).

000F One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be replaced or deleted because they are critical system components that must always have an installed instance. The only possibility is to perform data object renewal using Send-and-

Install with removability prohibited or desired (but not required) (see Note).

0010 One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be stored or installed because an implementation-defined limit on the number of changes has been exceeded (see Note).

0011 One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be deleted or replaced because they are required in order to maintain removability of other changes. They may be in backup state or installed marked in-production (see Note).

0012 One or more of the corequisite changes referred to in a Request Change Control MS major vector are missing or are in a state incompatible with the request (see Note).

0013 The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be replaced because it is installed marked in-production and non-removable and another change is not being installed in this operation (see Note).

0014 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed because a precluded combination of values in the Removability, Automatic Removal, Automatic Acceptance, or Activation Use subfields was specified (see Note).

0015 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed because one or more changes already installed are still removable for one or more components to be altered by these changes (see Note).

0016 One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be replaced because they would be required for removable installation, and removability is required (see Note).

0017 Execution of the request referred to in an MS Cancel major vector has proceeded too far to cancel.

001A The request will not be honored because it was either queued or active at a node at the time a local operator assumed control of the node, thus effecting its cancellation.

001B The request will not be honored because it was submitted to a node at a time when a local operator or other application was in control of the node.

001C One or more of the changes referred to in a Request Change Control MS major vector cannot be installed removably because the implementation does not support removability for certain classes of data objects (see Note).

001D One or more of the changes referred to in a Request Change Control MS major vector cannot be installed because the implementation precludes corequisite specification for certain classes of data objects (see Note).

001E One or more of the changes referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be installed or stored because the implementation has identified a prerequisite change that is either not installed or is installed at an incompatible level (see Note).

001F The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be stored because the implementation has identified a missing larger entity that must first be stored before the subentity may be stored (see Note).

0020 One or more of the changes referred to in a Request Change Control MS major vector are installed in production but cannot be removed or accepted because they are required in order to maintain removability of other changes (see Note).

0021 One or more of the changes referred to in a Request Change Control MS major vector cannot be installed, removed, or accepted because the implementation does not support certain classes of data objects (see Note).

0022 The change referred to in a Report-FS-Action command cannot be stored because another change having the same data object class already exists in sent state and the implementation prohibits more than one change of the same object class to exist in sent state for certain classes of objects. The previous change must first be installed or deleted.

A reported-on token string is used to identify the change currently in sent state when the report code is carried in an SNA condition report.

0023 The request will not be honored because a system resource file was locked at the time execution was attempted.

0024 One or more of the changes referred to in a Request Change Control MS major vector could not be installed because an unexpected error was encountered while performing the installation process (see Note).

0025 The change referred to in a Request Change Control MS major vector or Report-FS-Action command cannot be stored because a reactivation of the node must first be performed (see Note).

0026 The target group object of one or more changes referred to in a Request Change Control MS major vector does not exist (see Note).

0027 The target group object of one or more changes referred to in a Request Change Control MS major vector is not accessible (see Note).

0028 The class code of one or more changes referred to in a Request Change Control MS major vector is inconsistent with the class code of the target group object for the affected component (see Note).

0029 The installation of one or more changes referred to in a Request Change Control MS major vector would require nesting of group objects, which is not supported (see Note).

002A The class code of one or more changes referred to in a Request Change Control MS major vector is inconsistent with the class code of installed changes affecting the same component (see Note).

002B The installation procedure for one or more changes referred to in a Request Change Control MS major vector was not found (see Note).

002C The command processor to execute the installation procedure for one or more changes referred to in a Request Change Control MS major vector could not be found (see Note).

002D The name of one or more changes referred to in a Request Change Control MS major vector does not contain an architecturally defined subtree as required by the receiver (see Note).

002E The specification of Alter_Active_Components=No is not supported by the receiver for the class code of one or more changes referred to in a Request Change Control MS major vector (see Note).

002F No group object is defined, and either insufficient or illegal default group information was provided for one or more changes referred to in a Request Change Control MS major vector (see Note).

0030 One or more of the changes referred to in a Request Change Control MS major vector cannot be removed (or accepted) because a previous Accept (or Remove) request has failed and its effects could not be backed out. Only a retry of the same request can be attempted (see Note).

0031 One or more of the changes referred to in a Request Change Control MS major vector are not independent and can not be installed as corequisites (see Note).

0032 The modification level of one or more changes referred to in a Request Change Control MS major vector is inconsistent with the modification level of installed changes affecting the same component (see Note).

0033 The version tokens of one or more changes referred to in a Request Change Control MS major vector are inconsistent with the version tokens of installed changes affecting the same component (see Note).

0034 The request will not be honored because it was canceled by the operating system at the node to which it was sent.

0035 The class code specified in a Request Change Control MS major vector for one or more changes is inconsistent with the class code specified in the local FS catalog for the same object (see Note).

0036 Access to one or more local files associated with one or more changes referred to in a Request Change Control MS major vector was denied (see Note).

0039 Queuing not supported

003A The requested function cannot be completed because the specified adjacent node′s CP Capabilities GDS variable does not indicate support for the complementary function.

003B The number of target installation parameters in the system-specific information file exceeds the implementation limit.

003C The number of target hardware parameters in the system-specific information file exceeds the implementation limit.

003D The internal version of the named change file is not supported by the current target agent.

003E The prerequest command specified in the named change file has failed.

003F The postrequest command specified in the named change file has failed.

0040 One or more substitution tokens were unresolved within a target file specification contained in the named change file.

0041 The request-specific customization controlling the pre-change management phase specified in the named change file has failed.

0042 The request-specific customization controlling the post-change management phase specified in the named change file has failed.

0043 The request-specific customization controlling the pre-change management phase specified in the named change file was not found.

0044 The request-specific customization controlling the post-change management phase specified in the named change file was not found.

0045 One or more AIX licensed program product options have been applied or committed either outside the change management product process or as a part of another change file.

0046 One or more AIX licensed program product options are in a state in which a re-installation of the named change file should be performed.

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.

0004 A service TP send failure was detected by the CP, causing both contention-winner and contention-loser CP-CP sessions to be deactivated. (Retired)

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 or route setup procedure duplicated the PCID assigned to another session or route setup procedure, the received PCID intended as an identifier for an existing session or route setup procedure could not be associated with such an existing session or route setup procedure, or an error was detected in the format of the received PCID.

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

0000 No specific code applies.

0001 The PCID contained in CDINIT(Initiate or Queue), INIT-OTHER-CD, or CDTAKED duplicates a PCID received previously in one of these requests.

0002 The received fully-qualified PCID duplicated one assigned to another session.

0003 The received fully-qualified PCID contains a network-qualified CP name identical to that of the receiving node.

0004 The received fully-qualified PCID duplicated one assigned to another route setup procedure.

0005 The fully-qualified PCID received in BFCINIT is not assigned to an existing route setup procedure. The BFCINIT is rejected.

0006 The fully-qualified PCID received in BFCLEANUP is not assigned to an existing route setup procedure. The BFCLEANUP is rejected.

083C Domain Takedown Contention: While waiting for a response to a CDTAKED, a CDTAKED request is received by the SSCP containing the SSCP-SSCP primary half-session. Contention is resolved by giving preference to the CDTAKED sent by the primary half-session.

083D Dequeue Retry Unsuccessful—Removed from Queue: The SSCP cannot successfully honor a CDINIT(Dequeue) request (which specifies "leave on queue if dequeue-retry is unsuccessful") to dequeue and process a previously queued CDINIT request (for example, because the LU in its domain is still not available for the specified session), and removes the queued CDINIT request from its queue.

083E Implementation-Defined Retry Limit Exhausted.

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

0000 No specific code applies.

0001 The implementation-defined limit on XID exchanges was exceeded before link activation completed.

0002 The implementation-defined limit on XID exchanges was exceeded before a nonactivation exchange completed.

0004 The implementation-defined limit on contention-winner CP-CP session activation attempts has been exceeded.

083F Terminate Contention: While waiting for a response to a CDTERM, a CDTERM is received by the SSCP of the SLU. Contention is resolved by giving preference to the CDTERM sent by the SSCP of the SLU.

0840 Procedure Invalid for Resource: The received RU is not supported in the receiver for this type of resource (for example, (1) SETCV specifies boundary function support for a type 1 node but the capability is not supported by the receiving node, or (2) the PU receiving an EXECTEST or TESTMODE is not the primary PU for the target link.)

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

0000 No specific code applies.

0001 Surrogate session setup failed.

0002 Invalid Link: A BIND was received over a subarea link, but the next hop is over a port that supports only HPR links. The receiver does not support this configuration.

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.

0007 Resource Not Found: A Delete or Find could not be satisfied because the specified entry does not exist in the receiver′s directory.

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

000A Procedure Invalid for Resource: A node supporting independent LUs has dialed into a subarea boundary node that does not support sessions with independent LUs.

000C Conflicting Entry Type on Delete: The Delete request attempted to delete a home entry, i.e., one defined at the receiver by its own network operator facility (NOF).

0010 A SETCV with control vector X′43′ has been received for a nonswitched resource.

0011 A dynamically added or a switched resource has not yet been activated.

0012 A request was received that is allowed only for a primary link station. The request must use the service link, and that link is defined as secondary.

0013 A CONNOUT request was received that contained an invalid X.21 call type.

0014 A CONNOUT or CONTACT was received specifying that the receiver is to designate itself as an APPN end node in XID3s that it sends to an attached APPN or LEN node, but the receiver does not support this option.

0015 An APPN session route must be calculated in two pieces (using two separate RSCVs) but it is determined that the two RSCVs identify a common node; that is, the session route passes through a given node twice.

0016 An RSCV is precalculated because the OLU or DLU was thought to be in a subarea network but it is determined (based on the RSCV) that the location of the DLU is incorrect; that is, the RSCV indicates that the DLU is in the APPN network, but the DLU is really in a subarea network, or vice versa.

0017 An ENCP(DLU) receiving a Locate request with a precalculated RSCV detects that the route was calculated using an incorrect set of the ENCP′s TG vectors; it stores this sense data value locally and returns a Locate reply indicating “resubmit via a directed Locate” along with the correct TG vectors.

0841 Duplicate Network Address: In an LU-LU session initiation request, one of the specified LUs has a duplicate network address already in use.

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

0000 The SSCP of the DLU determines that the OLU network address specified in the CDINIT request is a duplicate of an LU network address assigned to a different LU name.

0001 A duplicate SLU address is found during session initiation.

0002 A duplicate PLU address is found during session initiation.

0003 An SSCP finds a duplicate network address for the DLU on the OLU side of the gateway.

0004 An SSCP finds a duplicate network address for the DLU on the DLU side of the gateway.

0005 An SSCP finds a duplicate network address for the OLU on the OLU side of the gateway.

0006 An SSCP finds a duplicate network address for the OLU on the DLU side of the gateway.

0008 An ACTCDRM request was received that contained a network address already in use.

0842 Session Not Active.

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

0000 SSCP-SSCP Session Not Active: The SSCP-SSCP session, which is required for the processing of a network services request, is not active; for example, at the time an LU-LU session initiation or termination request is received, at least one of the following conditions exists:

- The SSCP of the ILU and the SSCP of the OLU do not have an active session with each other, and therefore INIT-OTHER-CD cannot flow.
- The SSCP of the OLU and the SSCP of the DLU do not have an active session with each other, and therefore CDINIT or CDTERM cannot flow.

Note: This value is used if there is not enough data to select one of the more specific codes listed below.

0001 The session between T2.1 CPs is not active.

0002 For a session-initiation request, an SSCP does not have an SSCP-SSCP session with an SSCP in the direction of the DLU.

0003 For a session-initiation request, an SSCP does not have an SSCP-SSCP session with an SSCP in the direction of the OLU.

0004 An intermediate SSCP has lost connectivity with an SSCP in the session setup path for an LU-LU session. This sense data is used when the SSCP previously lost connectivity with one or more participating gateway nodes so that it cannot learn that the LU-LU session is ended by receiving a NOTIFY RU from a gateway node.

FFFF The session is not active because the session initiation request has been transferred to another PLU.

0843 Required Synchronization Not Supplied: For example, a secondary LU (LU type 2 or 3) received a request with Write Control Code = Start Print, along with RQE and ¬CD.

0844 Initiation Dequeue Contention: While waiting for a response to a CDINIT(Dequeue), a CDINIT(Dequeue) is received by the SSCP of the SLU. Contention is resolved by giving preference to the CDINIT(Dequeue) sent by the SSCP of the SLU.

0845 Permission Rejected—SSCP Will Be Notified: The receiver has denied an implicit or explicit request of the sender; when sent in response to BIND, it implies that the secondary LU will notify the SSCP (via NOTIFY vector key X′0C′) when a BIND can be accepted, and the SSCP of the SLU supports the notification. (See the X′080A′ sense code for a contrasting response.)

0846 ERP Message Forthcoming: The received request was rejected for a reason to be specified in a forthcoming request.

0847 Restart Mismatch: Sent in response to STSN, SDT, or BIND to indicate that the secondary half-session is trying to execute a resynchronizing restart but has received insufficient or incorrect information.

0848 Cryptography Function Inoperative: The receiver of a request was not able to decipher the request because of a malfunction in its cryptography facility.

0849 User Names Lost: An exception condition has resulted in the loss of user names associated with the identified message unit.

084A Presentation Space Alteration: The presentation space was altered by the end user while the half-duplex state was not-send, (¬S,*R); request not executed.

084B Requested Resources Not Available: Resources named in the request, and required to honor it, are not currently available. It is not known when the resources will be made available.

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

0000 No specific code applies.

0002 Requested Resource Not Available: For dynamic reconfiguration MOVE, ADD, or ADDLIKE operation, the requested local address is already assigned to an active resource. For MOVE PU this is the DLC address; for MOVE LU, the LU local address.

0003 The application transaction program specified in the request is not available.

0004 Session Resources Unavailable: The receiver of the RNAA cannot satisfy the request for reserved session resources specified on the Assign LU Characteristics (X′30′) control vector.

0005 Controller resource is not available.

6002 The resource identified by the destination program name (DPN) is not supported.

6003 The resource identified by the primary resource name (PRN) is not supported.

6031 Transaction Program Not Available—Retry Allowed: The FMH-5 Attach command specifies a transaction program that the receiver is unable to start. Either the program is not authorized to run or the resources to run it are not available at this time. The condition is temporary. The sender is responsible for subsequent retry. This sense data is sent only in FMH-7.

084C Permanent Insufficient Resource: Receiver cannot act on the request because resources required to honor the request are permanently unavailable. The sender should not retry immediately because the situation is not transient.

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

0000 For LU 6.2, Transaction Program Not Available—No Retry: The FMH-5 Attach command specifies a transaction program that the receiver is unable to start. The condition is not temporary. The sender should not retry immediately. This sense data is sent only in FMH-7.

For non-LU 6.2, no additional information is specified.

0001 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0002 Creating Allocation Exception: The receiver is unable to create the specified data object as a result of an insufficient storage condition that occurred at allocation time. When this SNA report code is used in an SNA condition report, it is optionally accompanied by one or more structure reports that identify the allocation requests that failed.

0003 Replacing Allocation Exception: The receiver is unable to replace the specified data object as a result of an insufficient storage condition that occurred at allocation time. When this SNA report code is used in an SNA condition report, it is optionally accompanied by one or more structure reports that identify the allocation requests that failed.

0004 Reserved

0005 Reserved

0006 Data-Object Storing Exception: The receiver is unable to store the specified data object as a result of an insufficient storage condition that occurred during the storing process. When this SNA report code is used in an SNA condition report, it is optionally accompanied by one or more structure reports that identify containing the allocation requests that failed.

0007 Data-Object Classification Code Not Supported: The receiver is unable to satisfy the allocation requirements of the specified data-object classification code. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report containing the data-object classification code that failed.

0008 Volume Not Mounted: The receiver is unable to perform the requested allocation/storing operation because the required volume is not mounted. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report identifying the volume that was not mounted.

0009 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

hnnn where h≥8, i.e., the high-order bit in byte 2 is set to 1. The 15 low-order bits of bytes 2 and 3 contain a binary count that indexes (zero-origin) the first byte of the field found to be in error.

084D (Retired) Invalid Session Parameters for Boundary Function.

084E Invalid Session Parameters—PRI: A positive response to an activation request (for example, BIND) was received and was changed to a negative response because of invalid session parameters carried in the response. The LU receiving the response will send a deactivation request for the corresponding session.

084F Resource Not Available: A requested resource is not available to service the given request.

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

0000 No specific code applies.

0001 The receiver's disk is full; therefore, a received load module cannot be stored.

0002 Security component not available: The security component required to process the request is currently not available. This sense data is sent only in UNBIND and -RSP(BIND) records.

0003 A coded graphics character set ID (CGCSID) needed to interpret the request is not supported by the receiver. When this report code is used in an SNA condition report, it is accompanied by a supplemental report containing the 2-byte CGCSID not supported.

0850 Link-Level Operation Cannot Be Performed: An IPL, dump, or RPO cannot be performed through the addressed link station because the system definition or current state of the hardware configuration does not allow it.

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

0000 No specific code applies.

0001 Link Activation Limit Reached: The specified TG was not activated because the maximum number of active link stations allowed on this port has already been reached.

0851 Session Busy: Another session that is needed to complete the function being requested on this session is temporarily unavailable.

0852 Duplicate Session Activation Request: Two session activation requests have been received with related identifiers. The relationship of the identifiers and the resultant action varies by request.

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.

0002 A REQACTPU has been received by an SSCP that has already sent an ACTPU for the same PU.

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 Route Setup Procedure Failure: An intermediate or destination node was unable to successfully complete the processing of a high-performance routing (HPR) Route Setup request or reply.

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

0000 No specific code applies.

0001 The destination LU is not ready to accept sessions.

0002 An unknown destination LU was specified in the request.

0003 A TG specified in the RSCV could not be activated.

0004 An unknown TG was specified in the RSCV.

0005 A TG specified in the RSCV has failed.

0006 A TG specified in the RSCV was not activated as an HPR TG.

0007 An intermediate node had insufficient storage to activate a TG specified in the RSCV.

0008 The receiving node had insufficient storage to process the Route Setup request.

0009 The Route Setup request was received over a TG that was not activated as an HPR TG.

000A A VR within a composite network node is inoperative.

000B The receiving node does not support HPR protocols.

000D The Route Setup request or reply could not be forwarded over the TG in the FID2 format because its size was greater than the TG′s maximum BTU size.

000E The value of the Current Hop Count field of the RSCV received in a Route Setup request exceeded the value of the Destination Hop Index field in the request (i.e., the Route Setup request appeared to have passed the destination node.)

000F The Route Setup reply was not received in the allotted time (i.e., the Route Setup timer expired).

0020 The intended destination was not able to perform the RTP function (e.g., it does not support the RTP option set).

0856 SSCP-SSCP Session Lost: Carried in the Sense Data field in a NOTIFY (Third-Party Notification vector, X′03′) or -RSP(INIT_OTHER) sent to an ILU to indicate that the activation of the LU-LU session is uncertain because the SSCP(ILU)-SSCP(OLU) session has been lost. (Another sense code, X′0842′, is used when it is known that the LU-LU session activation cannot be completed.)

0857 SSCP-LU Session Not Active: The SSCP-LU session, required for the processing of a request, is not active; for example, in processing REQECHO, the SSCP did not have an active session with the target LU named in the REQECHO RU.

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

0000 No specific code applies.

0001 The SSCP-SLU session is in the process of being reactivated.

0002 The SSCP-PLU session is inactive.

0003 The SSCP-SLU session is inactive.

0004 The SSCP-PLU session is in the process of being reactivated.

0005 The SSCP lost connectivity with the PLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0006 The SSCP lost connectivity with the SLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0007 The selected ALS for the OLU is not in a state permitting LU-LU sessions to be established using it. The condition is detected when the session request (BFINIT) was received, but, when the request was processed, the ALS was no longer in an active state. The session request is rejected.

0008 The selected ALS for the DLU is not in a state permitting LU-LU sessions to be established using it. The condition is detected when the session request was being processed in the DLU domain and the ALS selected for the DLU is no longer in an active state. The session request is rejected.

0858 SSCP-SSCP Session Activation Rejected.

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

0000 No specific code applies.

0001 An SSCP rejects a received ACTCDRM attempting to restart a session that terminated as a result of an operator-initiated non-disruptive deactivation request.

0859 REQECHO Data Length Error: The specified length of data to be echoed (in REQECHO) violates the maximum RU size limit for the target LU.

085A Specific Server Exception: An architecturally defined or user-defined server that is sensitive to data object contents, has detected an exception.

085B Unknown Resource Name: The identified resource, required to complete the requested unit-of-work, is not known to the SNA node.

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

0000 No specific code applies.

0001 Unknown server name. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report containing the server name.

0002 Unknown agent.

0003 The clock identifier specified in an MS Set Clock major vector is unknown to the receiver.

0004 The timing source name specified in an MS Set Clock major vector is unknown to the receiver.

0005 The agent unit-of-work correlator referred to by an MS Cancel major vector is unknown to the receiver, or represents a unit of work already completed.

0006 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

085C System Exception: The node experiences an exception condition within a resident system or subsystem that inhibits subsequent processing by the SNA component.

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

0000 No specific code applies.

0001 The exception is identifiable as a system-related problem.

0002 The exception is identifiable as a permanent system-related problem.

0003 The system has logged additional information that should be used when reporting this defect.

085D MU-ID Could Not Be Accepted in the MU-ID Registry.

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

0001 The MU-ID is a duplicate. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify information about the receiver′s MU-ID registry: supplemental report 1 contains the lowest MU-ID the receiver would accept; supplemental report 2 contains the highest MU-ID the receiver would accept; supplemental report 3 contains the time stamp of the receiver′s MU-ID registry.

0002 The MU-ID value is greater than expected. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify information about the receiver′s MU-ID registry: supplemental report 1 contains the lowest MU-ID the receiver would accept; supplemental report 2 contains the highest MU-ID the receiver would accept; supplemental report 3 contains the time stamp of the receiver′s MU-ID registry.

0003 A temporary condition prevents acceptance of the MU-ID.

0004 A permanent condition prevents acceptance of the MU-ID.

0005 The MU-ID registry is not initialized.

085E Operator Intervention

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

0000 No specific code applies.

0001 The operator has suspended the transmission of the message unit.

0002 The operator has purged the message unit.

0860 Function Not Supported—Continue Session: The function requested is not supported; the function may have been specified by a request code or some other field, control character, or graphic character in an RU.

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

nnnn Bytes 2 and 3 contain a 2-byte binary count that indexes (0-origin) the first byte in which an error was detected. This sense data is used to request that the session continue, thereby ignoring the error.

0861 Invalid COS Name: The class-of-service (COS) name, either specified by the ILU or generated by the SSCP of the SLU from the mode table is not in the “COS name to VR identifier list” table used by the SSCP of the PLU.

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

0000 COS name was generated by the SSCP.

0001 COS name was generated by the ILU.

0002 The COS name generated by the T2.1 CP local to, or the T2.1 NNCP server for, the ILU is not in the COS name definition table.

0003 The CDINIT request or response contains a Session Initiation (X′14′) control vector that has Class-of-Service (COS) Name fields that have not been properly specified.

0862 Medium Presentation Space Recovery: An error has occurred on the current presentation space. Recovery consists of restarting at the top of the current presentation space. The sequence number returned is of the RU in effect at the top of the current presentation space.

nnnn Bytes 2 and 3 following the sense code contain the byte offset from the beginning of the RU to the first byte of the RU that is displayed at the top of the current presentation space.

0863 Referenced Local Character Set Identifier (LCID) Not Found: A referenced character set does not exist.

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

0000 No specific code appplies.

hnnn where h≥8, i.e., the high-order bit in byte 2 is set to 1. The 15 low-order bits of bytes 2 and 3 contain a binary count that indexes (0-origin) the first byte of the field found to be in error.

0864 Function Abort: The conversation was terminated abnormally. Other terminations may occur after repeated reexecutions; the request sender is responsible to detect such a loop.

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

0000 For LU 6.2, Premature Conversation Termination: The conversation is terminated abnormally; for example, the transaction program may have issued a DEALLOCATE_ABEND verb, or the program may have terminated (normally or abnormally) without explicitly terminating the conversation.

For LU 6.2 half-duplex conversations, this sense data is sent only in FMH-7 or UNBIND.

For LU 6.2 full-duplex conversations, this sense data is sent also in the negative response that precedes an FMH-7 when there is a chain to respond to. The sense data in the negative response results in advance notice to the transaction program (in the form of an error return code) that an ERP message is forthcoming. For this error, the ERP message will contain the same sense data value used in the negative response.

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.

For LU 6.2 half-duplex conversations, this sense data is sent only in FMH-7 or UNBIND.

For LU 6.2 full-duplex conversations, this sense data is sent also in the negative response that precedes an FMH-7 when there is a chain to respond to. The sense data in the negative response results in advance notice to the transaction program (in the form of an error return code) that an ERP message is forthcoming. For this error, the ERP message will contain the same sense data value used in the negative response.

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.

For LU 6.2 half-duplex conversations, this sense data is sent in UNBIND when there is no chain to respond to; otherwise, it is sent in FMH-7.

For LU 6.2 full-duplex conversations, this sense data is sent also in the negative response that precedes an FMH-7. The sense data in the negative response results in advance notice to the transaction program (in the form of an error return code) that an ERP message is forthcoming. For this error, the ERP message will contain the same sense data value used in the negative response.

0003 Allocation Error Message Forthcoming: An error has been detected in a received Attach request, resulting in a rejection of the Attach. The sense data value that indicates the reason for rejection will be specified in a forthcoming FMH-7.

This sense data is sent in the negative response that precedes an allocation error FMH-7 for an LU 6.2 full-duplex conversation. The negative response results in advance notice to the transaction program (in the form of an error return code) that an ERP message is forthcoming.

0004 Don′t-Know Signal During Sync Point: The final outcome of the sync point request is in doubt. Communication with the sync point coordinator has been lost; therefore, it may be some time before the results are known. The application program is terminated in a system-dependent fashion, and resync processing is used to convey the final sync point results to the sync point managers.

Note: The phrases following the sense data are symbolic return codes provided to a full-duplex transaction program when a negative response with sense data is received by the LU. (See *SNA Transaction Programmer's Reference Manual for LU Type 6.2* for full-duplex verbs and their possible return codes.)

Sense Data	Return Code
08640000	ERROR_INDICATION (with a subcode of DEALLOCATE_ABEND_PROG)
08640001	ERROR_INDICATION (with a subcode of DEALLOCATE_ABEND_SVC)
08640002	ERROR_INDICATION (with a subcode of DEALLOCATE_ABEND_TIMER)
08640003	ERROR_INDICATION (with a subcode of ALLOCATION_ERROR)
08640004	System dependent

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 a control vector or 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 key (nn) of the control vector or the MS subvector containing the subfield with the invalid value, and byte 3 contains the subfield key (mm) of the subfield with the invalid value.

Note: See sense code X′0870′ for the case in which the invalid value occurs in an unformatted subvector, that is, one not containing subfields with keys and lengths, or in the unformatted portion of a partially formatted subvector.

086C Required Control Vector or Subvector Missing: One or more control vectors or MS subvectors that are required by the receiver to perform some function are missing from the received message, or are not present in the required position.

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

nn00 Byte 2 following the sense code contains the key (nn) of one of the control vectors or subvectors that is missing, or improperly positioned. Byte 3 is reserved (00).

Note: See the X′080C0006′ sense data for the case in which the major vector key is recognized but a subvector representing the function to be performed cannot be identified.

086D Required Subfield Missing: A control vector or MS subvector lacks one or more subfield keys that are required by the receiver to perform the function requested.

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

nnmm Byte 2 following the sense code contains the key (nn) of the subvector or control vector lacking a required subfield, and byte 3 contains the subfield key (mm) of a missing subfield.

086E Invalid Subvector Combination: Two or more subvectors, each permissible by itself, are present in a combination that is not allowed.

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

nnmm Bytes 2 and 3 following the sense code contain the subvector keys (nn) and (mm) of two of the subvectors that should not be jointly present.

086F Length Error: A length field within a structure is invalid, or two or more length fields are incompatible.

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

0000 No specific code applies.

0001 The MS major vector length is incompatible with the RU length.

0002 The sum of the MS subvector lengths is incompatible with the MS major vector length.

nn03 The sum of the subfield lengths in a MS subvector is incompatible with the subvector length. Byte 2 following the sense code contains the subvector key (nn).

nn05 MS subvector length invalid. Byte 2 following the sense code contains the relevant subvector key (nn). (This is specified only if the sum of the subvector lengths is compatible with the major vector length.)

nn06 Subfield length invalid. Byte 2 following the sense code contains the subvector key (nn) of the MS subvector containing the invalid subfield length. (This is specified only if the sum of the subfield lengths is compatible with the subvector length.)

0007 The length field of an MDS_MU is incompatible with the sum of the lengths of the imbedded GDS variables or an invalid length was found in an imbedded structure (or GDS variable).

0008 The length field of a CP-MSU is incompatible with the sum of the lengths of the imbedded structures.

nn09 The sum of the subfield lengths in a control vector is incompatible with the control vector length. Byte 2 following the sense code contains the control vector key (nn).

000A The length field of a Route Setup GDS variable is incompatible with the sum of the lengths of the imbedded structures.

000B The sum of the control vector lengths in an RU or XID is incompatible with the length of the RU or XID.

nn0C The length field of a control vector in an RU or XID is invalid. Byte 2 following the sense code contains the control vector key (nn).

0870 Unformatted Subvector Value Invalid: A value in an unformatted MS subvector, or in an unformatted portion of a partially formatted MS subvector, is invalid.

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

nnxx Byte 2 following the sense code contains the subvector key (nn) of the MS subvector containing the invalid value. Byte 3 contains a one-byte binary count that indexes the first byte in which the invalid value falls. The indexing is zero-origin, from the beginning of the subvector.

Note: See sense code X′086B′ for the case in which the invalid value occurs in a formatted MS subvector, that is, one containing subfields with keys and lengths, or in the formatted portion of a partially formatted subvector.

0871 Read Partition State Error: A Read Partition structured field was received while the display was in the retry state.

0872 Explicit or Implied Orderly Deactivation Refused

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

0000 An NC_DACTVR(Orderly) request has been received, but sessions are assigned to the VR and it will not be deactivated.

0001 An MS major vector specifying or implying orderly deactivation of the receiving node has been received, but the node is not in a quiesced state and deactivation is not allowed; the requested action will not proceed.

0002 An MS major vector specifying or implying orderly deactivation of the receiving node has been received, but the receiver cannot determine if a quiesced state has been attained; the requested action will not proceed.

0873 Virtual Route Not Defined: No ERN is designated to support this VRN.

0874 ER Not in a Valid State: The ER supporting the requested VR is not in a state allowing VR activation.

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

0000 No specific code applies.

0001 A gateway node (GWN) received an RNAA from a gateway SSCP but either (1) no operative ER exists in the network adjacent to the GWN on the origin NAU′s side to the subarea containing the origin NAU′s address, or (2) no operative ERs exist at all in the network adjacent to the GWN on the destination NAU′s side.

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 or resource 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 nontailed 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 node 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, or the network ID in the BIND SLU name is not equal to the LU network ID 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 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.

0019 Multiple sets of DLC signaling information were received for a port, but only one set of DLC signaling information is supported by the port.

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 the BFCINIT is a secondary address; the BFCINIT is rejected.

001E The subject LU specified in the 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 the boundary function for a resource defined at system definition time, which is not allowed.

0022 The link for which ACTLINK was issued is a S/390 channel that has been defined for connections only to a T2.1 node. However, the SSCP that sent ACTLINK had previously indicated it does not support T2.1 connections.

0023 Modem test support cannot be changed. The RNAA or SETCV containing the SDLC Secondary Station (X′03′) or the Extended SDLC Secondary Station (X′43′) control vector is rejected.

0024 The data mode cannot be changed. The RNAA or SETCV containing the SDLC Secondary Station (X′03′) or the Extended SDLC Secondary Station (X′43′) control vector is rejected.

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

0026 Link Definition Error: A link is defined as not supporting HPR, but the port only supports HPR links.

0027 A link connection request for a nonempty active link connection configuration was received by the management services element; the active link connection configuration of the DLC element is empty; i.e., it has no link connection components present.

0028 An RNAA(Move) was received for a link station, and the link station′s primary-secondary role is incompatible with the target link.

0029 The RU refers to a resource, and the sender and receiver disagree about its status. One considers it a static resource, the other a dynamic resource.

002A A session cannot be activated because the node does not support segment generation and the maximum link BTU size is too small to satisfy a requirement on the minimum send RU size as defined for the session mode.

002B A session cannot be activated because the node does not support segment reassembly and the maximum link BTU size is too small to satisfy a requirement on the minimum receive RU size as defined for the session mode.

002C A BFINIT session request was received from a PLU that is not in the same network as this SSCP, or a BFSESSINFO was received reporting a subject LU in another network.

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

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

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

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

0031 BFSESSINFO was received for a subject LU, but the receiver has the address associated with anything other than a static LU or an other-domain resource.

0032 BFSESSINFO was received for a subject LU that is verified, but, for a given session, either the partner LU is reported as the primary and the receiver does not consider that LU to be primary-capable, or the partner LU is reported as the secondary and the receiver does not consider that LU to be secondary-capable.

0033 Upon receipt of BFSESSINFO, the receiver considers the control block associated with a partner LU to be an other-domain resource that is not active or an application program that is not active.

0034 Upon receipt of BFSESSINFO, an SSCP is unable to associate the information received about a partner LU to be associated with an LU, an other-domain resource, or an application program.

0035 A network address was returned in RSP(RNAA) that the receiver believes is already associated with a different resource.

0036 BFSESSINFO received containing an invalid ALS address. For example, the ALS does not represent a T2.1 node.

0037 BFSESSINFO received for a subject LU, where the secondary address specified in the BFSESSINFO does not match the secondary address the SSCP believes is associated with the LU.

0038 The subject LU specified in the BFSESSINFO RU is not defined to the SSCP as an LU or an other-domain resource.

0039 A request that is valid only for a switched subarea link was received for a link that is not subarea-capable.

003A A request that is valid only for a 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.

0042 A BFINIT session request was received from a PLU that is not controlled by this SSCP.

0043 A request was received for a nonswitched resource that is valid only for a switched resource.

0044 A CONNOUT requested X.21 dial and auto-call capability was not present; resource mismatch.

0045 The DLU for a session request specified a network ID that did not match the network ID for the ALS providing services for the session.

0046 A CONNOUT was received indicating the sender and receiver have a system-definition mismatch: the CONNOUT Connection Type field specified a nonswitched link, but the receiver does not define the affected node as a T2.1 node on a nonswitched link or as one that supports XID3 exchange.

0047 The OLU for a session request or subject LU for a BFSESSINFO specified a network ID that did not match the network ID for the ALS providing services for the session.

0048 The DLU is an independent LU but the boundary function selected to provide its services is not capable of supporting independent LUs. The condition detected during session initiation processing after the ALS was selected for a switched resource.

0049 During processing of a BFSESSINFO for a subject LU, that LU was found to be inactive.

0050 The element address of an intra-FRSE PVC segment subport specified in a SETCV resides on the same frame-relay port as another subport within a subport set.

0051 The maximum frame size in the system-definition differs for any two partners in an intra-FRSE PVC segment subport set specified in a SETCV.

0052 Adjacent frame-relay equipment management protocols are not supported on either of the frame-relay ports for the primary or its backup subport specified in the SETCV for the intra-FRSE PVC segment subport set.

0053 A node identifies itself as an extended border node for some sessions but claims not to be an extended border node for other sessions.

0054 SETCV was received to define an intra-FRSE segment subport set between subports that are incompatible: one of the subports does not support alternate physical paths.

0055 SETCV was received to define an intra-FRSE segment subport set between subports that are incompatible: one of the subports *is* on an outboard DLC and the other is *not* on an outboard DLC.

0056 A CPSVRMGR session cannot be established over a LEN connection that is not of type TCP.

0057 An RNAA(Type=X′05′) requested the addition of an Internet Protocol (IP) PU to a frame-relay line that does not support IP traffic.

0058 A request was received for either an intra-FRSE segment subport set or a subport in a set, and the subport set contains incompatible subports: one of the subports is on an outboard DLC and another is not. A microcode-level upgrade is required to support this configuration.

0059 The same signaling information subfield has been received in a FID2 Encapsulation GDS variable from both the DLUR and DLUS, but the subfields have different values, and the DLUR cannot reconcile the differences.

005A The same signaling information subfield has been received in a FID2 Encapsulation GDS variable from both the DLUR and DLUS, but the subfields have different values, and the DLUS cannot reconcile the differences.

005B A CPSVRMGR session cannot be established over a path in which a branch uplink is an intermediate hop and cannot be established over a path including a branch downlink.

005D A received CONTACT request for the specified link station cannot be honored; the underlying microcode must be upgraded to the functional level of the using control program in order to support the requested function.

0878 Insufficient Storage: The storage resource required for a data format is not available.

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

0000 No specific code applies.

0001 CONNOUT contained more dial digits than can be stored by the receiving product.

0879 Storage Medium Exception: An exception has occurred involving a storage medium.

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

0000 No specific code applies.

0001 Disk I/O error.

0002 A nonrecoverable I/O exception has been encountered.

0003 Automatic DUMP/RE-IPL switches not saved to disk.

087A Format Processing Error: A processing error occurred during data formatting.

087B Resource Unknown: The request contains a session key that does not identify a session known to some gateway node; for example, a session activation request arrives at a gateway node after it has released the address transform for the intended session.

087C SSCP-PU Session Not Active: For example, a gateway SSCP-PU session that is needed to establish an address transform for a requested cross-network LU-LU session was not active.

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

0000 No specific code applies.

0001 An SSCP in the session setup path for an LU-LU session has lost connectivity with a gateway node traversed by the session, and has no other way to learn that the session has ended. An intermediate SSCP sends this sense data to one adjacent SSCP when it had previously lost connectivity with the other adjacent SSCP on the same session setup path. An endpoint SSCP sends this sense data to its adjacent SSCP when it had previously lost connectivity to a dependent LU or the boundary function of an independent LU.

0002 The SSCP lost connectivity with the boundary function of an independent PLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

0003 The SSCP lost connectivity with the boundary function of an independent SLU after the LU-LU session was started, and has no other way to learn that the session has ended; the SSCP

either never had a session to a gateway node in the LU-LU session path, or had previously lost connectivity to it.

087D Session Services Path Error: A session services request cannot be rerouted along a path of SSCP-SSCP sessions. This capability is required, for example, to set up a cross-network LU-LU session.

Bytes 2 and 3 contain sense code specific information that indicates the specific reason for not rerouting the request. Settings allowed are:

0000 No specific code applies.

0001 An SSCP has attempted unsuccessfully to reroute a session services request to its destination via one or more adjacent SSCPs; this value is sent by a gateway SSCP when it has exhausted trial-and-error rerouting.

Note: This code is used when SSCP rerouting fails completely. The remaining codes are used for failures to reroute to a particular SSCP. For example, they are associated with specific SSCPs when information about a rerouting failure is displayed in the node that was trying to reroute.

0002 An SSCP is unable to reroute a session services request because a necessary routing table is not available, that is, there is no adjacent SSCP table corresponding to the rerouting key in the Resource Identifier control vector. The receiver of this value will, if possible, try rerouting to another SSCP.

0003 This SSCP has no predefinition for an LU, but an adjacent SSCP does not support dynamic definition in partner SSCPs. As a result, this SSCP cannot both dynamically define the LU and reroute to that adjacent SSCP.

0004 A conflict in gateway definition or capabilities has been detected during cross-network session establishment.

0005 (Retired) An SSCP is unable to use the gateway node specified in CDINIT because that gateway node cannot allocate an address transform for the intended cross-network LU-LU session.

0006 (Retired) An SSCP is able to use only a subset of the alternate gateway nodes available to it. However, for the subset that it can use, none can provide the needed alias address pair.

0007 Sessions services path error: Two resources have been defined to represent the real and alias cross-domain resources.

0008 The adjacent SSCP does not support the requested CDINIT function (for example, notification of resource availability or XRF).

0009 A gateway SSCP is unable to reroute a CDINIT request: An address assigned by a gateway node duplicated an address assigned to a different LU name.

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.

000D An SSCP has purged a session services request because the adjacent SSCP did not respond to the request within a specified installation-defined time limit.

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 Session Services Path Error: A session services request cannot be rerouted into an APPN or subarea network.

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 A Locate/CD-Initiate reply, indicating Resubmit on Directed Search, was received after a directed search had been performed in response to a previous Locate/CD-Initiate reply.

0002 Duplicate or invalid search request received.

0003 A subarea search was not routed into an APPN network because a requested function was not supported by the APPN-subarea interchange node.

0004 An APPN search was not routed into a subarea network because a “search of subarea” was not permitted.

0005 A subarea search was not routed into an APPN network because the request originated in the APPN network containing the receiving node and the APPN network is capable of executing a broadcast search.

0006 A subarea search was not routed into an APPN network because a required component was not available.

0007 An interchange node received from an APPN network a Locate/CD-Initiate request that contains a control vector X′5D′ (with subfield X′81′) but cannot route to SSCPs in the network whose net ID is specified in that control vector because the control vector′s Disjoint Network indicator is not set to 1.

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 Insufficient Resource — Additional Control Block Information Included: The NCP receiver cannot act on the request because of a condition that prevents it from allocating the required control block.

Bytes 2 and 3 may contain the following sense code specific information:

ciii A condition “c” (see below) prevents the allocation of a control block from an NCP control block pool having the control block identifier (CBID) “iii”; the “iii” values are explained in *NCP/EP Reference Summary and Data Areas*, LY43-0030, under the “GPACBID” field. The values of “c” have the following meanings:

1 The NCP DYNPOOL threshold for dynamically allocated resource control blocks has been reached.

2 The NCP buffer utilization is too near the slowdown threshold.

3 No network element addresses are available.

4 The NCP generation limit for the associated control block has been reached.

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.

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 conflicts with a previous usage, or is unknown, or is known and does not have the required capabilities, or is a duplicate resource for the specified resource type. When a name conflict is detected, further name checking ceases; multiple name conflicts are not reported or detected.

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

0000 No specific code applies.

0001 The specified DLU real network name is known, but identifies a resource that is not LU-LU session capable.

0002 The specified DLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0003 The specified OLU real network name is known, but identifies a resource that is not LU-LU session capable.

0004 The specified OLU alias network name is known, but identifies a resource that is not LU-LU session capable.

0005 Name translation was invalid; that is, a different LU name was returned with the same network ID as the original LU name.

0006 The specified DLU real network name is known, but is a duplicate resource.

0007 The specified DLU alias network name is known, but is a duplicate resource.

0008 The specified OLU real network name is known, but is a duplicate resource.

0009 The specified OLU alias network name is known, but is a duplicate resource.

000B A cross-network DLU name is defined as a shadow resource, but shadow resources are not supported for cross-network sessions.

000C Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

000D When processing a session initiation RU, an SSCP has found two different resource definitions for the OLU, one with the real OLU name and one with the alias OLU name.

000E When processing a session initiation RU, an SSCP has found two different resource definitions for the DLU, one with the real DLU name and one with the alias DLU name.

000F The specified DLU network name is defined as a generic resource. The session should be reinitiated using the name of an LU.

0010 The LU6.2 partner returned a name in the User Data field of its RSP(BIND) that differs from the name it returned in the User Data field of its RSP(BIND) for a previous BIND. Either the partner changed its name or name changes in the network have caused delivery of the latest BIND to a different partner.

0011 The LU6.2 partner receiving a BIND carrying one specific target SLU name returned a name in the User Data field of its RSP(BIND) that is the same as it returned in response to a previous BIND carrying a different target SLU name. Name changes in the network name allowed two names to resolve to the same LU.

0012 The network qualifier of the name returned in the User Data field of a RSP(BIND) is not equal to the network identifier provided by the application that is using network qualified names.

Name changes in the network have caused alteration of the network identifier.

0013 A border node received a Topology Database Update from a node within its native subnet containing the CP name of a node that is adjacent to the border node across an intersubnet TG, i.e., in a nonnative subnet.

0015 A generic name of a resource has been received when only the real name of the resource can be specified.

0016 The DLUR-specified network name is known but is a duplicate resource.

0889 Transaction Program Error: The transaction program has detected an error.

This sense code is sent only in FMH-7.

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

0000 Program Error—No Data Truncation: The transaction program *sending* data detected an error but did not truncate a logical record.

Program Error—Purging: The transaction program *receiving* data detected an error. All remaining information, if any, that the receiving program had not yet received, and that the sending program had sent prior to being notified of the error, is discarded.

0001 Program Error—Data Truncation: The transaction program *sending* data detected an error and truncated the logical record it was sending.

0100 Service Transaction Program Error—No Data Truncation: The service transaction program *sending* data detected an error and did not truncate a logical record.

Service Transaction Program Error—Purging: The service transaction program *receiving* data detected an error. All remaining information, if any, that the receiving service transaction program had not yet received, and that the sending service transaction program had sent prior to being notified of the error, is discarded.

0101 Service Transaction Program Error—Data Truncation: The service transaction program *sending* data detected an error and truncated the logical record it was sending.

088A Resource Unavailable—NOTIFY Forthcoming: The SSCP cannot satisfy the request because a required resource is temporarily unavailable. When the required resource becomes available, NOTIFY NS(s) key X′07′ or X′08′ will be sent.

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

0000 No specific code applies.

0001 SSCP-SSCP Session Not Active: A SSCP-SSCP session required to reroute the cross-network request was not active.

0003 SSCP-LU session not active: The SSCP(DLU) is currently not in session with the DLU.

0004 LU session limit exceeded: The DLU is currently at its session limit and the requested session would cause the limit to be exceeded.

088B BB Not Accepted—BIS Reply Requested: Sent in response to a BB (either an LUSTAT bid or an Attach) to indicate that the receiver has sent a BIS request and wishes to terminate the session without processing any more conversations, but without sending an UNBIND. A BIS reply is requested so that the negative response sender may send a normal UNBIND. This sense code is sent only by LUs not supporting change-number-of-session (CNOS) protocols.

088C Missing Control Vector or Subfield: The RU or XID did not contain a required control vector or subfield.

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

nnyy Byte 2 contains the key (nn) of the subject control vector and byte 3 (yy) contains 0′s or the control vector′s type or missing subfield, if appropriate.

088D Duplicate Network Name: An SSCP has detected a violation of the requirement that network names used across multiple domains be unique within the multiple-domain network. For example, the SSCP(DLU) has detected that the OLU name received in CDINIT is currently also defined in the domain of the SSCP(DLU).

088E Capability Mismatch: A network component detected a capability mismatch between different resources involved in the same network function. For example, an SSCP detects that an LU has been assigned a subarea address too large for one of the other resources involved in the session initiation to support.

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

0000 A resource encountered during LU-LU session initiation is not ENA-capable; the session initiation request may be rerouted.

0001 A resource encountered during LU-LU session initiation is not ENA-capable; the session initiation request should not be rerouted.

0002 An SSCP has requested a “pre-ENA compatible” SLU address for an SLU that already has an ENA address.

0003 The gateway node selected by the gateway SSCP from the gateway node list is not ENA-capable when an ENA-capable gateway node is required. Another gateway node may be tried.

0004 During a dynamic path update, the SSCP detected that the update contained a path definition with an ER number greater than 7 and that the target node does not support extended subarea addresses. Therefore, the dynamic path update information for this destination subarea was not forwarded to the target node.

0005 The session could not be established because a specified extended subarea address exceeded that allowed at a node along the selected session setup path. The gateway SSCP doing gateway node selection may retry the session setup by selecting another gateway node having a larger subarea address limit in the network containing the DLU.

0006 The session could not be established because a specified extended subarea address exceeded that allowed at a node along the selected session setup path. The gateway SSCP doing gateway node selection may retry the session setup by selecting another gateway node that uses a smaller subarea address in the network containing the DLU.

0007 During a dynamic path update, the SSCP detected that the update contained a path definition with a subarea address above 255 and that the target node does not support extended subarea addresses. Therefore, the dynamic path update information for the destination subarea was not forwarded to the target node.

0008 The session could not be established because the dependent LU server detected an incompatibility between its capabilities and those of its dependent LU requester.

0009 The session could not be established because the dependent LU requester detected an incompatibility between its capabilities and those of its dependent LU server.

000A An attempt was made to establish a connection between a boundary node that does not support intersubnetwork link connections and a border node.

000B The Border Node indicator was set during XID exchange but both the Border Node and Intersubnet Extended Session Services support indicators were not set in CP-CP capabilities.

000C Intersubnetwork TG mismatch: Two nodes may have a system-definition mismatch, or two nodes may already have a non-intersubnetwork TG active and one attempts to activate an intersubnetwork TG between them.

000E VRTG is not supported across subnetwork boundaries.

000F An attempt was made to establish a CP-SVR pipe across a subnet boundary between a dependent LU server and a dependent LU requester with limited multisubnet support.

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 session).

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, but the OLU does not support XRF sessions. This is a definition mismatch between the OLU and the SSCP(OLU).

000F An invalid XRF-backup command was received.

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.

0011 Cryptography unavailable for XRF-backup session.

0012 An XRF-backup BIND associated with the existing XRF session supporting data compression was received that did not support compression.

0013 The existing session was negotiated using an extended BIND carrying the Length-Checked Compression (X′66′) control vector, but the XRF-backup BIND is nonextended.

0890 Search Failure.

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

0000 No specific code applies.

0010 Routing Error during a Directed Search: A Locate GDS variable for a directed search was received by an intermediate NNCP and could not be successfully routed to the destination control point.
Dudley 11/10/98

0011 The path used to transport the directed Locate request (i.e., a message containing Locate, Find Resource, and Cross-Domain Initiate GDS variables) does not support a sufficiently-large Locate message size to return the Locate response (i.e., a message containing Locate, Found Resource, and Cross-Domain Initiate GDS variables). The NNS(OLU) was requested to retry the directed search over a path supporting a sufficient Locate message size.

0020 Resource Not Found during a Directed Search: A Locate GDS variable for a directed search was received by the named destination CP and the search argument resource is not a local resource.

0021 Resource could not be located using only the nonverify function and a verify Locate was not permitted as part of the search.

0022 Destination of search not served by this CP.

0024 A search request or BIND was received from an unauthorized end node identifying an origin LU not represented in the network node server′s directory, and thus could not be authenticated.

0030 Resource Deleted, No Broadcast Required: A Locate GDS variable for a directed search was received by the named destination CP and the search argument resource has been deleted.

0036 Duplicate search to a subnetwork. This is an attempt to search a network previously reached by this search procedure. This condition indicates an attempt to loop back into a subnetwork through a different entry point.

0037 Unknown TG Vectors to Dependent LU Requester: A resubmitted Locate search for a dependent LU at its dependent LU requester was unsuccessful. This condition arises only after the dependent LU server has verified the existence of the dependent LU; retry.

0038 Too Many Directed Search Subprocedures: A Locate search exceeded the maximum height of the search tree; too many directed search subprocedures were tried; no retry.

0040 Resource Not Found during a Broadcast Search: A Locate GDS variable for a broadcast search was received by a CP that does not provide network services for the search argument resource and neither do any of the CPs searched in its broadcast subtree. This condition is detected by crossing search requests (a CP sends and receives a search request with the same FQPCID and the same search argument resource) or by

a local search failure and all CPs in the broadcast subtree returning this sense data.

0048 Neutral Reply Received from an End Node: A Locate reply with no Found and no Extended Sense Data (X′35′) control vector was received from an APPN end node.

0050 Quiesced CP: A CP in the broadcast search tree is in a quiescent state and, therefore, not receiving Locate GDS variables. This condition is detected when a CP in the search subtree is quiesced and no other CP in the subtree found the requested resource.

0060 Storage Not Available: A CP in the broadcast search tree does not have sufficient storage to participate in the search and no other CP in the search subtree found the requested resource.

0070 Session Outage: A CP in the search tree has lost its CP-CP session with a CP that had been sent a Locate GDS variable and no reply had been received.

0080 Duplicate Fully Qualified PCID: A CP in the search tree detected a duplicate fully qualified PCID for a different session request from the session request that first used the fully qualified PCID.

0081 PCID Modifier Too Long: A PCID Modifier List was received that had a length greater than 10 bytes.

0082 PCID Modifier Space Exhausted: A PCID Modifier List was received that contained the maximum of 10 bytes. As the maximum list size has been reached, another list entry cannot be made that was longer than 10 bytes.

0891 Invalid or Missing Network ID.

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

0000 No specific code applies.

0001 PLU Network ID Invalid: The network ID of the PLU is not the same as that of the SSCP(PLU).

0002 Invalid Network ID: The Network ID field in CONNOUT does not match the network ID defined for the target of the CONNOUT.

0003 Invalid network ID: The Network ID field in the RNAA is not the same as the native network ID because of a mismatch between the system definitions of the SSCP and the type 4 node.

0004 The Network Name control vector appended to the received XID3 does not contain a valid network ID. The network ID preceding the CP name must be greater than 0 and less than 8 bytes in length.

Alternatively, a network ID was received as an entry in a Register GDS variable without an accompanying resource name, resulting in an invalid resource name at the receiver; the entry was not registered.

0005 The Network Name control vector appended to the received XID3 does not contain a valid CP name. The CP name, following the network ID, must be greater than 0 and less than 8 bytes in length.

0006 Invalid Network ID: The sender has deactivated CP-CP sessions with the adjacent nonnative CP because neither CP contains border node support (i.e., neither sets byte 9, bit 7, to 1 in the CP Capabilities GDS variable that it sends).

0007 Invalid Network ID: Establishment of a switched link connection failed because the network ID of the destination PU was not equal to that of the requesting SSCP.

0008 Insufficient control blocks for dynamic network ID assignment. A CONNOUT specified a network ID that is not currently defined and sufficient control blocks are not available.

0009 The network ID specified in the VRID List (X′1B′) control vector is invalid.

000A Invalid Network ID: the network ID in the Network Name control vector does not match the network ID of the target resource of the REQACTPU.

0892 Automatic network shutdown (ANS) has occurred.

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

0000 No specific code applies.

0001 Session Reset After Loss of an SSCP: The SSCP controlling an LU has been lost. The session will be terminated because the T4 node, by system definition, terminates such sessions for this LU upon loss of the SSCP.

0002 The LU-LU session was in pending-active state when the SSCP failed. Although the T4 node, by system definition, continues an active LU-LU session upon the loss of the SSCP, the session was not completely set up, and thus it was reset.

0003 XRF-backup Session Reset. The XRF-backup session was reset because the T4 node resets the session upon loss of the SSCP.

0893 Takeover Not Complete

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

0000 No specific code applies.

0001 PLU Lacking a Control Point, Retry: The PLU is not currently receiving network services from a control point. The BIND is rejected because the session cannot be established. This sense data is returned by the boundary function of the PLU.

0002 SLU Lacking a Control Point, Retry: The SLU is not currently receiving network services from a control point. The BIND is rejected because the session cannot be established. This sense data is returned by the boundary function of the SLU.

0003 Sequence Error: The SSCP should not send an RNAA for an independent LU until the takeover sequence is complete for the link station, that is, until all BFSESSINFOs for that LU have been received and accepted.

0894 Migration Support Error: The sender of the request is relying on migration support that is not available.

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

0000 No specific code applies.

0001 BIND cannot be extended: A BIND that is not an LU6.2 BIND was received and cannot be extended by the receiver.

0895 Control Vector Error: The RU or XID contained a control vector that was in error.

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

0000 No specific code applies.

xxyy Byte 2 (xx) contains the key of the control vector first detected in error. If more than one control vector is in error, only the first erroneous one is reported. Byte 3 (yy) of the sense code specific data contains the (0-origin) byte offset of the error within the control vector.

0896 Control vector too long.

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

0000 No specific code applies.

0001 Network Name (X′0E′) control vector is too long; the vector data portion is greater than 18 bytes long.

0897 System Definition Mismatch: The requested function is not supported by the receiver, or a mismatch exists between the system definitions of the sender and receiver.

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 could also 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 field 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.

000D Resource type not defined in receiver.

000E Reserved

000F Reserved

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

0012 The receiving SSCP has a system-defined name for the SSCP(OLU) that differs from the SSCP(OLU) name in the session initiation request.

0013 A CDINIT was received that indicated that the receiving SSCP controls the OLU.

0014 The receiving T4 node (though capable of supporting the function) was not defined by local system-definition option to support the requested dump type.

0015 The OLU is represented using a dynamically defined resource but the ALS selected to provide its services does not permit dynamic definitions. The condition is detected when a session initiation request is received for an independent LU and no predefinition is found for the OLU resource. The session initiation is rejected.

0016 The DLU is represented using a dynamically defined resource but the ALS selected to provide its services does not permit dynamic definitions. The condition is detected when a session initiation request is being processed for an independent destination LU and no predefinition is found for the DLU resource. The session initiation request is rejected.

0017 The request was received for an independent LU over a specific ALS but that ALS is not defined to provide services for the subject LU. The condition is detected when a session initiation request is recieved and the ALS for which the request was received was not predefined to provide service for that independent LU. The session initiation request is rejected.

0018 Session Initiation Status Not Supported: A session initiation request was received that contained a session initiation status field invalid for the receiving node.

0019 The SSCP has received a CONTACTED or REQCONT containing an XID3 carrying an unrecognized CP name; the SSCP supports only predefined CP names.

001A The source or destination service access point address (SSAP or DSAP) in the logical link control protocol data unit of the XID information field for a token-ring LAN is unknown.

001B An XID was received over a branch uplink but the XID sender is already connected over a branch downlink; or an XID was received over a branch downlink but the XID sender is already connected over a branch uplink.

001C An XID was received from a peripheral or extended border node over a link which the local node defines as a branch downlink.

001D An XID was received from a Dependent Logical Unit Requester (DLUR) over a link which the local node defines as a branch downlink; or an XID was received by a DLUR over a link which the XID sender defines as a branch downlink.

0898 Session Reset: The XRF session is being reset.

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

0000 No specific code applies.

0001 The XRF-active session has been reset because the XRF-backup PLU forced a takeover.

0002 XRF-backup Hierarchical Reset: The identified XRF-backup LU-LU session is being deactivated because the related XRF-active session terminated normally. The LU sending this sense data is resetting its half-session before receiving the response from the partner LU. (See UNBIND type X′12′.)

0003 XRF-active Hierarchical Reset: The identified XRF-active LU-LU session is being deactivated because the related XRF-backup session performed a forced takeover of this session (via

SWITCH). The LU sending this sense data is resetting its half-session before receiving the response from the partner LU. (See UNBIND type X′13′.)

089A Invalid File or File Not Found: The requested file was not found, or was found to be an invalid file.

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

0000 No specific code applies.

0001 Requested file not found.

0002 The specified load module already exists and, therefore, cannot be added.

0003 An IPL time has not been set for the specified load module.

0004 Another load module on the MOSS disk has the same IPL time as the one specified for the load module in the Modify load command.

0005 Unable to locate required associated object.

089B Session Correlation Exception: The session correlation procedure detected an exceptional condition at the SLU.

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

0000 No specific code applies.

0001 RUs Out of Order: A BIND request with the correlating Fully Qualified PCID (X′5F′) control vector arrived before UNBIND(Type X′02′) was received for the correlated session. This sense data is sent in an UNBIND that terminates the correlated session.

0002 Correlator Not Found: A BIND request with the correlating Fully Qualified PCID (X′5F′) control vector cannot be correlated with any previous session.

089C Duplicate Session-Related Identifier.

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

0000 No specific code applies.

0001 Invalid URC: The URC received in a BFINIT duplicates a URC for an outstanding session initiation attempt from the same BF.

089D Gateway Node Error Detected during Cross-Network Session Initiation.

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

0000 No specific code applies.

0001 The gateway node list used to select a gateway node to cross a network boundary is exhausted.

0003 RNAA has failed; another gateway node should be tried.

0004 Address conversion based on the subarea/element address split was unsuccessful.

0005 The gateway node selected by one gateway SSCP is not known to another gateway SSCP in the same gateway. This can be a system definition error in the gateway SSCP that does not recognize the gateway node.

0006 A gateway SSCP has found that a gateway node has assigned duplicate addresses.

089E Identified Data Object Already Exists.

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

0000 No specific code applies.

0001 A request to create a new data object has failed because the identified data-object already exists at the target node.

0002 A request to replace a data object has failed because it specifies a to-be-deleted data object different from the to-be-stored data object; however, the to-be-stored data object already exists.

089F Component Not Available: The node component required to satisfy a request is not currently available.

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

0000 No specific code applies.

0001 The control point was not available for assignment of the FQPCID.

0002 Reserved

0003 The control point was not available to initialize the next TG of an LU-LU session in an intermediate network node.

0004 A session initiation has failed because a generic-resource coupling facility is unavailable to do a necessary information update.

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 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 received DACTPU, DACTLU, or DISCONTACT. The LU will send an UNBIND with a reason code of X′0A′ (SSCP Gone).

0002 The LU or session connection manager (SCM) will send UNBIND with a reason code of X′0F′ (Cleanup).

0003 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 has occurred. The gateway node will send UNBIND with a reason code of X′11′ (Gateway Node Cleanup).

0004 Reversed FRSN Values: The value in the Last FRSN Sent field is greater than the value in the Current FRSN field in a received TDU GDS variable (no retry). The CP will send an UNBIND with a reason code of X′0F′ (Cleanup)

0005 TDU Sent Out of Order: The value in the Last FRSN Sent field of the current TDU GDS variable is less than the value of the Current FRSN field in the TDU GDS variable that immediately preceded it, or is greater than it and the receiver cannot store the out-of-sequence value (no retry). The CP will send an UNBIND with a reason code of X′0F′ (Cleanup)

0006 Invalid FRSN Value: In a CP Capabilities GDS variable, the adjacent node indicated receipt of a TDU with a FRSN value greater than the last one sent.

0007 DLUS-DLUR Session Deactivation (Disruptive): LU-LU sessions for DLUR-supported dependent LUs should be reset.

0008 DLUS-DLUR Session Deactivation (Nondisruptive): LU-LU sessions for DLUR-supported dependent LUs should not be reset.

0009 DLUS-DLUR Session Deactivation (Nondisruptive): Protocol violation detected; LU-LU sessions for DLUR-supported dependent LUs should not be reset.

000A DLUS-DLUR Session Deactivation (Nondisruptive): DLUR should wait for DLUS reactivation of the DLUS-DLUR session; LU-LU sessions for DLUR-supported dependent LUs should not be reset.

000B DLUS-DLUR Session Deactivation (Nondisruptive): DLUR should activate a DLUS-DLUR session with a different DLUS if possible; otherwise, it should wait for DLUS reactivation of the DLUS-DLUR session; LU-LU sessions for DLUR-supported dependent LUs should not be reset.

000C DLUS-DLUR Session Deactivation (Disruptive): The DLUS received a CP-SVR pipe activation request without a REQACTPU; DLUR should activate a DLUS-DLUR session with a different DLUS if possible.

000D DLUS-DLUR Session Deactivation (Persistent): The DLUR is initiating deactivation of a persistent DLUS-DLUR session.

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.

08A3 Invalid Data in Control Vector.

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

0000 No specific code applies.

0001 Invalid data returned in Call Security Verification control vector.

0002 Route error: the first TG Descriptor (X′46′) control vector contained in this session Route Selection (X′2B′) control vector contains a Composite Route Selection (X′85′) subfield specifying one or more additional TGs beyond the local node, which is the CP(OLU).

0004 Route error: the TG Descriptor (X′46′) control vector for the last or only TG of the session Route Selection (X′2B′) control vector contains a Composite Route Selection (X′85′) subfield specifying one or more additional TGs beyond the local node, which is the CP(DLU).

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:

0000 No additional code applies.

0001 Server object not found.

08A8 Multiple-Domain Support Routing Exception: The MDS router in the reporting NAU is unable to perform the required routing for an MDS-MU.

When this SNA report code is used in an SNA condition report (X′1532′) GDS variable, the destination NAU name is included in the Reported on Location Name (X′09′) subvector and the destination MS application name is included in the Reported On Agent (X′04′) subvector of the condition report.

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

0000 No specific code applies.

0001 Destination NAU name unknown. Directory services could not locate the requested destination name.

0002 Directory services unavailable. No routing possible.

0003 MS application program name not recognized.

0004 Use of CPSVCMG session not permitted. The reporting network node has received an MDS-MU over a CPSVCMG session from another network node. These sessions are used for MDS-MUs only between a network node and its served end nodes.

0005 Function not supported by EN destination. The back-level end node destination does not support receipt of MS messages (reported by serving network node).

0006 Function not supported by destination. The back-level destination does not support receipt of MS messages other than MS Capabilities and Alert.

0007 Function not supported by serving NN. The serving network node of the end node destination does not support routing of MS messages (reported by network node performing routing).

0008 Function not supported by EN. The reporting end node has received an MDS-MU with a destination other than itself.

0009 Destination not supported by reporting NN. A network node has received an MDS-MU from another network node that cannot be routed. The destination is not the reporting network node itself nor is it one of the served end nodes.

If the MDS-MU was routed based on nonverified directory information (as indicated by the Routing verification indicator in the MDS Routing Information), the MDS-MU will be returned to the routing network node along with the SNA condition report.

000A Unrecoverable session failure. The MDS_SEND TP in the reporting node was unable to send the message because of an allocation error. Retries have been exhausted.

000B Unrecoverable TP failure in remote node. The MDS_SEND TP in the reporting node was unable to send the message because of a TP failure in a remote node. Retries have been exhausted.

000C MS Application program failure. The MDS router in the destination NAU is unable to communicate with the destination MS application program.

000D Unrecoverable TP failure in reporting node. The MDS router in the reporting node was unable to send the message because of a local TP failure.

000E Correlation error. An MDS-MU has been received that is not the first for a unit of work (First MDS Message indicator in the MDS Routing Information Message is 0), but the agent unit of work correlator is unknown (does not match any active MDS transaction). Also used to report the receipt of a duplicate correlator (MDS-MU with first MDS message indicator is 1, but the agent unit of work correlator matches one currently in use).

000F MS application program congestion. The MDS router in the destination NAU is unable to communicate with the destination MS application program because of local congestion (implementation buffer space for queuing additional MDS-MUs has been exhausted).

0010 MDS HPO not supported by remote node. The destination NAU does not support the MDS high performance option.

0011 MDS HPO not supported by MS application program. The destination MS application program does not support the use of the MDS high performance option.

0012 Unrecoverable failure of user-mode session. MDS has detected an error on a user-mode session (a user-mode session in this context is one with a mode name other than SNASVCMG or CPSVCMG). Retries have been exhausted. Application program data may have been lost.

0013 Session UNBIND notification. The last session to the indicated destination has been deactivated. Refer to product documentation for additional information.

08A9 Multiple-Domain Support Transaction Failure: The reporting MDS router or MS application program has detected a condition that has impacted an outstanding unit of work (identified by the agent unit of work correlator of the MDS error message).

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

0000 No specific code applies.

0001 Failure caused by outage of a CPSVCMG session.

0002 Failure caused by outage of an SNASVCMG session. All retries have been exhausted.

0003 Unit of work canceled by reporting MS application program. The unit of work has been canceled because of a timeout in the reporting MS application program.

0004 Unit of work canceled by reporting MDS Router. The unit of work has been canceled by a garbage-collection timeout in the reporting MDS router.

0005 MDS router internal failure. The unit of work has been canceled because of an internal failure in the reporting MDS router.

0006 MS Application internal error. The unit of work has been canceled either because the reporting MS application program was terminated or because another application program served by it was terminated. The type of program termination (normal or abnormal) is not indicated.

0007 MS Application router re-initialization. The unit of work has been canceled by the reporting MDS router because of a re-initialization of the application-level router.

0008 The CMIP agent client has received an MDS Error Message for its transaction with a CMIP manager.

0009 The CMIP agent client has received an MDS Error Message for its transaction with the CMIP agent server.

08AA Required GDS Variable Missing: The MS Multiple-Domain Support Message Unit (MDS-MU) is missing a required GDS variable.

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

nnnn Bytes 2 and 3 following the sense code contain the ID of the missing GDS variable.

08AC Buffer Processing Failure: During the processing of a buffer, a failure has occurred.

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

0000 No specific code applies.

0001 A buffer-related failure occurred during the processing of an UNBIND request or response.

0002 A buffer-related failure occurred during the processing of a BIND request or response.

08AE Length Corruption: A component failure occurred within a node processing a request or response.

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

0000 No specific code applies.

0001 The length of a control vector has been corrupted during processing.

08B1 SNA/FS Request Failure: The SNA/FS server was unable act on the request specified in the SNA/FS unit of work.

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

0000 No specific code applies.

0001 Reserved.

0002 Reserved.

0003 Reserved.

0004 The Data Object Class is not valid for the Intention specified. When this report code is used in a SNA Condition Report, it is accompanied by two supplemental_reports, the first of which contains the Data Object Class requested and the second of which contains the Intention requested.

0005 Request rejected because back-out cannot be supported.

0006 Unable to perform a fetch. Object may be in use. When this report code is used in an SNA Condition Report, it is accompanied by a supplemental report identifying the to_be_fetched_name at the source node.

0007 Object in use. Unable to perform a delete or replace. When this report code is used in an SNA Condition Report, it is accompanied by a supplemental report identifying the to_be_deleted_name at the target node.

0008 The specified Data Object package cannot be restored to it′s correct set of system-specific objects, because the package is either in an incorrect format or has been corrupted.

0009 Catalog Damaged. Unable to restore the Object.

000A Stored algorithm unknown. The server attempted to decompress a stored object which had been compressed using an algorithm unsupported by the server. When this report code is used in an SNA Condition Report, it is accompanied by a supplemental report identifying the stored object′s compression algorithm value present in the source′s catalog.

08B2 Data Transmission Failure: The data transmission between an application program in an SNA MS entry point and an application program in a subentry point was incomplete, causing abnormal termination of the function.

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

0000 No specific code applies.

0001 A timeout has occurred while waiting for transmission of data between the two application programs. For example, a service processor has timed out while waiting to receive data from the main processor.

0002 A timeout has occurred while waiting for transmission of data between two applications.

08B3 DS DTMU Build Exception: Building of the DS Distribution Transport MU was unsuccessful.

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

0000 No specific code applies.

0001 A maximum-sized agent object in the MU being built is insufficient to contain all the data required, and segmented agent objects are not supported. The MU is built to include as much data in the agent object as possible.

08B5 Network Node Server Not Required: Sent by an APPN end node control point to a network node control point (1) to deactivate CP-CP sessions with the NNCP, or (2) to reject a CP-CP session BIND from the NNCP. This sense data may also be sent to the node operator with notification that CP-CP sessions with the NNCP have been deactivated. The end node no longer requires network node services from the receiver; e.g., the end node has found a more preferred network node server.

Note: This sense data value is carried within the X′35′ control vector on an UNBIND(Type = X′01′) for case (1) above, or on an UNBIND(Type = X′FE′) for case (2).

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

0000 No specific code applies.

08B6 CP-CP Sessions Not Supported: Used to inform the receiver that an attempt to activate CP-CP sessions has failed because the given TG does not, in fact, support CP-CP sessions.

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

0000 support for CP-CP sessions on this TG has been removed since the time when the TG was first activated.

0001 During link activation on a switched link, it was discovered that the partner node does not support CP-CP sessions on this TG.

Request Error (Category Code = X′10′)

This category indicates that the request was delivered to the intended NAU component, but could not be interpreted or processed. This condition represents a mismatch of NAU capabilities.

Category and modifier (in hexadecimal):

1001 RU Data Error: Data in the request RU is not acceptable to the receiving component; or a +RSP(BIND) received at an intermediate node within a session path contained such data. 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.

0010 A User Data control vector 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.

0024 A BIND or +RSP(BIND) that was not for LU type 6.2 and not in extended format was received at an intermediate APPN network node. The session is terminated.

0026 The length of a GDS variable within the request RU is invalid.

0027 A GDS variable within a Locate is invalid.

0029 The IP address specified in an RNAA(Type=X′05′) for a new Internet Protocol (IP) PU is not a valid internet host address.

0032 The Maximum Transfer Unit (MTU) value specified in an RNAA(Type=X′05′) for a new Internet Protocol (IP) PU is not in the valid range 283 – MAXFRAME.

0033 The name of the deciphering CP in the X′82′ subfield of a Cryptography (X′63′) control vector does not match the name of the CP(PLU) receiving the control vector.

0034 A topology database update was received across an intersubnetwork link carrying topology information about an adjacent subnet.

0036 The Message Authentication Code received in the RU did not match the one generated by the receiver for that RU.

0037 The subnetwork mask specified for the DR ADD of a frame-relay IP PU is not a valid subnetwork mask for the network involved.

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; or a +RSP(BIND) received at an intermediate node within a session path 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. For example, the LU 6.2 half-session may have received a SIGNAL RU when its local conversation style is full-duplex. (However, the half-session rejects the SIGNAL only if it is for the current bracket. Early SIGNALs are held for the correct bracket by saving the SIGNAL value until the correct BB arrives.)

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.

0008 Broadcast Search with Reservation: An NNCP received a broadcast search request with reservation.

0009 Initiate Type: The initiate type requested in the CD-Initiate GDS variable or Initiate-Other-CD GDS variable is not supported at the receiver.

000B A BIND specifying delayed request mode was received from a non-6.2 type LU, but delayed request mode is not supported in the receiver.

000C A standalone BIND was received from a node that is served by an SSCP that does not support standalone BINDs.

000D The function identified in the request is not supported by the processing application transaction program.

0010 The RU is not known to session services.

0011 A session key is not supported.

0012 A control vector is not supported.

0014 Cryptography is not supported but a nonzero length was specified for the cryptography key.

0015 Queuing not supported for a controller session.

0016 Service parameter not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report identifying the service parameter triplet (or triplets) that was not supported.

0017 Service parameter level not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report identifying the service parameter triplet (or triplets) that was not supported.

0018 Destination-role function not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying the structure and containing the contents that specified the unsupported function. Whenever the structure report is not sufficient to identify the unsupported functions, the supplemental report may also be present.

0019 All-role function not supported. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying the structure and containing the contents that specified the unsupported function. Whenever the structure report is not sufficient to identify the unsupported functions, the supplemental report may also be present.

001A Reserved.

001B Unable to initiate Agent.

001C Function conflicts with Format Set 1 encodings. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying the structure and containing the contents that specified the conflicting function.

001D Reserved

001E Reserved

001F Multiple-destination traffic not supported. The reporting location is a specialized, end-only role implementation that supports single-destination traffic only.

0020 A session initiation request specified an OLU and DLU that are the same LU. A dependent LU cannot establish a session with itself.

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

0023 A session initiation request requiring extended session services support from its network node server was received at an APPN end node that does not have this service available to it.

0025 The component received a NOTIFY RU whose type is not supported.

0027 LU type not supported.

0028 Nonnegotiable BIND not supported by the receiver.

0029 Transmission service (TS) profile of BIND not supported by the receiver.

0030 Normal-flow send/receive mode conflicts with the mode specified in the transmission services (TS) profile of a received BIND.

0031 The primary LU cannot support being first speaker; secondary LU must be first speaker.

0032 In BIND, the specified bracket termination is not supported by the receiver.

0033 Definite response mode is not supported by the receiver.

0034 Secondary LU cannot send EB when normal-flow send/receive mode is full-duplex.

0035 Bracket error resulting from failure of sender to enforce bracket rules for the session.

0036 A network node server received a Notify GDS variable whose type is not supported by the client DLU.

0037 Direction Error: An RU was received inappropriate to the NAU type.

0038 Surrogate autologon support is not provided for this session.

0039 A third-party-initiated session request specified an initiating LU (ILU) and a destination LU (DLU) that are the same LU. An application program cannot initiate a third-party-initiated session to itself.

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 data value can also be used instead of X′0826′.

1004 Request error: function not supported.

1005 Parameter Error: A parameter modifying a control function is invalid, or outside the range allowed by the receiver.

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

0000 No specific code applies.

0001 For NMVT, the address type field in an SNA Address List subvector does not match the address type required by the command subvector.

0002 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0004 Invalid display type was requested.

0005 Invalid storage length for display type requested.

0006 Invalid storage address; out of specified range.

0007 The command in a Request Change Control MS major vector is incompatible with the SNA/FS server instruction.

0008 Set aside for implementation-specific use, and will not be otherwise defined in SNA; see implementation documentation for details of usage.

0010 Incorrect setting of backup focal point flag. The MS Capabilities (X′80F0′) major vector received from the focal point contains a backup focal point flag with a value of 1 (indicating that the entry point is to keep its current focal point), but the Focal Point Identification (X′21′) subvector in the same major vector names a new backup focal point.

0121–0229, 0260 Set aside for implementation-specific use, and wil not be otherwise defined in SNA; see implementation documentation for details of usage.

1006 Required field or parameter is missing.

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

0000 No specific code applies.

0001 One or more required COS names were omitted.

0002 A required name was omitted.

0003 A required network identifier was omitted.

0004 A required session key was omitted.

0005 A required control vector was omitted.

0006 A required subfield of a control vector was omitted.

0007 The TG number field was omitted.

0008 The system-defined ID number, used within the Node Identification field of an XID, was omitted.

0009 A required GDS variable 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 data value is sent in FMH-7 or UNBIND.

Bytes 2 and 3 following the sense code contain sense code specific information. Figure 10-2 on page 10-99 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	Unrecognized FM Header Command Code: The partner LU received an FM header command code that it does not recognize. For LU 6.2 this sense data is sent only in FMH-7.
600C	A null sequence field is required.
600D	User to user program transition is not allowed.
600E	User to non-SNA defined program transition is not allowed.
600F	The FMH-5 reset attached program (RAP) was not sent properly.
6010	The FMH-5 reset attached program (RAP) was sent with an inactive Attach register.
6011	Invalid Logical Unit of Work (LUW): The LUW Length field (in a Compare States GDS variable or an FMH-5) is incorrect, or the length field is invalid, or a LUW ID is not present but is required by the setting of the synchronization level field.
6021	Transaction Program Name Not Recognized: The FMH-5 Attach command specifies a transaction program name that the receiver does not recognize. This sense data is sent only in FMH-7.
6031	PIP Not Allowed: The FMH-5 Attach command specifies program initialization parameter (PIP) data is present, but the receiver does not support PIP data for the specified transaction program. This sense data is sent only in FMH-7.
6032	PIP Not Specified Correctly: The FMH-5 Attach command specifies a transaction program name that requires program initialization parameter (PIP) data, and either the FMH-5 specifies PIP data is not present or the number of PIP subfields present does not agree with the number required for the program. This sense data is sent only in FMH-7.
6034	Conversation Type Mismatch: The FMH-5 Attach command specifies a conversation type that the receiver does not support for the specified transaction program. This sense data is sent only in FMH-7.

6040 Invalid Attach Parameter: A parameter in the FMH-5 Attach command conflicts with the statement of LU capability previously provided in the BIND negotiation.

6041 Synchronization Level Not Supported: The FMH-5 Attach command specifies a synchronization level that the receiver does not support for the specified transaction program. This sense data is sent only in FMH-7.

6042 Reconnection Not Supported: The FMH-5 Attach command specifies reconnection support but the receiver does not support reconnection for the specified transaction program. This sense data is sent only in FMH-7.

6043 Unable to Reconnect Transaction Program—No Retry: The FMH-5 Reconnect command specifies the conversation correlator of a transaction program to which the receiver cannot reconnect. The condition is not temporary. This sense data is sent only in FMH-7.

6044 Unable to Reconnect Transaction Program—Retry Allowed: The FMH-5 Reconnect command specifies the conversation correlator of a transaction program to which the receiver cannot reconnect. The condition is temporary. This sense data is sent only in FMH-7.

6045 Reserved

6046 An SNA/DS transaction program is unable to allocate a conversation with a SNA/DS partner.

6047 An SNA/DS transaction program in conversation with an adjacent SNA/DS transaction program has detected from LU 6.2 PS a return code of RESOURCE_FAILURE.

6048 An SNA/DS transaction program in conversation with an adjacent SNA/DS transaction program has detected from LU 6.2 PS a return code of DEALLOCATE Type(Abend).

6050 Missing or unexpected X′12F6′ GDS.

6051 The length of the Attach Sequence Number field in the FMH-5 is not 0 or 8.

6052 The Attach sequence number is outside the expected range.

6053 Incorrect length of X′12F6′, X′12F7′, or X′12F8′ GDS variable.

6054 Byte 4, bit 5 of the Attach is 1, indicating the use of GSS-API based authentication, but either other security bits in the Attach are on also or the Attach access security information fields are present.

6055 Byte 4 bit 3 of the Attach FMH-5 is 1, indicating the presence of a password substitute, but an Attach sequence number is not in the FMH-5.

6056 The partner LU responded to an attach with any data other than an FMH-7 or an Authentication Token Data (X′12F6′) GDS variable.

6057 Deferred authentication processing was requested but the Attach FMH-5 did not contain a valid conversation correlator.

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.

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
6041–6044				X
6046				X
6047				X
6048				X
6051				X
6052				X
6055				X
C000–C003			X	

Figure 10-2. Usage of X′1008′ Sense Code Specific Information by LU Type

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 struc-

ture report that identifies and contains the header of the excessively long structure, plus a supplemental report that contains the allowed maximum length.

0007 Length exception. Length arithmetic is out of balance. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies and contains the header of the structure that exceeded its parent′s boundary.

0008 Required combination of structures absent. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures that make up the combination, indicating for each whether it was present or absent.

0009 Precluded combination of structures present. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures that make up the precluded combination.

000A Required combination of structures and data values absent. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures and data values that are present, plus structure reports that identify the absent structures needed to complete the combination.

000B Precluded combination of structures and data values present. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures and data values that make up the precluded combination.

000C Unknown or unsupported data value. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure and contains the unknown or unsupported data value.

000D Incompatible data values. When this SNA report code is used in an SNA condition report, it is accompanied by structure reports that identify the structures and the incompatible data values.

000E Precluded character present. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure, indicates the byte offset of the offending byte, and includes the byte containing the precluded code point.

000F Data-value out of range. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure and contains the offending data value, plus a supplemental report that contains the maximum value allowed within the range (if a maximum range value is applicable).

0010 Segmentation present where precluded. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure that should not have been segmented.

0011 Precluded data value. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure and contains the offending data value.

0012 Recognized but unsupported structure. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the structure.

0013 None of several possible structures found. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the parent of the absent structure and may contain an unrecognized structure that was found in the place of the absent structure. The structure report also contains a sibling list of the possible structures.

0014 Incorrect order of child structures found. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report that identifies the parent of the incorrectly ordered child structures.

100C Unrecognized Message Unit

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

0001 The received byte stream could not be identified by the receiving SNA component. When this SNA report code is used in an SNA condition report, it is accompanied by a structure report identifying and containing the unrecognized message unit, plus a sibling list of the allowed message units.

100D Request Inconsistency: The control information provided for the request is not consistent with other information in the request.

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

0000 No specific code applies.

0001 Server object size is incompatible with service level. When this SNA report code is used in an SNA condition report, it is accompanied by one structure report containing the capacity service parameter triplet and one supplemental report containing the server object size.

0002 A reply DTMU was received before completing a three-way responsibility flow in an SNA/DS request. Retry is allowed.

100E Directing Exception: A node is unable to perform the required directing or redirecting function for a request as a result of insufficient directory support, or incompatibility between TP name and presence/absence of a user name.

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

0000 No specific code applies.

0001 Agent name known but not supported for specified user destination.

0002 Agent name known but not supported for specified node destination.

0003 Agent name is known at this DSU but not available.

100F Improper SNA/DS Usage of LU 6.2.

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

0001 An SNA/DS transaction program in conversation with an adjacent SNA/DS transaction program has detected an improper sequence of LU 6.2 basic conversation verbs.

1010 Error on Locate Search or CP Capabilities Message Detected.

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

0000 Unrecoverable error, such as a duplicate control vector, was detected.

0001 A broadcast search resulted in two or more conflicting positive replies that differ on the CP owning the target resource. Multiple positive replies are acceptable, as long as all indicate the same owning CP.

0002 Retired — see 1010 1002 for its replacement.

0003 An error was detected that prevented the exchange of CP capabilities. Recovery may be attempted.

0004 Unrecoverable error on CP Capabilities GDS variable exchange prevented its initiation or completion on a contention-winner CP-CP session.

0005 The intersubnetwork Locate failed because an entry for the destination network ID does not exist in the border node′s subnetwork list.

0006 The CP Capabilities GDS variable received across an intersubnetwork link from an adjacent network node did not indicate support for an intersubnetwork connection to a border node.

0007 The internetwork Locate was failed by an IBN because the initition type of the Locate was not supported by the IBN.

1000 Length error in CP Capabilities GDS variable.

1002 Invalid GDS variable received when CP Capabilities GDS variable was expected.

4004 Incomplete negative or neutral reply received on a search, or reservation indicated on Broadcast, or “All” specified on a directed search.

5000 Length error in CD-Initiate GDS variable.

5002 No CD-Initiate GDS variable returned on a search request.

5006 Session polarity or initiate type value received in CD-Initiate GDS variable not supported.

500A Mode name length error in CD-Initiate GDS variable.

A002 Find GDS variable not present on Locate search request.

B080 Command Parameters (X′80′) control vector not present on Found GDS variable.

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

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

1014 Control Vector Error on a Directory Services or Session Services GDS Variable.

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

003C Missing Associated Resource Entry (X′3C′) control vector on Find or Found.

003D Missing Directory Entry (X′3D′) control vector on Find or Found.

0060 Missing Fully Qualified PCID (X′60′) control vector on Locate.

0080 Invalid control vector.

023C Conflicting directory entry or invalid Associated Resource Entry (X′3C′) control vector.

502B No RSCV received from a network node server.

502C No COS/TPF control vector received in a CD-Initiate reply from a network node server.

502D The COS/TPF control vector received on BIND is different from that on the corresponding Locate.

5046 TG vectors are not present on a CD-Initiate from an end-node OLU or DLU.

A080 Missing Command Parameters (X′80′) control vector on Find.

A082 Missing Search Argument Directory Entry (X′82′) control vector on Find.

B280 A Found from an end node indicated the directory entry for a located resource was a wild-card entry.

1015 XID Length Error: The XID3 was too long or too short.

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

0000 No specific code applies.

0001 The received XID3 has fewer than 29 bytes.

0002 There is a mismatch between the number of bytes specified in the Length field of XID3 and the actual length of the received XID3.

1016 XID Format 3 Parameter Error: Data in the XID3 is not acceptable to the receiving component because the value in the received XID3 field, whose byte and bit offset is specified by the XID Negotiation Error (X′22′) control vector (which also carries this sense data), is inconsistent with the corresponding field in the sent XID3.

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

0000 No specific code applies.

0001 The field in the received XID3 that specifies the maximum number of I-frames that the sender can receive before acknowledgment is set to 0.

0002 The adjacent node has been inconsistent in its request for ACTPU. In a nonactivation XID3 exchange, it has changed the value of the ACTPU Suppression indicator sent in the previous XID3 exchange.

0003 The field in the received XID3 that specifies the maximum BTU length that the sender can receive is set to less than 99 bytes, the minimum required.

0004 The received XID was not XID format 3 when XID format 3 was expected.

0005 The adjacent node does not support BIND segment generation but does support receipt of BIND segments. Any T2.1 node supporting receipt of BIND segments must also support generation of BIND segments.

0006 The adjacent node is an end node, does not support BIND segment receipt, and has a maximum BTU size of less than 265, the minimum required in this case.

0007 The adjacent node is a network node, does not support BIND segment receipt, and has a maximum BTU size of less than 521, the minimum size required in this case.

0008 The adjacent node has changed its networking capabilities in an XID3 from those declared in the previous negotiation-proceeding or nonactivation XID3. A node may not change from an end node to a network node or *vice versa* in two different negotiation-proceeding or nonactivation XID3s.

0009 The adjacent node is an APPN network node, does not provide CP services, and supports CP-CP sessions, a combination not allowed.

000A During a nonactivation XID3 exchange, the adjacent node has changed the TG number that was negotiated during the activation exchange.

000B The adjacent node is the TG number negotiation winner and designates a TG number that the receiving node cannot allocate to this connection. When parallel TGs are supported between the two nodes, 0 is always such a number.

000C The adjacent node is an APPN network node that does not support BIND segment generation, and this node has a maximum BTU receive size of less than 521. This node may, therefore, be unable to receive a BIND with RSCV from the adjacent network node.

000D The adjacent node indicates that it does not support the SDLC command/response profile in its XID3. This is the only command/response profile supported by APPN and LEN nodes.

000E Different product set IDs have been given in the Product Set ID (X′10′) control vectors appended to two different received XID3s from the same adjacent node.

000F The link station roles specified in the sent and received negotiation-proceeding XID3s are not compatible. To activate a connection, one node must contain a primary link station; the other, a secondary link station.

0010 The support of combined asynchronous balanced mode link stations indicated in the sent and received negotiation-proceeding XID3s is not in agreement.

0011 A received XID3 indicates an attempt to activate multiple connections has been made when parallel transmission groups are not supported between the two nodes involved in the XID exchange.

0012 The adjacent node has sent the Network Name (X′0E′, CP name) control vector in XID3 but indicates it does not support the Exchange State indicators.

0013 The DLC type indicated in the sent and received negotiation-proceeding XID3s is not in agreement.

0014 After sending a negotiation-proceeding XID3 with the Link Station Role field set to either “primary” or “secondary,” the adjacent node sends a negotiation-proceeding XID3 with this field set to “negotiable.”

0015 During a negotiation-proceeding XID3 exchange, the adjacent node indicated that it supports BIND pacing as a sender but not as a receiver; i.e., byte 10, bits 0–1 in XID3 are set to 10, which is not allowed.

0016 The node receiving the XID3 is attempting to activate a predefined TG, i.e., a TG that will be assigned a number in the range of 1 to 20, but the TG number sent in the adjacent node′s XID3 does not agree with the number that the receiving node has assigned to the TG.

0017 After two negotiation-proceeding XID3 exchanges, the Node Identification field of the received and sent XID3s have identical values. When both nodes contain negotiable link stations, link station roles must be resolved within two exchanges of XIDs after link station role negotiation has begun.

0018 The adjacent node is an APPN node but does not support adaptive BIND pacing as a sender and receiver.

0019 The receiving node does not support CP name changes on APPN connections, but has received a nonactivation XID3 with a CP name that differs from that received during the previous XID exchange.

001A The adjacent node is inconsistent in its support of parallel TGs. Support of parallel TGs between two nodes cannot change either in link-activation XID exchanges on different TGs or in successive XID exchanges on the same TG.

001B The adjacent node provides or requests CP services but does not support CP-CP sessions; i.e., bytes 8–9, bits 10–11 of the received negotiation-proceeding XID3 were set to 10, a setting combination not allowed for T2.1 nodes.

001C The adjacent node declares that its link station role is not primary, secondary, or negotiable; i.e., byte 19, bits 2–3 of the received negotiation-proceeding XID3 were set to 10, a value not allowed for T2.1 nodes.

001D The adjacent node supports two-way alternating as its transmit-receive capability while the receiving node supports two-way simultaneous and cannot negotiate down to a two-way alternating transmit-receive capability.

001E The adjacent node has not appended its CP name in the Network Name (X′0E′, CP name) control vector on XID3, but indicates that it supports CP-CP sessions and requests them from the receiver. Such a node is interpreted as a LEN end node that is requesting APPN function, which is not permitted.

001F The setting of the Intersubnetwork Link indicator of the TG Descriptor control vector received in XID3 is inconsistent with the receiving node′s system definition. This sense data value is issued only if both sender and receiver support the setting of this bit.

0020 The node type the adjacent node declares itself to be in its XID3 is one to which the receiving node cannot activate a TG.

0021 The setting of the Error Recovery Mode field of the HPR Capabilities (X′61′) control vector received during a negotiation-proceeding XID3 exchange is inconsistent with the receiving

node′s system definition. One node specified that error recovery is required, but the other node specified that no error recovery is required. HPR protocols will not be used on this TG. Sense data X′10160021′ is not carried in the XID Negotiation Error (X′22′) control vector.

0022 The adjacent node is an HPR node (i.e., it included an HPR Capabilities (X′61′) control vector in XID3), but the receiving node detected that it specified a maximum BTU size less than 768.

0023 The adjacent node is an HPR node (i.e., it included an HPR Capabilities (X′61′) control vector in XID3), but the receiving node detected that it specified an invalid ANR label length (i.e., less than 1 or greater than 8).

0024 The adjacent node is an HPR node (i.e., it included an HPR Capabilities (X′61′) control vector in XID3), but the receiving node detected that it specified an invalid CP NCE identifier length (i.e., less than 1 or greater than 8).

0025 The adjacent node is an HPR node (i.e., it included an HPR Capabilities (X′61′) control vector in XID3), but the receiving node detected that it specified an invalid route setup NCE identifier length (i.e., less than 1 or greater than 8).

0026 The adjacent node is an HPR node (i.e., it included an HPR Capabilities (X′61′) control vector in XID3), but the receiving node detected that the length of the HPR Transport Tower (X′81′) subfield of the control vector is inconsistent with the length of a field included in the subfield.

0027 The adjacent node has specified that it supports the Control Flows Over RTP (1402) option set but has not included the Control Flows Over RTP Tower (X′81′) subfield in the XID3 HPR Capabilities (X′61′) control vector.

0028 The adjacent node has specified an invalid value for the error recovery mode in the XID3 HPR Capabilities (X′61′) control vector.

0031 The XID3 received from the adjacent node does not contain an HPR Capabilities (X′61′) control vector. The port supports only HPR links.

0032 The RTP Supported indicator is set to 0 in the HPR Capabilities (X′61′) control vector of the XID3 received from the adjacent node. The port supports links only to nodes that support RTP.

0033 The Control Flows over RTP Supported indicator is set to 0 in the HPR Capabilities (X′61′) control vector of the XID3 received from the adjacent node. The port supports links only to nodes that support control flows over RTP.

0034 The LDLC Supported indicator is set to 0 in the HPR Capabilities (X′61′) control vector of the XID3 received from the adjacent node. The port supports links only to nodes that support LDLC.

0035 A negotiable or 0 TG number was received in XID3, but multiple links are defined between the paired switched ports. Use of predefined TG numbers is required.

0036 A predefined TG number received in XID3 was not defined at the receiver.

0037 A nonzero negotiable TG number was received in XID3, but another link between the paired switched ports is using a different TG number.

0038 A TG number of 0 was received in XID3, but an active link between the paired switched ports is using a predefined TG number.

0039 A “race” was detected (i.e., both nodes are simultaneously attempting to activate a switched virtual connection for the same link), but multiple virtual connections are not needed. According to the race resolution algorithm, the other virtual connection will be used for the link.

003A Parallel TG support is pre-requisite to MLTG support, but the received XID indicates that MLTG is supported but parallel TGs are not.

003B A link belonging to an MLTG may not traverse a connection network, but an XID was received over a connection network with MLTG support indicated.

003C The MLTG TG number is invalid (must be 0-20 or 240).

003D The maximum send size in the received XID is greater than the size that is currently being used for the existing MLTG links.

003E The received MLTG TG number is already defined for an existing single-link TG.

003F The branch extender field settings are inconsistent with those for the existing MLTG links.

0040 The maximum receive BTU size in the received XID is less than the size that is currently being sent over the existing MLTG links.

0041 The locally defined security level or user defined characteristics for the link over which the XID was received does not match the security level or user defined characteristics for the other MLTG links.

0044 The HPR Capabilities (X′61′) control vector received in XID3 indicates that Logical Data Link Control (LDLC) is supported, but the control vector does not include an IEEE 802.2 LLC (X′80′) HPR Capabilities subfield. Configuration services in the receiving node requires the subfield when LDLC is supported on the link being activated.

0045 Multiple defined links between a pair of switched ports is not supported by the local node. A link activation request was received for a defined link, but there is an active defined link between the paired switched ports.

0046 Multiple dynamic links across a connection network between a pair of switched ports is not supported by the local node. A link activation request was received for a dynamic link, but there is an active dynamic link between the paired switched ports across the same connection network.

1018 MU Sequence Exception: An SNA/DS transaction program has detected an improper sequence of SNA/DS MUs.

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

0001 A DMU has been received, but the MU ID has already been terminated.

0002 The MU ID state received from the partner is incompatible with the state in the MU ID registry.

0003. Reserved

0004 A previous terminate conversation indication has been ignored.

0005 An RRMU was received but was not followed by a Change Direction indicator (i.e., the RECEIVE_AND_WAIT verb issued after receiving the RRMU, returned something other than WHAT_RECEIVED=SEND).

1019 Invalid Restart Byte Position.

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

0001 The restart byte position value specified in the DCMU is greater than 1 plus the value of the last byte received in the CRMU. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify the invalid restart byte position in the DCMU and the values specified in the CRMU. Supplemental report 1 contains the restart byte position value in the DCMU. Supplemental report 2 contains the last structure received value in the CRMU. Supplemental report 3 contains the last byte received value in the CRMU. If this value was not specified in the CRMU, this report will be omitted.

0002 The receiver does not support the byte-count restart elective, and the restart byte position value specified in the DCMU is not the beginning of the LLID structure following the last successfully received LLID structure. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify the invalid restart byte position in the DCMU and the values specified in the CRMU: Supplemental report 1 contains the restart byte position value in the DCMU. Supplemental report 2 contains the last structure

received value in the CRMU. Supplemental report 3 contains the last byte received value in the CRMU. If this value was not specified in the CRMU, the report will be omitted.

0003 The receiver supports the byte-count restart elective, and the restart byte position value specified in the DCMU is not equal to 1 and is less than or equal to the last byte received value specified in the CRMU. When this SNA report code is used in an SNA condition report, it is accompanied by three supplemental reports that identify the invalid restart byte position in the DCMU and the values specified in the CRMU; Supplemental report 1 contains the restart byte position value in the DCMU. Supplemental report 2 contains the last structure received value in the CRMU. Supplemental report 3 contains the last byte received value in the CRMU. If this value was not specified in the CRMU, the report will be omitted.

101A Invalid Control Vector Sequence: A control vector was found containing a key that was invalid for the position of the control vector within a TDU.

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

0000 No specific code applies.

nnmm Byte 2 following the sense code contains the key (nn) of the vector previous to the one in error; byte 3 contains the key (mm) of the vector in error.

101C Invalid Data Received

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

0000 No specific code applies.

0001 Alteration of input data not allowed.

101D Insufficient Length: The length of the received signal is insufficient to contain additional required fields.

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

0000 No specific code applies.

0001 A BIND or RSP(BIND) was received that was too large to be extended. The BIND was rejected.

0002 An UNBIND was received that was too large to be extended. An UNBIND cleanup is sent on both session stages.

101E CP Capabilities Mismatch: The CP capabilities of the adjacent node as indicated in the Support Indicators field of the CP Capabilities GDS variable are unacceptable.

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 bit within the Support Indicators field of the received CP Capabilities GDS variable that was considered to be unacceptable by the receiving node. Values X′00′ – X′1F′ are the only valid settings.

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 not 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: A normal-flow request was received while either (1) the half-duplex flip-flop state was not *receive* or (2) the LU 6.2 full-duplex conversation state was not *send-receive* or *receive-only*.

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 or an EXPD RU was sent before the +RSP(EXPD) was received for the previous EXPD RU.

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 or BIND request was received 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.

0003 Pacing Response Indicator Incorrectly Set: The pacing indicator was set in a non-IPM response received while adaptive pacing was being used.

2012 Invalid Sense Data Received: A negative response was received that contains an SNA-defined sense data value that cannot be used for the sent request.

2013 Decompression Protocol Error: A request containing compressed data was received in error.

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

0000 No specific code applies.

0001 The decompressor received a compressed RU without an expected Reset decompression control sequence. The compressor and the decompressor are not synchronized.

0002 The decompressor received a compressed RU containing an invalid decompression control sequence. The compressor and the decompressor are not synchronized.

0003 The length of the decompressed RU did not match the length given in the compression header.

0004 The decompressor has determined that the compression header indicates an illegal compression algorithm was used. The compression algorithm was not agreed to during the session-activation negotiation.

0005 The decompressor has detected that the decompressed RU size exceeds the maximum RU size.

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; for example, RQD2|3 was received on an LU 6.2 full-duplex conversation.

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, or an LU 6.2 full-duplex conversation received a CD indicator with EC specified.

400C Brackets Not Supported: The bracket indicators (BBI, CEBI, and EBI) were specified incorrectly; e.g., a bracket indicator was set (BBI=BB, CEBI=CEB, or EBI=EB), but brackets are not used for the session.

400D CD Not Supported: The Change-Direction indicator was set, but is not supported.

400E Reserved

400F Incorrect Use of Format Indicator: The Format indicator (FI) was specified incorrectly; for example, FI was set with BCI=¬BC, or FI was not set on a DFC request.

4010 Alternate Code Not Supported: The Code Selection indicator (CSI) was set when not supported for the session.

4011 Incorrect Specification of RU Category: The RU Category indicator was specified incorrectly; for example, an expedited-flow request or response was specified with RU Category indicator = FMD.

4012 Incorrect Specification of Request Code: The request code on a response does not match the request code on its corresponding request.

4013 Incorrect Specification of (SDI, RTI): The Sense Data Included indicator (SDI) and the Response Type indicator (RTI) were not specified properly on a response. The proper value pairs are (SDI=SD, RTI=negative) and (SDI=¬SD, RTI=positive).

4014 Incorrect Use of (DR1I, DR2I, ERI): The Definite Response 1 indicator (DR1I), Definite Response 2 indicator (DR2I), and Exception Response indicator (ERI) were specified incorrectly; for example, a SIGNAL request was not specified with DR1I=DR1, DR2I=¬DR2, and ERI=¬ER.

4015 Incorrect Use of QRI: The Queued Response indicator (QRI) was specified incorrectly; for example, QRI=QR on an expedited-flow request.

4016 Incorrect Use of EDI: The Enciphered Data indicator (EDI) was specified incorrectly; for example, EDI=ED on a DFC request.

4017 Incorrect Use of PDI: The Padded Data indicator (PDI) was specified incorrectly; for example, PDI=PD on a DFC request.

4018 Incorrect Setting of QRI with Bidder′s BB: The first speaker half-session received a BB chain requesting use of a session (via LUSTAT(X′0006′)), but the QRI was specified incorrectly; that is, QRI = ¬QR.

4019 Incorrect Indicators with Last-In-Chain Request: A last-in-chain request has specified incompatible RH settings; for example, RQE*, CEBI=¬CEB, and CDI=¬CD.

4021 QRI Setting in Response Different From That in Request: The QRI setting in the response differs from the QRI setting in the corresponding request.

4022 Length-Checked Compression Not Supported: The Length-Checked Compression indicator (LCCI) was set on a request, but the receiving half-session does not support length-checked compression for this session.

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.

8007 Segmenting Error: First BIU segment had less than 10 bytes; or Mapping field sequencing error, such as first, last, middle; or segmenting not supported and Mapping field not set to BBIU, EBIU; or (in APPN) an expedited request or response was received segmented (see Note 2 located at the end of this section).

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

0000 No specific code applies.

0001 The node does not support receipt of segments, and a Mapping field value other than BBIU, EBIU was received. Sent in UNBIND.

0002 Interleaved BIND Segments Not Allowed: A BIND receiver that is in the middle of receiving segments of one BIND receives a segment from a different BIND; the receiver rejects both BINDs and disconnects the link.

8008 PU Not Active: The SSCP-PU secondary half-session in the receiving node has not been activated and the request was not ACTPU for this half-session; for example, the request was ACTLU from an SSCP that does not have an active SSCP-PU session with the PU associated with the addressed LU.

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

0000 No specific code applies.

0001 A physical unit name was specified for an independent LU session stage. The specified PU name is either unknown or in an invalid state.

0002 NO ALS is defined for use by the origin independent LU. An implementation-defined automatic logon (autolog) request was specified for the subject resource, but, when the session establishment was attempted, no ALS was found to be associated with the subject resource.

8009 LU Not Active: The destination address specifies an LU for which the SSCP-LU secondary half-session has not been activated and the request was not ACTLU.

800A Too-Long PIU: Transmission was truncated by a receiving node because the PIU exceeded a maximum length or sufficient buffering was not available.

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

0000 No specific code applies.

0001 A received PIU exceeded 255 buffers.

0002 The node signaling this error on an EXR is including additional information in an EXR extension.

800B Incomplete TH: Transmission received was shorter than a TH (see Note 1 located at the end of this section).

800C DCF Error: Data Count field inconsistent with transmission length.

800D Lost Contact: Contact with the link station for which the transmission was intended has been lost, but the link has not failed. If the difference between link failure and loss of contact is not detectable, link failure (X′8002′) is sent.

800E Unrecognized Origin: The origin address specified in the TH was not recognized.

800F Invalid Address Combination.

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

0000 The (DAF′, OAF′) (FID2) combination or the LSID (FID3) specified an invalid type of session, for example, a PU-LU combination.

0001 The FID2 ODAI setting in a received BIND is incorrect; the BIND is rejected.

8010 Segmented RU Length Error: An RU was found to exceed a maximum length, or required buffer allocation that might cause future buffer depletion.

8011 ER Inoperative or Undefined: A PIU was received from a subarea node that does not support ER and VR protocols, and the explicit route to the destination is inoperative or undefined.

8012 Subarea PU Not Active or Invalid Virtual Route: A session-activation request for a peripheral PU or LU cannot be satisfied because there is no active SSCP-PU session for the subarea node providing boundary function support, or the virtual route for the specified SSCP-PU (type 1 or type 2 nodes) or SSCP-LU session is not the same as that used for the SSCP-PU session of the type 1 or type 2 node′s PU or the LU′s subarea PU.

8013 Route Not Available: No route is available to connect the specified OSA and DSA for the specified COS.

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

Byte 2 indicates the environment in which the failure was detected:

00 Single network

01 Interconnected network: Failure was detected at a node in a subnetwork other than that of the NAU sending the activation request.

Byte 3 indicates the reason for the failure:

00 No Specific Code Applies: This means an error occurred, but none of the conditions listed below applies.

01 No Mapping Specified: A session-activation request cannot be satisfied because for each VR in the VR identifier list for the session, no VR to ER mapping is specified.

02 No Explicit Routes Defined: A session-activation request cannot be satisfied because each VR in the VR identifier list for the session maps to a corresponding ER that is not defined.

03 No VR Resource Available: A session-activation request cannot be satisfied because each VR specified in the VR identifier list for the session requires a node resource that is not available.

04 No Explicit Routes Operative: A session-activation request cannot be satisfied because no underlying ER is operative for any VR specified in the VR identifier list for the session.

05 No Explicit Route Can Be Activated: A session-activation request cannot be satisfied because no VR specified in the VR identifier list for the session mapped to a defined and operative ER that could be activated.

06 No Virtual Route Can Be Activated: A session-activation request cannot be satisfied because no VR specified in the VR identifier list for the session can be activated by the PU, though for at least one VR an underlying ER is defined, operative, and activated.

07 No Virtual Route Identifier List Available: A session-activation request cannot be satisfied because a VR identifier list is not available.

Note: If none of the virtual routes specified in the VR identifier list for the session is active or can be activated, the reported reason is set based on a hierarchy of failure events. The “highest” of the failures that occurred within the set of virtual routes is returned on the response. For example, if the VR manager receives a negative response to an NC_ACTVR request for a VR specified in the VR identifier list and for all other VRs in the list no VR to ER mapping is specified, then reason X′06′ is reported. The hierarchy of the failure reasons is in ascending numeric order, that is, reason X′02′ is higher than reason X′01′.

8014 No Path Exists to the Destination Node: Route selection services in the CP has determined from the topology database that no path exists to the destination node.

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

0000 No specific code applies.

0001 No route to the destination node exists for the specified class of service.

0002 Invalid APPN COS name received.

0003 The topology database indicates that the destination node is not available at this time; the node either has inconsistent data or is quiescing.

0004 The topology database indicates that the endpoint resources are depleted; the node is out of either half-session control blocks or message buffers.

0005 The length of the generated RSCV exceeds the maximum allowed.

0006 No path using only HPR TGs exists to the destination node.

0007 A BIND RSCV specifies only an interchange TG.

0008 The local node calculated an RSCV specifying a boundary node that it does not own.

8015 Path Not Available.

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

0000 No specific code applies.

0004 The intersubnetwork Locate failed because an intersubnetwork route did not exist that matched the requested class of service.

8017 PIU from Adjacent Pre-ER-VR Subarea Node Rejected: A PIU that requires intermediate path-control routing was received by a subarea node from an adjacent subarea node that does not support ER-VR protocols, but the receiving subarea node does not support intermediate path-control routing for adjacent subarea nodes that do not support ER-VR protocols.

8018 Management Services component is unable to find or recognize the name of the application transaction program specified in the request.

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

0000 No specific code applies.

0001 The application transaction program specified in the request is not recognized by PUMS.

0002 The Cascaded Resource Name List is unrecognized.

0003 The Destination Application Name is unrecognized.

0004 The Destination Instance Identifier is unrecognized.

8019 Routing Exception: A node is unable to perform the required routing function for a request.

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

0000 No specific code applies.

0001 Unknown Routing Group Name.

0002 Unknown Routing Group Name, Routing Element Name combination.

0003 Reserved

0004 No connection is available for level of service required. When this SNA report code is used in an SNA condition report, it is accompanied by a supplemental report containing the service parameter triplet (or combination of triplets) for which a connection could not be found.

0005 The Routing and Targeting Instructions GDS variable is required but is not present.

000C Upon receiving a Locate request for a DLUS-served dependent LU, the DLUS node has determined that at least one intersubnetwork TG on the path between the DLUS and the PLU was not between two extended border nodes.

801A Confirmation of Forwarding

0001 The identified request has been successfully forwarded by the reporting node. When this SNA report code is used within an SNA condition report, the reported-on destination list identifies the list of destinations to which the request has been forwarded.

801B Confirmation of Acceptance

0001 The identified request has been successfully received by the intended destination(s) at the reporting node. When this SNA report code is used within an SNA condition report, the reported on destination list identifies the list of destinations for which the request has been accepted.

801C Hop Count Exhausted

0001 The request has been forwarded by an excessive number of nodes (e.g., the count has been decremented at each node and has reached 0) and, therefore, the request could not be delivered to one or more destinations. Typically, this exception indicates that one or more nodes have incorrectly routed or directed the request. The exception may also indicate that the routing/directing count was not appropriately initiated according to network size.

801E Invalid Switching Mode: The Switching Mode field received in a network header was invalid or not supported by the receiving node.

801F Invalid Link Header: The link header in the received frame was not valid for the current 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 A frame in the supervisory or information format was received over a TG configured for no error recovery.

8020 Session Reset: The LU-LU session identified in the UNBIND is being deactivated because of a reset condition.

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

0000 No specific code applies.

0001 Virtual Route Inoperative: The virtual route used by the LU-LU session has become inoperative, thus forcing the deactivation of the identifed LU-LU session.

0002 Hierarchical Reset of Both XRF-active and XRF-backup Sessions: The XRF-backup session has failed; therefore, both the XRF-active and XRF-backup session are being reset.

0003 Virtual Route Deactivated: The identified LU-LU session had to be deactivated because of a forced deactivation of the virtual route being used by the LU-LU session.

0004 Route Extension Failure: The route extension used by the LU-LU session has become inoperative, thus forcing the deactivation of the identified LU-LU session.

0005 Route Extension Failure: The route extension used by the XRF-backup LU-LU session has become inoperative, thus forcing the deactivation of the identified XRF-backup LU-LU session.

0006 Virtual Route Inoperative: The virtual route used by the LU-LU session has become inoperative, thus forcing the deactivation via VR-INOP of the identifed XRF-backup LU-LU session.

0007 Third-Party Termination: The network operator caused the forced or cleanup termination of the LU-LU session.

0008 BFTERM has been received with no indication of the cause of the reset.

0009 Termination was requested by the dependent SLU with a TERMINATE SELF or character-coded logoff.

000A The identified LU-LU session had to be deactivated because its underlying RTP connection has become inoperative.

000B The identified LU-LU session had to be deactivated because its underlying RTP connection was deactivated.

000D After an operating system (MVS) or control point (VTAM) failure, an MNPS (multinode persistent sessions) application program has attempted recovery on a VTAM that owns one or more of the application program′s current session partners; the sessions involving these session partners are terminated in order to allow the application program to issue its new OPEN successfully; this sense data value is carried on the UNBIND for a terminated session.

8021 Path Switch Failure: The attempt to switch the path traversed by an RTP connection has failed.

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

0000 No specific code applies.

0001 The original path of the RTP connection is inoperative. Because the path switch failed, all sessions using the RTP connection will be terminated.

0002 The original path of the RTP connection is still operative. The RTP connection will continue operation over the original path.

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.

Rapid Transport Protocol (RTP) Error (Category Code = X′A0′)

This category indicates that an error was detected by RTP. Two types of RTP errors can occur:

- Semantic: message errors that cannot be detected without knowledge of the message context, i.e., the history of prior, relevant messages as well as nodal definition. Errors of this type are errors of meaning. An example of a semantic error is the detection of a duplicate transport connection identifier.
- Syntactic: message errors that can be detected without knowledge of the message context. These errors can be detected by knowing the rules used to build a message. Errors of this type are errors of form. An example of a syntactic error is the absence of a required control vector.

Category and modifier (in hexadecimal):

A001 RTP Protocol Error: A protocol violation has been detected. For example, during reassembly of a user message, the next sequential packet has the Start of Message bit set to 1.

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

0000 No specific code applies.

0003 The setting of the Start of Message bit in a received packet was unexpected. For example, during reassembly of a user message, the next sequential packet has the Start of Message bit set to 1.

0004 The setting of the End of Message bit in a received packet was unexpected. For example, a packet is received that contains data (i.e., the User Message

Length field is nonzero) and has the Last Message bit set to 1, but the End of Message bit is set to 0.

0008 The setting of the Last Message bit in a received packet was unexpected. For example, a packet with the Last Message bit set to 1 was received in a packet that filled a previously detected gap.

000B The setting of the Connection Qualifier/Source Identifier Field Present bits in a received packet was unexpected. For example, a packet was received with the TCID Assignor bit set to 1, but the Connection Qualifier/Source Identifier Field Present bits were not set to 01.

000C The setting of the Optional Segments Present bit in a received packet was unexpected. For example, a packet is received for which no active context is found. The Setup Packet bit is set to 1, but the Optional Segments Present bit is set to 0.

000D The setting of the Payload Offset/4 field in a received packet was unexpected. For example, the User Message Length field has a value greater than 0, but the Payload Offset/4 field points to an offset beyond the end of the packet.

000E The setting of the User Message Length field in a received packet was unexpected. For example, the User Message Length field has a value greater than 0, but the Payload Offset/4 field points to an offset from which there is insufficient remaining length for the data.

000F The setting of the Byte Sequence Number field in a received packet was unexpected. For example, the value specified in the Byte Sequence Number field was higher than the sequence number of the next expected packet (i.e., a new gap in the data stream is detected), but the Last Message bit was set to 1 in an earlier packet.

0014 The setting of the Target Resource Identifier Field Present bit of the Connection Setup segment in a received packet was unexpected. For example, the bit was set to 0 when target identification was expected by the receiver.

0016 The setting of the Rate-Based Flow/Congestion Control Used bit of the Connection Setup segment in a received packet was unexpected. For example, the Rate-Based Flow/Congestion Control Used bit was set to 0, but the receiver requires the use of rate-based flow/congestion control for the connection.

0017 A field setting in the Topic Identifier (X′28′) control vector in a received packet was unexpected. For example, the identified topic is not supported by the receiver.

0018 A field setting in the Network Identifier (X′03′) control vector in a received packet was unexpected. For example, the first byte of the Network Identifier field does not contain an uppercase letter from the type-1134 symbol string.

0019 A field setting in the Node Identifier (X′00′) control vector in a received packet was unexpected. For example, the first byte of the Node Identifier field does not contain an uppercase letter from the type-1134 symbol string.

0021 The setting of the RSEQ field of the Status segment in a received packet was unexpected. For example, the RSEQ field indicates that data has been received, but the data up to that sequence number has not yet been sent.

002D RTP has detected an error made by the using layer. For example, a parameter passed to RTP is outside its acceptable range.

0032 The beginning and ending sequence numbers for an acknowledged byte-span pair (ABSP) in a Status segment in a received packet were unexpected. For example, two ABSPs overlap.

0033 A field setting in the HPR Switching Information (X′83′) control vector in a received packet was unexpected. For example, the maximum packet size specified is less than 768 bytes.

0035 A field setting in the NCE Identifier (X′26′) control vector in a received packet was unexpected. For example, the length of the NCE Identifier field is greater than 8 bytes.

003A The using layer terminated abnormally.

003B The receiving RTP has insufficient storage to process a received request to establish a new RTP connection.

003D A Status segment has been received with a gap, indicating data that has already been sent and successfully acknowledged. This error occurs only on RTP connections where messages cannot arrive out of order.

A018 RTP Optional Segment or Control Vector Length Error: The value in the Length field added to the current byte offset within the embedding structure exceeds the actual length of the embedding structure, or the value in the Length field is inconsistent with the format definition for the optional segment or control vector.

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

mmnn The embedding structure is an optional segment or control vector for which X′mm′ is the Key field value. The control vector in error is identified by the Key field value X′nn′.

FFnn — The embedding structure is the RTP transport header. The optional segment or control vector in error is identified by the the Key field value X′nn′.

A019 — RTP Unexpected Optional Segment or Control Vector: RTP received an optional segment or control vector that is not valid for the current state of the connection. For example, a Connection Identifier Exchange segment was received from the partner that initiated the connection.

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

mmnn — An unexpected control vector was embedded within an optional segment or control vector. X′mm′ is the Key field value that identifies the embedding optional segment or control vector. X′nn′ is the Key field value of the unexpected control vector.

FFnn — An unexpected optional segment or control vector was embedded within the RTP transport header. The unexpected structure is identified by the Key field value X′nn′.

A01A — RTP Duplicate Optional Segment or Control Vector: RTP received two or more optional segments or control vectors with the same key. The number of occurrences of the optional segment or control vector is more than is valid for the current state of the connection. For example, two Status segments were received in the same packet.

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

mmnn — A duplicate control vector was embedded within an optional segment or control vector. X′mm′ is the Key field value of the embedding optional segment or control vector. X′nn′ is the Key field value of the duplicate control vector.

FFnn — A duplicate optional segment or control vector was embedded within the RTP transport header. X′nn′ is the Key field value of the duplicate structure.

A01B — RTP Missing Required Optional Segment or Control Vector: RTP received a packet that did not contain a required optional segment or control vector. For example, a Connection Setup segment was received that did not contain a Topic Identifier control vector.

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

mmnn — An optional segment or control vector did not contain a required control vector. X′mm′ is the Key field value that identifies the embedding optional segment or control vector. X′nn′ is the Key field value of the missing control vector.

FFnn	The RTP transport header did not contain a required optional segment or control vector. X′nn′ is the Key field value of the missing structure.

End of Chapter 10

Chapter 11. Function Management (FM) Headers

Introduction

For sessions that support FM headers, the request header (RH) contains a format indicator (FI) that, when *on*, indicates that an FM header is at the beginning of an FMD request unit (RU).

FM headers appear only at the beginning of an RU. An RU containing an FM header may appear anywhere within a chain. When the FM header is longer than one RU will hold, the header is continued in as many additional RUs of a chain as needed to hold it. Figure 11-1 and Figure 11-2 show the placement of FM headers within an RU:

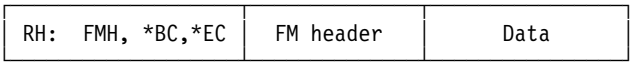


Figure 11-1. FM Header Contained in One RU

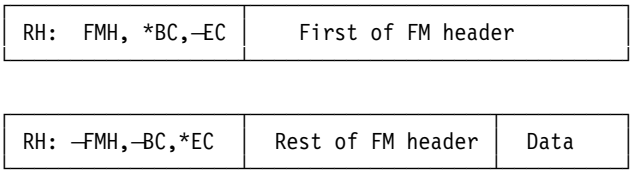
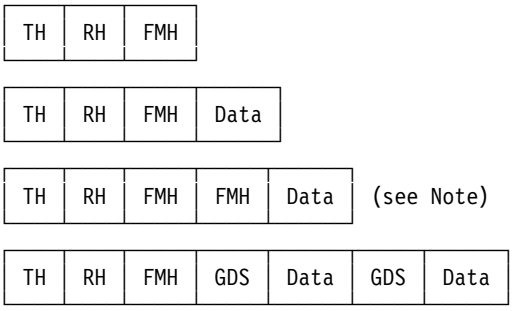


Figure 11-2. FM Header Contained in Two Contiguous RUs of a Chain

Figure 11-3 shows some instances where FM headers are used and Figure 11-4 identifies the logical unit (LU) types that use each FM Header.



FMH — Function Management (FM) Header
GDS — General Data Stream identifier
TH — Transmission Header
RH — Request/Response Header

Note: In LU type 6.2 a maximum of one FM header per RU is allowed.

Figure 11-3. Usage of FM Headers

LU Type	FM Header Type
0	None required, but may use any header
1	1, 2, 3
2	None
3	None
4	1, 2, 3
6.1	4, 5, 6, 7, 8, 10
6.2	5, 7, 12
7	None

Figure 11-4. LU Types That Support FM Headers

FM Header 1

This header is used to select a destination within a logical unit (LU). A destination may be represented by a device, a data set residing on a device, or merely a data stream. The LU initiates, interrupts, resumes, and concludes data traffic for the half-session using the FMH-1.

FM Header 1

Byte	Bit	Content
0		Length, in binary, of FMH-1, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-1 1 another FMH follows this FMH-1
	1–7	Type: 0000001

FM Header 1

Byte	Bit	Content
2	0– 3	Select desired medium for data (see Notes 1 and 2): 0000 console 0001 exchange 0010 card 0011 document 0100 nonexchange disk 0101 extended document 0110 extended card 0111 data set name select destination (see Note 3) 1000 word processing (WP) media 1 1001 WP media 2 1010 WP media 3 1011 reserved 1100 WP media 4 1101 reserved 1110 reserved 1111 reserved
	4– 7	Logical subaddress (see Note 2): 0000–1110 specific device in medium class 1111 any device in medium class (see Note 3)
3	0	SRI: Stack Reference indicator: 0 stack to be used is the sender's send stack 1 stack to be used is the receiver's send stack
	1	Demand select: 0 receiver may direct data to alternate medium/subaddress 1 receiver must direct data to specified medium/subaddress (spooling is prohibited)
	2– 3	Reserved
	4– 7	DSPs: data stream profiles: 0000 default (the DSP is implied by the Medium Select field) 0001 base 0010 general 0011 job 0100 WP raw-form text 0101 WP exchange diskette 0110 reserved 0111 Office Information Interchange level 2 1000 reserved 1001 reserved 1010 document interchange 1011 structured field 1100 reserved 1101 reserved 1110 reserved 1111 reserved

FM Header 1

Byte	Bit	Content
4		FMH-1 properties
	0–2	DSSEL: destination selection: 000 resume 001 end 010 begin 011 begin/end 100 suspend 101 end-abort 110 continue 111 reserved
	3	DST: data set transmission (see Note 6): 0 transmission exchange format 1 basic exchange format
	4	Reserved
	5	CMI: FMH-1 SCB compression indicator (see Notes 4 and 5): 0 no FMH-1 SCB compression 1 FMH-1 SCB compression (the first byte following the FMH(s) is a string control byte)
	6	CPI: compaction indicator (see Notes 4 and 5): 0 no compaction 1 compaction (the first byte following the FMH(s) is a string control byte)
	7	Reserved
5	0–7	ECRL: exchange record length if medium select = exchange or card; otherwise, reserved. For medium select = card, a hexadecimal value indicates maximum card length: 00000000 80-column length
6–7		Reserved (optional)
8		DSLEN: length of destination name (optional)
9 – n		DSNAME: destination name (optional; reserved when DSSEL = continue)

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 for specific values): Identifies the function that this FMH-2 is to perform
3 – n		Parameter fields (These fields provide the information needed to perform the selected function. They are different for each FMH-2 function, and are described in *SNA: Sessions Between Logical Units*.)

FM Header 2

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 FMH-1 SCB compression character
X′07′	Execute program offline
X′20′	Create data set
X′21′	Scratch data set
X′22′	Erase data set
X′23′	Password
X′24′	Add
X′25′	Replace
X′26′	Add replicate
X′27′	Replace replicate
X′28′	Query for data set
X′29′	Note
X′2B′	Record ID
X′2C′	Erase record
X′2D′	Scratch all data sets
X′2E′	Volume ID
X′AA′	Note reply (SRI is always *on*)

FM Header 3

This header handles data management tasks that are common to all destinations in the LU-LU session.

The FMH-3 format is identical to the FMH-2 format except that an FMH-3 does not have a Stack Reference indicator (SRI) in byte 2. An FMH-3 is used when information is needed or used by all destinations managed by the half-session. By contrast, an FMH-2 is used for a specific destination.

Two functions, the compaction table and the prime FMH-1 SCB compression character, can be sent in an FMH-2 or FMH-3. They are sent in an FMH-2 when they apply to a specific destination at the half-session and in an FMH-3 when they apply to all destinations at the half-session.

The FMH-3 functions are as follows:

Function Code	*Function*
X′02′	Compaction table
X′03′	Query for compaction table
X′04′	Prime FMH-1 SCB compression character
X′05′	Status
X′06′	Series ID

FM Header 4

This header carries a logical block command and its parameters that, together with information, apply to a logical block within a logical message as defined for Logical Message Service.

FM Header 4

Byte	Bit	Content
0		Length, in binary, of FMH-4, including this Length byte
1	0	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 half-session.

FM Header 5: Attach (LU 6.2)

Byte	Bit	Content
0		Length, in binary, of FMH-5, including this Length byte
1	0	Reserved
	1– 7	Type: 0000101
2– 3		Command code: X′02FF′ (Attach)

FM Header 5: Attach (LU 6.2)

Byte	Bit	Content
4	0– 2	Security Indicators
	0	Already-verified indicator: 0 user ID is not already verified 1 user ID is already verified (Password subfield not included in this Attach)
	1– 2	Persistent-verification indicator: 00 persistent verification not supported or needed 01 sign-on requested 10 already signed on (Password subfield not included in this Attach) 11 reserved
	3	Substituted password indicator: 0 If a Password subfield is present in this Attach, the password is in the clear. 1 A Password subfield is present and contains a substituted password.
	4	Program initialization parameter (PIP) presence: 0 PIP not present following this FMH-5 1 PIP present following this FMH-5 (see “PIP (X′12F5′) GDS Variable” on page 11-13 for format)
	5	Extended authentication indicator: 0 Authentication Token Data GDS variable does not follow the FMH-5. 1 Authentication Token Data GDS variable follows the FMH-5. (In this case, the other security bits in byte 4, bits 0 – 3 are also set to 0.)
	6– 7	Reserved
5		Length (j minus 5), in binary, of Fixed Length Parameters field (currently 3—future expansion possible)
6 – j		Fixed Length Parameters
6		Resource type: X′D0′ half-duplex basic conversation X′D1′ half-duplex mapped conversation X′D2′ full-duplex basic conversation X′D3′ full-duplex mapped conversation
7		Reserved
8(=j)	0– 1	Synchronization level: 00 none 01 confirm 10 confirm, sync point, and backout 11 reserved
	2	Reconnection support: 0 no 1 yes
	3– 7	Reserved
j+1 – q		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 minus (k+1)), in binary, of Access Security Information subfields

FM Header 5: Attach (LU 6.2)

Byte	Bit	Content
k+2 – m		Zero or more Access Security Information subfields (see "Access Security Information Subfields" on page 11-12 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 subfield
m+2 – n		Logical-Unit-of-Work Identifier Subfield
m+2		Length (values 1 to 17 are valid), in binary, of network-qualified LU name o.)
m+3 – w		Network-qualified LU network name
w+1 – w+6		Logical-unit-of-work instance number, in binary
w+7 – w+8 (=n)		Logical-unit-of-work sequence number, in binary
n+1 – p		Conversation Correlator Field
n+1		Length (values 0 to 8 are valid), in binary, of conversation correlator of sender
n+2 – p		Conversation correlator of the sending transaction: a 1- to 8-byte symbol-string type G identifier (unique between partner LUs) of the conversation being allocated via FMH-5 (an example construction of this field would be the composition of a transaction program instance identifier and a resource identifier)
p+1 – q		Attach Sequence Number Field
p+1		Length (values 0 and 8 are valid), in binary, of the sequence number for this Attach (always present if byte 4, bit 3 is set to 1; otherwise, optional)
p+2 – q		8-byte sequence number for this Attach, in binary, if preceding Length field = 8; otherwise, not present
Note:		Trailing Length fields (bytes p+1, 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

Access Security Information Subfields

Byte	Bit	Content
2 – i		Data: a symbol string identifying access security information known at the receiver; receivers may constrain such information to be type A, AE, GR, DB, or 1134, depending on the implementation. *Note 1:* 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 2:* When byte 4, bit 3 in the FMH-5 is set to 1, the password symbol string will be an 8-byte, type-G password substitute.

Note: The Access Security Information subfields may appear in any order in the Access Security Information field of the FMH-5. The profile and password subfields are not present if no user ID subfield is present. For full details of the conditions of presence and the receive checks for these subfields, see the ATTACH_SECURITY_CHECK procedure described in *SNA LU 6.2 Reference: Peer Protocols*.

PIP (X′12F5′) GDS Variable

The PIP GDS variable is present following FMH-5 Attach if the PIP Presence indicator (byte 4, bit 4) in the Attach is set to 1. Although not part of the Attach (i.e., the Attach Length byte does not include its length), it is shown here because of its affinity with the Attach.

PIP (X′12F5′) GDS 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 (X′12E2′) GDS structured fields, each of which has the following format (shown in "PIP (X′12E2′) GDS Structured Field" using 0-origin)

PIP (X′12E2′) GDS Structured Field: Zero or more of these structured fields are contained in a PIP GDS variable (see "PIP (X′12F5′) GDS Variable").

PIP (X′12E2′) GDS Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of PIP GDS structured field, including this Length field
2– 3		GDS indicator: X′12E2′
4 – m		PIP structured field data: type-G symbol string is valid

FM Header 5: Attach (Not LU 6.2)

This header flows from the program using the sending half-session to the attach manager of the receiving half-session. This header identifies the program at the receiving LU that it wishes to have attached. An FMH-5 can be followed by other FMHs (for example, FMH-6, FMH-8, and FMH-4), a logical record header (LRH), and FM data. Optionally, it can be sent with CD or EB.

FM Header 5: Attach (Not LU 6.2)

Byte	Bit	Content
0		Length, in binary, of FMH-5, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-5 1 another FMH follows this FMH-5
	1– 7	Type: 0000101
2– 3		FMH5CMD: command code: X′0202′ attach transaction program X′0204′ reset attached process X′0206′ data descriptor
4		FMH5MOD: modifier
5		FMH5FXCT: fixed-length parameters: X′00′ reset attached process X′02′ attach transaction program, data descriptor
6		ATTDSP
7		ATTDBA
8 – n		Resource names

FM Header 6

This header flows from a currently active transaction program using a sending half-session to a currently active transaction program using a receiving half-session.

FM Header 6

Byte	Bit	Content
0		Length, in binary, of FMH-6, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-6 1 another FMH follows this FMH-6
	1– 7	Type: 0000110

FM Header 6

Byte	Bit	Content
2– 3		Command code (CC2): For service transaction programs, the first byte of the command code identifies a transaction program and the second byte identifies a function within a transaction program.
4		FMH6MOD: modifier
	0	FMH6LNSZ: length of parameter length fields: 0 1-byte field 1 2-byte field
	1– 7	Reserved
5 – n		Fixed: total length of fixed length parameters (LF): This field contains the sum of the lengths of all fixed length parameters that are mandatory for the particular command code located in bytes 2 and 3. This field is either one byte or two bytes in length, based on the setting of FMH6LNSZ (0 = one byte; 1 = two bytes).
n + 1 – m		Fixed length parameters (FDy): the fixed length parameters are positional by command code
m + 1 – p		Variable: length field of first, positional variable-length parameter (LV1): This field is either one byte or two bytes in length, based on the setting of FMH6LNSZ (0 = one byte; 1 = two bytes). If the Length field (LVx) is equal to 0, then the variable parameter is omitted. The next positional variable-length parameter length (LV2) occurs in byte q + 1.
p + 1 – q		Variable-length positional parameter (VD). The LV and VD fields are replicated to represent x number of variable-length parameters according to command code.

FM Header 7: Error Description (LU 6.2)

LU type 6.2 uses this header, following a negative response (0846), to carry information that relates to an error on the session or conversation. For example, an FMH-7 and additional error information are sent when an FMH-5 (Attach) specifies a nonexistent transaction program name.

FM Header 7: Error Description (LU 6.2)

Byte	Bit	Content
0		Length (7), in binary, of FMH-7, including this Length byte
1	0	Reserved
	1– 7	Type: 0000111

FM Header 7: Error Description (LU 6.2)

Byte	Bit	Content
2– 5		SNA-defined sense data listed below; the phrases following the sense data values are the symbolic return codes provided to the application program in LU 6.2 verbs (see *SNA Transaction Programmer's Reference Manual for LU Type 6.2*) when the sense data is received. See Chapter 10, "Sense Data" on page 10-1 for additional details on the sense data.
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

Sense Data	Return Code — Secondary Return Code (if present)
1008600B	RESOURCE_FAILURE_NO_RETRY
10086021	ALLOCATION_ERROR — TPN_NOT_RECOGNIZED
10086031	ALLOCATION_ERROR — PIP_NOT_ALLOWED
10086032	ALLOCATION_ERROR — PIP_NOT_SPECIFIED_CORRECTLY
10086034	ALLOCATION_ERROR — CONVERSATION_TYPE_MISMATCH
10086041	ALLOCATION_ERROR — SYNC_LEVEL_NOT_SUPPORTED_BY_PGM
080F0983	ALLOCATION_ERROR — ACCESS_DENIED
080F6051	ALLOCATION_ERROR — SECURITY_NOT_VALID
080F80xx	ALLOCATION_ERROR — SECURITY_NOT_VALID
080F81xx	ALLOCATION_ERROR — SECURITY_NOT_VALID
08240000	BACKED_OUT (resync not in progress: The state of the entire subtree headed by the sender is backed out. See Note.)
08240001	BACKED_OUT (resync in progress: The state of one or more other partners of the sender is unknown. See Note.)
084B6031	ALLOCATION_ERROR — TP_NOT_AVAIL_RETRY
084C0000	ALLOCATION_ERROR — TP_NOT_AVAIL_NO_RETRY
08640000	DEALLOCATE_ABEND_PROG
08640001	DEALLOCATE_ABEND_SVC
08640002	DEALLOCATE_ABEND_TIMER
08890000	PROG_ERROR_NO_TRUNC or PROG_ERROR_PURGING
08890001	PROG_ERROR_TRUNC
08890100	SVC_ERROR_NO_TRUNC or SVC_ERROR_PURGING
08890101	SVC_ERROR_TRUNC

Note: On a BACKOUT verb, the two X'0824' sense data values cause an OK rather than BACKED_OUT return code, since the meaning of OK is equivalent to the meaning of BACKED_OUT on this verb. The two X'0824' sense data values may cause different secondary return codes (ALL_AGREED and LUW_OUTCOME_PENDING) to be returned for the SYNCPT, (MC_)PREPARE_FOR_SYNCPT, and BACKOUT verbs. Other verbs have no secondary return code for the the BACKED_OUT return code. For the mapping of these two sense data values to secondary return codes, see the sync point chapter in *SNA LU 6.2 Reference: Peer Protocols*.

FM Header 7: Error Description (Not LU 6.2)

This header is sent after a negative response (0846) to provide further information about an error.

FM Header 7: Error Description (Not LU 6.2)

Byte	Bit	Content
0		Length, in binary, of FMH-7, including this Length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-7 1 reserved
	1– 7	Type: 0000111
2– 5		ERPSENSE: SNA-defined sense data, which would appear on error response (see Chapter 10, “Sense Data” on page 10-1)
6– 7		ERPSEQ: sequence number of RU chain in which error was detected

FM Header 8

This header is used only with IMS/VS logical message services that use LU type 6.1 protocols. Refer to the IMS publications for the formats and meanings of the bytes in this header.

FM Header 10

This header is sent to prepare the session for a sync point. It may be sent with data. The RU chain has CDI = CD so that the receiver may, on the next flow, request a sync point or abort the unit of work.

FM Header 10

Byte	Bit	Content
0		Length, in binary, of FMH-10, including this length byte
1	0	FMH concatenation: 0 no FMH follows this FMH-10 1 another FMH follows this FMH-10
	1– 7	Type: 0001010
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 to transport the second Security Reply Data value from the PLU to the SLU. The function management header 12 (FMH-12) has the following format:

FM Header 12: Security

Byte	Bit	Content
0		Length (10), in binary, of FMH-12, including this Length byte.
1	0 1– 7	Reserved Type: 0001100
2– 9		When the *basic* LU-LU verification protocol is supported, this subfield contains the enciphered version of the clear random data received in the RSP(BIND). When the *enhanced* LU-LU verification protocol is supported, this subfield contains the DES Message Authentication Code value of a two-part string composed of: 1. the random data value sent in the BIND and 2. the random data value received on the RSP(BIND). The DES Message Authentication Code algorithm is a standard cryptographic algorithm used to generate a value that can be used to verify the contents of a data field. In both cases, the installation defined LU-LU password is used as the cryptographic key for the DES algorithm.

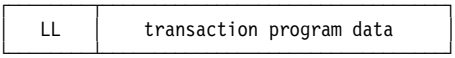
End of Chapter 11

Chapter 12. Presentation Services (PS) Headers

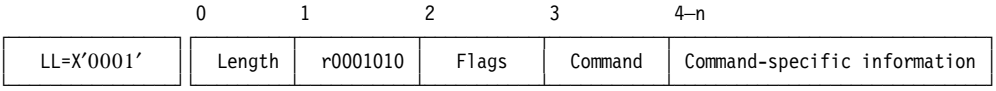
Presentation Services (PS) Headers

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

Typically, transaction program data exchanged over LU 6.2 sessions uses a 2-byte length field prefix called an LL. The LL specifies the number of bytes contained in the transaction program data plus 2 (the length of the LL field itself).



PS headers, however, deliberately violate this format. All PS headers are identified by an LL of X'0001' immediately preceding the header. X'0001' is an invalid LL value for use by transaction programs because the LL's value must include the length of itself, which is 2 bytes. All LLs indicating a length of less than 2 are reserved for use by the LU. The format of PS headers is shown below.



PS Header 10: Sync Point Control

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		Flags byte *Note:* The Flags byte is different for each sync-point command type specified in byte 3. The Flags byte is reserved and set to X'00' if the cold-start Exchange Log Name GDS variable indicates the partner does not use PS header byte 2 as the Flags byte.

PS Header 10: Sync Point Control

Byte	Bit	Content
Prepare flags:		
2	0	LU names indicator (used with presumed abort protocols; otherwise, reserved): 0 No LU names follow. 1 Network-qualified names of the LUs in this branch of the sync-point tree follow in consecutive LU Name (X′1215′) GDS variables, with the current sender of the Prepare adding its predecessor′s LU name to the series.
	1	Wait for outcome indicator: 0 Initiator′s TP wants agent to wait for resync to complete before replying to initiator. 1 Initiator′s TP does not want agent to wait for resync to complete before replying to initiator.
	2–4	Reserved
	5	LOCKS parameter indicator (used when PS header modifier, bytes 4–5, is request RECEIVE; otherwise, reserved): 0 LOCKS(SHORT) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL). 1 LOCKS(LONG) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL).
	6–7	Reserved
Request Commit flags:		
2	0	Support of the New LUWID PS header: 0 Not all participants in the subtree that include the sync-point manager sending this Request Commit support receipt of the New LUWID PS header. 1 All participants in the subtree that include the sync-point manager sending this Request Commit support receipt of the New LUWID PS header.
	1	Wait for outcome indicator (sent to last agent; otherwise, reserved): 0 Initiator′s TP wants agent to wait for resync to complete before replying to initiator. 1 Initiator′s TP does not want agent to wait for resync to complete before replying to initiator.
	2	Resource reliability indicator (used with presumed abort protocols when sending from not-last agent; otherwise, reserved): 0 Resource is subject to heuristic decisions. 1 Resource is not subject to heuristic decisions.
	3	OK to leave out of subsequent sync-point request indicator (used with presumed abort protocols when sending from a not-last agent; otherwise, reserved): 0 not OK to leave out 1 OK to leave out
	4	Initiator read-only reporting option (used with presumed abort protocols when sending to last agent; otherwise, reserved): 0 not reporting read only 1 reporting read only
	5	LOCKS parameter indicator (used when PS header modifier, bytes 4–5, is request RECEIVE; otherwise, reserved): 0 LOCKS(SHORT) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL). 1 LOCKS(LONG) was specified on PREPARE_TO_RECEIVE TYPE(SYNC_LEVEL).
	6–7	Reserved

PS Header 10: Sync Point Control

Byte	Bit	Content
Committed flags:		
2	0	Reserved
	1	Resync processing status (sent from last agent; otherwise, reserved): 0 not in progress: the state of the entire subtree is committed 1 in progress: the state of one or more agents of the sender is unknown
	2	Source of next LUWID: 0 to be generated locally by receiver of this Committed 1 New LUWID PS header provided by sender of this Committed
	3	OK to leave out of subsequent sync-point requests (used with presumed abort protocols when sending from last agent; otherwise, reserved): 0 not OK to leave out 1 OK to leave out
	4	Implied Forget expectation indicator (used with presumed abort protocols when sending to a not-last agent; otherwise, reserved): 0 implied Forget not expected 1 implied Forget expected
	5–7	Reserved
Forget flags:		
	0	Support of the New LUWID PS header (reserved if replying to Committed): 0 Not all participants in the subtree that includes the sync-point manager sending this Forget support receipt of the New LUWID PS header. 1 All participants in the subtree that include the sync-point manager sending this Forget support receipt of the New LUWID PS header.
	1	Resync processing indicator: 0 not in progress: the state of the entire subtree is committed 1 in progress: the state of one or more agents of the sender is unknown
	2–7	Reserved
HM flags:		
2	0–1	Reserved
	2	Source of next LUWID (sent in reply to Request Commit; otherwise, reserved): 0 to be generated locally by receiver of this HM 1 New LUWID PS header provided by sender of this HM
	3–7	Reserved
New LUWID flags:		
2	0–7	Reserved
End of description of various Flags bytes; main format description resumes		
3		Sync-point command type: X'05' Prepare X'06' Request Commit X'07' Committed X'08' Forget X'09' Heuristic Mixed X'0A' New LUWID

PS Header 10: Sync Point Control

Byte	Bit	Content
4 – n		Command-specific information (present only for Prepare (X′05′), Request Commit (X′06′), and New LUWID (X′0A′) commands. Data in this field depends on the value of byte 3, as shown below: Byte 3 value — Data in bytes 4 – n X′05′ (Prepare) — Modifier specifying next flow X′06′ (Request Commit) — Modifier specifying next flow X′0A′ (New LUWID) — LUWID for next transaction
If byte 3 =X′05′ (Prepare) or X′06′ (Request Commit):		
4– 5		Modifier specifying next flow (reserved, and may be omitted for half-duplex conversations when byte 3 = X′06′ and the Request Commit is being sent from a not-last agent to its initiator): X′0000′ request RECEIVE X′0001′ request DEALLOCATE X′0002′ request SEND X′0003′ request SEND/RECEIVE (used only with full-duplex conversations) *Note:* Bytes 4– 5 affect the Change Direction indicator (CDI) and Conditional End Bracket indicator (CEBI) settings of the RH for the last PS header in the sync-point sequence. For example, the CDI and CEBI bits on the Forget command are affected when Prepare was the first PS header received; similarly, the CDI and CEBI on the Committed command are affected when Request Commit was the first PS header received.
6– 1 3		Number of bytes sent and received (present if conversation is full-duplex; otherwise, omitted)
6– 9		Number of bytes sent, in binary
10– 1 3 (= n)		Number of bytes received, in binary
If byte 3 = X′0A′ (New LUWID):		
4		Length (values 10 to 26 are valid), in binary, of Logical-Unit-of-Work Identifier field (bytes 5 – n). Since the value may not be 0 (the LUWID may not be omitted), the value in byte 4 is 9 greater than the value in byte 5.
5 – n		Logical-unit-of-work identifier for the next logical unit of work
5		Length (values 1 to 17 are valid), in binary, of network-qualified LU name
6 – m		Network-qualified LU name: an optional 1- to 8-byte network ID and a 1- to 8-byte LU name, both of which are type-1134 symbol strings; when present, the network ID is concatenated ahead of the LU name, using a separating period; when the network ID is omitted, the period is also omitted
m + 1 – m + 6		Logical-unit-of-work instance number, in binary
m + 7 – m + 8 (= n)		Logical-unit-of-work sequence number, in binary

End of Chapter 12

Chapter 13. GDS Variables

General Content

In this chapter the following topics are covered:

- The definition of the *general data stream* (GDS)
- The description of GDS variables that are used by APPN and APPC service transaction programs that use LU 6.2 session protocols
- The description of the GDS variable used on the HPR control flow
- The description of GDS variables used for management services
- The description of GDS variables used for LU 6.2 application programs

GDS ID Description and Assignments

This section defines the *general data stream* (GDS), which is used in a variety of ways in SNA. 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. 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 13-1.

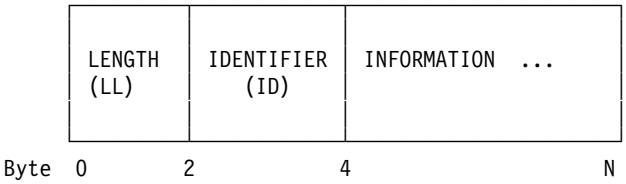


Figure 13-1. GDS Structured Field

Length (LL) Description

The LLID is a 4-byte field in which the two LL bytes are used to indicate the length of the LLID field itself (4 bytes) plus the data following the LLID; up to 32,763 bytes of data may follow the LLID. Values 0 and 1 of the LL are reserved for use as escape sequences; values 2 and 3 are not used. For example, a value of X′0001′ indicates a presentation services header, which is used for sync point management.

Bit 0 (high-order bit) of byte 0 is used as a length continuation (or not-last segment) indicator. If that bit is set to 1, the logical record is continued by another 2-byte LL; the ID occurs only following the first LL. The continuing LL is located immediately following the information bytes encompassed by the first LL.

The continuing LL might itself be continued. In other words, the length specified by the continuing LL might not be the entire remainder of the logical record; it might be followed by yet another LL. The amount of data spanned by each continuing LL can be any size convenient to the sender (including 0). Eventually, the chain of continuing LLs is ended by a final LL, i.e., one with the high-order (not-last) bit set to 0. The final LL may indicate a null information field follows (length = 2).

When an LLID encompasses a string of logical records identified by full LLIDs, the length of the string, determined by summing the (nested) encompassed LLs, equals the length definer of the (outer) 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. As usual, the asterisk (•) indicates “any value.”

Figure 13-2 (Page 1 of 4). Identifier Registry

GDS ID	Structured Field Owner
0000–01FF	3270
03••	3270
06••	3270
09••	3270
0B00–0EFF	3270
0F••	3270

Figure 13-2 (Page 2 of 4). Identifier Registry

GDS ID	Structured Field Owner
101•	3270
1030–1034	Print Job Restart
1058–105B	WorkStation Platform/2
1100–1104	SNA Character String
12••	LU 6.2 and APPN
13••	SNA/Management Services
140•	3820 Page Printer
1500	Dependent LU Requester/Server
1501	Subarea Routing Services
1520	DLSw Capabilities Exchange
1521	DLSw Capabilities Exchange Positive Response
1522	DLSw Capabilities Exchange Negative Response
1530–1531	SNA File Services
1532	SNA Condition Report
1533–154F	SNA File Services
1550–155F	SNA File Services
1570–158F	SNA/Distribution Services
4000–41FF	3270
4A00–4CFF	3270
71••	3250
8000–81FF	3270
C00•	Document Interchange Architecture
C100 – C104	Document Interchange Architecture
C105	SNA/Distribution Services
C10A – C122	Document Interchange Architecture
C123 – C124	SNA/Distribution Services
C219	Document Interchange Architecture
C300 – C345	Document Interchange Architecture
C350 – C361	SNA/Distribution Services

Figure 13-2 (Page 3 of 4). Identifier Registry

GDS ID	Structured Field Owner
C366 – C46F	Document Interchange Architecture
C500 – C56F	Document Interchange Architecture
C600 – C66F	Document Interchange Architecture
C7••	Graphical Display Data Manager
C800 – C87F	Document Interchange Architecture
C900 – CB0F	Document Interchange Architecture
CC00 – CC3F	Document Interchange Architecture
CD00 – CD3F	Document Interchange Architecture
CF0•	Document Interchange Architecture
D0••	Distributed Data Management
D3••	Document Content Architecture
D6••	Intelligent Printer Data Stream
D780 – D7BF	Facsimile Architecture
D820 – D821	AS/400 (5250)
D822 – D826	AS/400 (5394)
D930 – D95F	AS/400 (5250)
E10•	Level-3 Document Content Architecture
E20•	Level-3 Document Content Architecture
E30•	Level-3 Document Content Architecture
E40•	Level-3 Document Content Architecture
E50•	Level-3 Document Content Architecture
E60•	Level-3 Document Content Architecture
E70•	Level-3 Document Content Architecture
E80•	Level-3 Document Content Architecture
E90•	Level-3 Document Content Architecture
EA0•	Level-3 Document Content Architecture
EFFF	IBM Token-Ring Network PC Adapter
F000 – FEFF	Non-IBM Reserved Block

Figure 13-2 (Page 4 of 4). Identifier Registry

GDS ID	Structured Field Owner
FF••	Context-Dependent Block

GDS Variables for SNA Service Transaction Programs (STPs)

General Context

This section describes GDS variables that are used by SNA service transaction programs that use LU 6.2 session protocols (including over CP-CP sessions). See *SNA Transaction Programmer's Reference Manual for LU Type 6.2* for a complete list of the currently defined service TP names (TPNs); TPNs are specified in FMH-5s (Attaches).

Refer to Chapter 14, "SNA/DS FS1 Encodings" or to Chapter 15, "SNA/DS FS2 Encodings" for additional SNA/DS information and refer to "GDS Variables for Management Services" on page 13-77 and "GDS Variables for LU 6.2 Application Programs" on page 13-85 for information about GDS variables that are not specific to SNA service transaction programs.

See also "GDS ID Description and Assignments" on page 13-5 for a discussion of the general notion of general data stream (GDS) structured fields and a comprehensive list of the block assignments of GDS identifiers by architecture (or other use).

Descriptions of GDS Variables for SNA STPs

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

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

Byte	Bit	Content
0–1		Length (17 or n+1), in binary, of Change Number of Sessions GDS variable, including this Length field
2–3		GDS ID: X′1210′
4		Service flag:
	0–3	Reserved
	4–7	Request/reply indicator: 0010 request 1000 reply, function completed abnormal 1010 reply, function accepted but not yet completed
5		Reply modifier (reserved if byte 4, bits 4–7 = 0010): X′00′ normal—no negotiation performed X′01′ abnormal—command race detected X′02′ abnormal—mode name not recognized X′03′ reserved X′04′ normal—negotiated reply X′05′ abnormal—(LU,mode) session limit is 0
6		Action: X′00′ set (LU,mode) session limits X′01′ reserved X′02′ close
7		Drain immediacy:
	0–2	Reserved
	3	Source LU drain (reserved if byte 6 ≠ 02): 0 no (send BIS at next opportunity) 1 yes
	4–6	Reserved
	7	Target LU drain (reserved if byte 6 ≠ 02): 0 no (send BIS at next opportunity) 1 yes
8		Action flags:
	0–6	Reserved
	7	Session deactivation responsibility: 0 sender of Change Number of Sessions request (source LU) 1 receiver of Change Number of Sessions request (target LU) *Note:* Bytes 9–14 are reserved if byte 6 ≠ 0.
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 = X′02′)
16		Length (values 0 to 8 are valid; reserved if byte 15, bit 7 = 1), in binary, of mode name
17 – n		Mode name (omitted if byte 16 = X′00′)

Compare States (X′1213′) GDS Variable

Compare States (X′1213′) GDS Variable

Byte	Bit	Content
0– 1		Length (q+1 or r+1), in binary, of Compare States GDS variable, including this Length field
2– 3		GDS ID: X′1213′
4		Service Flags
	0– 3	Reserved
	4– 7	Request/reply indicator: 0010 request resync 1000 reply to resync, function completed abnormally: The partner′s specified LUW state was a legal value, but the partner′s state along with the local state do not comprise a legal combination. 1001 reply to resync, function completed normally
5		Sync point manager′s LUW state: X′01′ RESET X′03′ IN_DOUBT X′04′ COMMITTED X′05′ HEURISTIC_RESET X′06′ HEURISTIC_COMMITTED X′07′ HEURISTIC_MIXED
6		Flag Byte (reserved if Service flag ≠ 0010 or 1001):
	0	Reserved
	1	Resync processing status (reserved when sent from initiator): 0 resync not in progress: Byte 5 reflects the state of the entire subtree headed by the sender of this Compare States. 1 resync in progress: The state of one or more agents of the sender of this Compare States is unknown.
	2– 7	Reserved
7		Length, in binary, of Logical-Unit-of-Work Identifier field (values 10 to 26 are valid)
8 – n		Logical-Unit-of-Work Identifier
8		Length, in binary, of network-qualified LU name (values 1 to 17 are valid)
8 – w		Network-qualified LU network name
w + 1 – w + 6		Logical-unit-of-work instance number, in binary
w + 7 – w + 8 (= n)		Logical-unit-of-work sequence number, in binary
n + 1		Length (values 1 to 8 are valid), in binary, of conversation correlator
n + 2 – p		Conversation correlator of the transaction program that allocated the conversation that failed (see FMH-5 for the format of this correlator)
p + 1		Length (values 2 to 8 are valid), of session-instance identifier
p + 2 – q		Session-instance identifier of session being used by the conversation at the time of failure (see Chapter 8, “User Data Structured Subfields” on page 8-1 for the format of this identifier)
q + 1		Length (values 0 to 17 are valid), in binary, of the network-qualified name of the LU that created the conversation correlator carried in byte n+2.

Compare States (X′1213′) GDS Variable

Byte	Bit	Content
q+2 – r		Network-qualified name of the LU that created the conversation correlator
Note:		The network-qualified name of the conversation correlator creator is omitted if non-support of it is negotiated during the Exchange Log Name exchange. If the field is omitted, the length field (byte q+1) is also omitted.

Control Point Management Services Unit (X′1212′) GDS Variable

CP-MSU carries MS requests and data in general data stream (GDS) format.

Control Point Management Services Unit (X′1212′) GDS Variable

Byte	Bit	Content
0– 1		Length (m+1), in binary, of the CP-MSU
2– 3		GDS ID: X′1212′
4 – m		One or more MS major vectors, as described (using 0-origin indexing) in *SNA Management Services Formats*, and/or one or more of the following GDS variables if appropriate: X′1532′ SNA Condition Report: documented in Appendix B, “Common Structures.” Present if an SNA-registered condition was recognized by the management services application program or SNA/DS agent at the sending node, except in the case of SNA/File Services errors (when the report is contained within the FS Action Summary). X′1548′ FS Action Summary: defined by SNA/File Services. Present in a management services reply MU if a server object requesting SNA/FS action was present in the management services request MU. X′1549′ Agent Unit Of Work: defined by SNA/File Services. Present in a management services request MU if a Request Cancellation (X′8076′) major vector refers to another request MU, using its correlation value as its identifier. *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.

CP Capabilities (X′12C1′) GDS Variable

Immediately following CP-CP session activation, the CP Capabilities GDS variable is exchanged by the CP capabilities service transaction programs (REQ_CP_CAP_TP and CP_CAP_SON_TP) and describes the capabilities of the sending control point.

CP Capabilities (X′12C1′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12C1′
4 – n		GDS Variable Data
4– 7		Flow reduction sequence number (FRSN) (always defined for NNCP-NNCP sessions and also for ENCP-NNCP sessions when the NNCP supports registration of session endpoint TGs; otherwise reserved): a value that identifies the latest Topology Database Update GDS variable received by the sender of the CP Capabilities GDS variable from the receiver of the CP Capabilities GDS variable. The FRSN is an unsigned integer in the range of 0 to 2^{32}-1. A FRSN value of 0 indicates that the node requires a complete copy of the adjacent node′s topology database.
8– 1 1		Support indicators (bit is set to 1 if the sender supports the function):
8	0– 2	Retired (set to 100 by ENCPs; set to 111 by NNCPs)
	3	Registration of characteristics supported: the sending CP supports receipt of Register and Delete GDS variable requests that include resource characteristics (reserved for end nodes).
	4	Receipt of unknown control vectors in a TDU GDS variable supported: The sending CP supports the receipt of unknown control vectors in a Topology Database Update (TDU) GDS variable. Storing the unknown control vectors in the topology database, as well as forwarding them in an output TDU is also supported.
	5	RECEIVE_TDU_TP (X′22F0F0F4′) service transaction program supported: The sending CP supports receipt of TDUs on this session (always 1 between NNCPs; 1 from an NNCP to an ENCP when the NNCP supports registration of session endpoint TGs; reserved from end nodes).
	6	MS capabilities exchange supported: The sending CP supports the MS capabilities requests and replies.
	7	Bypass of Directed Locate Not Allowed indicator supported (reserved from end nodes or from an NNCP that does not support option set 1117): The sending NNCP honors the Bypass of Directed Locate Not Allowed indicator in the Find GDS variable.

CP Capabilities (X′12C1′) GDS Variable

Byte	Bit	Content
9	0	Retired (set to 1)
	1	Reserved
	2	Extended session services network node server support: The sending node supports acting as the network node server for ENs that use the APPN session services extensions (SSE) option set, such as for SLU-initiated and third-party-initiated sessions. (Support of SSE NNS does not imply support for SSE CP function.)
	3	Extended session services CP support: The sending (end or network) node supports the establishment of sessions to and from local LUs using the APPN session services extensions, such as for SLU-initiated or third-party-initiated sessions.
	4	Retired
	5	CP-CP session activation enhancements supported: The sending node (whether an EN or an NN) supports EN-initiated NN server selection and (if an NN) supports receiving a BIND without an RSCV from an adjacent NN.
	6	Extended border node supported: The sending node can act as an extended border node to support Locate and BIND flows that may span multiple subnetworks.
	7	Reserved
10	0	Reserved
	1	Topology awareness of CP-CP sessions supported: The sending node uses the CP-CP Session Support and Status field (byte 6, bits 3–4) in the TG Characteristics (X′47′) control vector (CV47) to indicate if the TG can support CP-CP sessions and if the sender has a contention-winner CP-CP session active or not on the TG. Otherwise (bit 1 set to 0), only byte 6, bit 3 in CV47 is used, which distinguishes only whether CP-CP sessions are supported or not supported, without indicating whether the contention-winner CP-CP session is active or not.
	2	Interior border node supported: The sending node can act as an interior border node to support Locate and BIND flows that may span multiple clusters within a network.
	3–5	Maximum Locate length - contains the maximum length Locate message that the CP capabilities sender can receive (in 3-bit exponent form). This field governs the length of Locate requests and replies (and indirectly the number of TGVs) that may be sent by an EN to its NNS. This field value also governs the length of Locates that may be received by ENs. **3-bit exponent form** is a 3-bit field indicating the number of kilobytes (K bytes) with values 0-7 used as an exponent of 2. Value 0 (B′000′) indicates 1K bytes (default), 1 (B′001′) indicates 2K, 2 (B′010′) indicates 4K, 3 (B′011′) indicates 8K, 4 (B′100′) indicates 16K, 5 (B′101′) indicates 32K, 6 (B′110′) indicates 64K, and 7 (B′111′) indicates 128K.
	6–7	Reserved
11		Reserved
12 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT. X′33′ ENCP Search Control control vector (optionally included only by ENCPs)

Cross-Domain Initiate (X′12C5′) GDS Variable

The CD-Initiate GDS variable is used in conjunction with the Locate, Find, and Found GDS variables to provide cross-domain session-initiation services.

Cross-Domain Initiate (X′12C5′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12C5′
4 – n		GDS Variable Data
4	0– 3	Format: X′0′ format 0 (only value defined)
	4– 7	Reserved
5	0	Reserved
	1– 3	Session initiation status: 000 initiate 001 queued 010 proceed 011 dequeue 100 session started 101 pending session characteristics 110 providing session characteristics 111 procedure terminated
	4	Role of sender: 0 OLU 1 DLU
	5	Extended session services CP support indicator: 0 not supported by sender 1 supported by sender
	6	Autologon query/support indicator (*query* indicator when sent on a Locate search request; *support* indicator when sent on a Locate search reply, and valid only when the associated Locate search request set this Autologon Query indicator to 1): 0 autologon support not queried (on a request); autologon not supported (on a reply) 1 autologon support query (on a request); autologon supported (on a reply)
	7	Incomplete TGVs indicator: 0 The endpoint TG vectors provided are the endpoint′s complete set. 1 The endpoint TG vectors provided are not the endpoint′s complete set.
6– 8		Initiate Parameters (except for the Session Polarity subfield, these three bytes are reserved unless the Session Initiation Status field = “initiate”)
6	0– 1	Retired
	2	SLU XRF support indicator: 0 SLU does not support XRF. 1 SLU supports XRF.

Cross-Domain Initiate (X′12C5′) GDS Variable

Byte	Bit	Content
	3	XRF backup-session request indicator: 0 SLU does not request activation of the XRF backup session. 1 SLU requests activation of the XRF backup session.
	4– 6	Initiate type: 000 search only (S); sender has not reserved session resources (invalid unless Session Polarity = “OLU is PLU”) 001 reserved 010 initiate only (I) (sender has reserved session resources) 011 reserved 100 queue only (Q) 101 reserved 110 initiate or queue (I/Q) 111 initiate or notify (I/N) (invalid unless ′session polarity′ = DLU is PLU)
	7	Session polarity (always set appropriately): 0 DLU is PLU. 1 OLU is PLU.
7	0	Session characteristics requested indicator: 0 not requested 1 requested
	1	LU status requested indicator (reserved unless Initiate Type = “S”): 0 not requested 1 requested
	2	Send session-release request indicator (reserved unless Initiate Type = “I/Q” and Session Polarity = “OLU is PLU”): 0 not requested 1 requested
	3	Suppress RSCV computation indicator (reserved unless Initiate Type = “S”) 0 Do not suppress. 1 Suppress.
	4	Partial route computation requested indicator (reserved unless Session Polarity = “DLU is PLU”) 0 partial route computation not requested 1 partial route computation requested
	5	End-to-end HPR-only route request indicator: 0 HPR-only route not requested 1 HPR-only route requested
	6– 7	Reserved

Cross-Domain Initiate (X′12C5′) GDS Variable

Byte	Bit	Content
8	0– 2	Queuing Parameters (reserved unless Initiate Type = “I/Q” or “Q”)
	0– 1	DLU Queuing Conditions (a 1 setting indicates that the OLU is willing to have its session initiation request queued if the associated condition is encountered at the DLU)
	0	Queue for LU not enabled.
	1	Queue for session limit exceeded.
	2	Queuing order indicator — specifies the queuing position for the session initiation request, should it become queued: 0 first-in, first-out (FIFO) queuing 1 last-in, first-out (LIFO) queuing
	3	Role of initiator: 0 ILU = OLU 1 ILU ≠ OLU
	4	Authentic mode-name indicator (when set, mode-name translation is bypassed): 0 not authentic 1 authentic
	5– 7	Notify Conditions (reserved unless Initiate Type = “I/N”; otherwise, a 1 setting indicates that a Notify message should be sent if the associated condition is encountered at the DLU)
	5	Notify on DLU enabled.
	6– 7	Reserved
9		Queue Parameters (reserved unless Session Initiation Status = “queued”)
	0– 1	Reason for Queue — indicates what queuing conditions caused the sender to queue or requeue the request (a 1 setting indicates that the session initiation attempt has been queued or requeued for the associated condition)
	0	LU not enabled
	1	Session limit exceeded
	2– 7	Reserved
10		Length, in binary, of mode name
11 – m		Mode name: 0 to 8 type-1134 symbol-string characters with optional (but not significant) trailing space (X′40′) characters
m + 1 – m + 2		Retired

Cross-Domain Initiate (X′12C5′) GDS Variable

Byte	Bit	Content
m + 3 – n		Control vectors, as described in "Control Vectors" in Chapter 9, "Common Fields" and in "Cross-Domain-Initiate Control Vectors" on page 13-21 below *Note:* The following control vectors may be included; they are parsed according to subfield parsing rule LT. X′0A′ User Request Correlation (URC) control vector (present for SLU-initiated sessions when provided by the SLU) X′2B′ Route Selection control vector (present in a CD-Initiate request when either the OLU or DLU is located beyond an interchange node and the Initiate Type field is other than "search only"; present in a CD-Initiate reply sent from the NNCP(OLU) to the ENCP(OLU) on a successful search for a Session Polarity field setting of "OLU is PLU") X′2C′ COS/TPF control vector (generated by an ENCP(OLU) if it supports mode-to-COS mapping — otherwise by its NN server; returned to the ENCP(OLU) by its NN server on a Locate search reply for use in a subsequent BIND): when present, the TPF field is reserved unless a Route Selection (X′2B′) control vector is also present X′2F′ LU Definition subfield (present if data is available to be sent and the flow is from the CP(SLU)) X′31′ BIND Image control vector (present on a search reply to return bytes 1-k of the BIND image [through and including the cryptography key, but excluding the PLU Name field], part of the needed information for a request having the Session Characteristics Requested indicator set to 1 — see the Device Characteristics [X′65′] control vector for the other returned session characteristics information) X′34′ LU Definition Override control vector (present if the model terminal support override values are available) X′3F′ SSCP(SLU) Capabilities control vector (present when the SSCP(SLU) supports suppressing notification of the SLU for specific session initiation errors) X′46′ TG Descriptor control vector (generated by APPN end nodes, or on replies by APPN network nodes for destination client LEN end nodes, but not received by end nodes; appears in ordered pairs with the TG Characteristics [X′47′] control vector; an EN(OLU) generates one pair for each active TG to a network node or a connection network; an EN(DLU) does the same, but also includes one pair for each active TG to the EN(OLU); in any case, the number of pairs cannot violate the limit of 1024 bytes on the total Locate search message) X′47′ TG Characteristics control vector (appears in ordered pairs with the TG Descriptor [X′46′] control vector) X′5D′ Subarea Message Transport control vector (present when searching an APPN network from an interchange node) X′5F′ Extended Fully Qualified PCID control vector (present in a third-party session initiation): contains the FQPCID for the session between the ILU and the OLU X′63′ Cryptography Key Distribution control vector (present only when key distribution information is to be transferred) X′64′ TCP/IP Information control vector (present to forward SLU TCP/IP information if that information was provided by the SLU) X′65′ Device Characteristics control vector (present on a search reply to return part of the needed information for a request having the Session Characteristics Requested indicator set to 1 — see the BIND Image [X′31′] control vector for the other returned session characteristics information) X′66′ Length-Checked Compression control vector (present when compression is supported)

Cross-Domain Initiate (X′12C5′) GDS Variable

Byte	Bit	Content
		X′68′ XRF/Session Cryptography control vector (present when session cryptography is required on an XRF-backup session)
		X′80′ User Data control vector (present only for SLU-initiated sessions, when the application program provides such data in its initiating request; a second User Data control vector is present to carry any excess over 253 user data bytes — in this case, the first User Data control vector carries 253 bytes of user data, while the second carries up to two additional user data bytes)
		X′81′ LU Status control vector: contains the information provided in response to a request having the LU Status Requested indicator set to 1
		X′82′ Additional Properties control vector: contains additional information associated with the CD-Initiate.

Cross-Domain-Initiate Control Vectors

User Data (X′80′) CD-Initiate Control Vector

User Data (X′80′) CD-Initiate Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′80′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – p		Vector Data
2 – p		User data: user-specific data to be passed in the CINIT User Data field

LU Status (X′81′) CD-Initiate Control Vector

The LU Status control vector carries DLU status information in a CD-Initiate reply in response to an OLU application program request for this information.

LU Status (X′81′) CD-Initiate Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′81′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – n		Vector Data
2– 5 (= n)		LU Status X′01′ control list — see Chapter 9, “Common Fields”

Additional Properties (X′82′) CD-Initiate Control Vector

The Additional Properties control vector carries various information associated a CD-Initiate reply.

Additional Properties (X′82′) CD-Initiate Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′82′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – n		Vector Data
2– 3 (= n)		An integer representing the weight of the entire path contained in the RSCV (X′2B′) control vector.

Delete Resource (X′12C9′) GDS Variable

The Delete Resource (X′12C9′) GDS variable requests a network node server to delete one or more entries from its directory. The network node server returns a Delete Resource (X′12C9′) GDS variable reply to report an error; no reply is sent if the deletion is completely successful.

Delete Resource (X′12C9′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12C9′
4 – n		GDS Variable Data
4 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” and in “Delete Control Vectors” on page 13-24. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X′80′ Command Parameters control vector (always present, always first) **For a Delete request**: X′37′ Directory Entry Correlator control vector (optionally present; paired with, and immediately preceding, a X′3C′ or X′3D′, control vector to provide error correlation of the reply data to the corresponding request data; if paired with a control vector that causes an error, the correlator is returned to provide a pointer to the data in error) X′3C′ Associated Resource Entry control vector (optionally present to indicate a hierarchical relationship to the following X′3D′ control vectors, e.g., the ENCP for the LUs to be deleted) X′3D′ Directory Entry control vector (present one or more times, not to exceed the length limit of 1024 bytes for the full Delete GDS variable X′4A′ Real Owning Control Point control vector: present when the type X′00F4′ (ENCP) Associated Resource Entry control vector in the hierarchy identifies a surrogate for the real owning control point; specifies the true control point that owns the resource identified in the Directory Entry control vector preceding it **For a Delete reply**: X′36′ Directory Error control vector (always present) X′37′ Directory Entry Correlator control vector (present when provided in the Delete request; returned in a reply to indicate the directory entry on which the error occurred)

Delete Control Vectors

Command Parameters (X′80′) Delete Control Vector

Command Parameters (X′80′) Delete Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X′80′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2	0	Request/reply indicator: 0 Delete request 1 Delete (negative) reply
	1	Delete directory entry condition (reserved on a Delete reply): 1 Delete a directory entry unconditionally and any subordinate directory entries. (only value defined)
	2–7	Reserved

Do Know (X′1217′) GDS Variable

The Do Know GDS variable precedes a Compare States GDS variable in a resync transaction flow. It identifies the log name of the local LU and indicates a decision for a unit of work (i.e., to commit or back out) is available. The Compare States GDS variable carries the rest of the information.

Do Know (X′1217′) GDS Variable

Byte	Bit	Content
0– 1		Length (7 to 70), in binary, of Do Know GDS variable, including this Length field
2– 3		GDS ID: X′1217′
4		Flags:
	0– 7	Reserved
5 – n		Name of the log at this LU
5		Length (values 1 to 64 are valid), in binary, of the local LU′s log name
6 – n		Local LU′s log name: a type-AE symbol string

Don′t Know (X′1219′) GDS Variable

The Don′t Know GDS variable indicates that a decision to commit or back out a unit of work has been delayed. It may be sent in reply to received Exchange Log Name and Compare States GDS variables.

Don′t Know (X′1219′) GDS Variable

Byte	Bit	Content
0– 1		Length (6), in binary, of Don′t Know GDS variable, including this Length field
2– 3		GDS ID: X′1219′
4– 5		Reserved

Exchange Log Name (X′1211′) GDS Variable

Exchange Log Name (X′1211′) GDS Variable

Byte	Bit	Content
0– 1		Length (p+1, r+1, s+1, t+1, or u+1), in binary, of Exchange Log Name GDS variable, including this Length field
2– 3		GDS ID: X′1211′
4		Service flag:
	0– 3	Reserved
	4– 7	Request/reply indicator: 0010 request 1000 reply, function completed abnormally: A log name or warm/cold log status mismatch was detected. 1001 reply, function completed normally
5		Sync point manager support and status flags:
	0	Reserved
	1	Presence of the LU name of the creator of the conversation correlator in Compare States: 0 not present 1 present
	2	Ability of the LU to treat byte 2 of the PS header as a Flags byte and accept the X′08240001′ sense data value (Backout Initiated—Resync in Progress) in FMH-7: 0 not able: Byte 2 of the PS header is reserved; X′08240001′ is not accepted in FMH-7. 1 able: Byte 2 of the PS header contains flags; X′08240001′ is accepted in FMH-7.
	3	Sync point manager support of presumed abort protocols: 0 not supported 1 supported
	4	Reserved
	5	Partner log name validation (Reserved if partner log name field is not present): 0 required 1 not required
	6	Extended capabilities: 0 not included 1 included
	7	Log status: 0 cold 1 warm
6		Length (values 1 to 17 are valid), in binary, of network-qualified LU name
7 – n		Network-qualified LU name
n + 1 – p		Name of the log at this LU
n + 1		Length (values 1 to 64 are valid), in binary, of the local LU′s log name
n + 2 – p		Local LU′s log name: a type-AE symbol string
p + 1 – r		Name of the log at the partner LU (may be included if log status is “warm” and request/reply indicator is “request”)
p + 1		Length (values 1 to 64 are valid), in binary, of the partner LU′s log name
p + 2 – r		Partner LU′s log name, a type-AE symbol string

Exchange Log Name (X′1211′) GDS Variable

Byte	Bit	Content
r+1(=s)		Actual sync point manager support and status flags (may be included if log status is "warm"; must be preceded by partner LU's log name if request/reply indicator is "request"):
	0	Reserved
	1	Presence of the LU name of the creator of the conversation correlator in Compare States: 0 not present 1 present
	2	Ability of the LU to treat byte 2 of the PS header as a Flags byte and to accept the X′08240001′ sense data value (Backout Initiated—Resync in Progress) in FMH-7: 0 not able: Byte 2 of the PS header is reserved; X′08240001′ is not accepted in FMH-7. 1 able: Byte 2 of the PS header contains flags; X′08240001′ is accepted in FMH-7.
	3	Sync point manager support of presumed abort protocols: 0 not supported 1 supported
	4–5	Reserved
	6	Extended actual capabilities: 0 not included 1 included
	7	Reserved
s+1 – t		Extended sync point manager capabilities length and field (must be preceded by actual sync point manager support and status flags if log status is "warm")
s+1		Length (values 1 to 8 are valid), in binary, of the extended sync point manager capabilities field
s+2 – t		Extended sync point manager capabilities field
s+2		First byte of extended sync point manager capabilities field
	0	Don't know response: 0 not supported 1 supported
	1	Batch resync protocols: 0 not supported 1 supported
	2–7	Reserved
t+1 – u		Extended sync point manager actual capabilities length and field (may be included if log status is "warm"; must be preceded by extended sync point manager capabilities field)
t+1		Length (values 1 to 8 are valid), in binary, of the extended sync point manager actual capabilities field
t+2 – u		Extended sync point manager actual capabilities field
t+2		First byte of extended sync point manager actual capabilities field
	0	Don't know response: 0 not supported 1 supported
	1	Batch resync protocols: 0 not supported 1 supported
	2–7	Reserved

FID2 Encapsulation (X′1500′) GDS Variable

The FID2 Encapsulation GDS variable transports control session information and data between an SSCP and a dependent LU or its associated PU.

FID2 Encapsulation (X′1500′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′1500′
4 – n		GDS Variable Data
4– 5		Length (m–3), in binary, of the FID2 PIU, including this Length field
6 – m		FID2 PIU
m + 1 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′0E′ Network Name control vector (contains the name of the PU, type X′F1′; see Note) X′25′ Security ID Control control vector (see Note) X′30′ Assign LU Characteristics control vector (see Note) X′43′ Extended SDLC Station control vector (see Note) X′46′ TG Descriptor control vector (see Note) X′51′ DLUR/S Capabilities control vector (see Note) X′60′ Fully Qualified PCID control vector (always present) X′81′ XID Image control vector (see Note) *Note:* See the “Format Changes” chapter in *APPN Dependent LU Requester Architecture Reference* for the conditions of presence for the above control vectors in this GDS variable.

FID2 Encapsulation Control Vectors

XID Image (X′81′) FID2 Encapsulation Control Vector

XID Image (X′81′) FID2 Encapsulation Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′81′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – n		XID I-field image: the bytes received in the information field of the XID response

Find Resource (X′12CA′) GDS Variable

The Find Resource (X′12CA′) GDS variable is used to request a node to search its directory for the search arguments provided.

Find Resource (X′12CA′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		GDS ID: X′12CA′
4 – n		GDS Variable Data
4 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” and in “Find Control Vectors” on page 13-31. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X′80′ Command Parameters Find control vector (always present, always first) X′3C′ Associated Resource Entry control vector: used to identify the search origin end node CP or network node server CP information to be saved at the search destination (e.g., in the server′s directory as a cache entry); hierarchical associations are indicated by the order of X′3C′ and X′3D′ control vectors, those appearing first being hierarchically above those that follow (an EN(OLU) generates one for its own CP; an EN(DLU) receives one for the OLU′s network node server CP and, if the OLU resides in an end node, one for the OLU′s ENCP) (this control vector is omitted if it would duplicate a following X′3D′ control vector that indicates LU=CP) X′3D′ Directory Entry control vector (always present): provides information about the search origin X′3E′ Directory Entry Characteristic control vector (present when characteristics are associated with the search origin; follows a X′3D′ control vector in order to cache a characteristic of a resource; precedes a X′40′ control vector if one is present) X′40′ Real Associated Resource Entry control vector (always type X′F6′=NNCP): identifies the real network node server of the resource identified in a preceding Directory Entry control vector (present only when an Associated Resource Entry control vector in the hierarchy does not represent the network node server of the target resource, but rather is a surrogate) X′4A′ Real Owning Control Point control vector: present when the type X′00F4′ (ENCP) Associated Resource Entry control vector in the hierarchy identifies a surrogate for the real owning control point; specifies the true control point that owns the resource identified in the Directory Entry control vector preceding it X′26′ NCE Identifier control vector (present when the node sending the accompanying CD-Initiate GDS variable is HPR capable and supports the RTP 1401 option set; although it is sent, this information is not used in the HPR protocols currently defined): identifies the component within the node that does processing for the origin LU X′82′ Search Argument Directory Entry Find control vector (always present): used to specify the search argument directory entry

Find Control Vectors

Command Parameters (X′80′) Find Control Vector

Command Parameters (X′80′) Find Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′80′ (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields")
2 – m		Vector Data
2	0	Origin information present indicator: 1 present (only value defined)
	1	Verification not permitted indicator: 0 Verification is permitted; if the receiving node cannot satisfy the request locally, Locate processing should continue. 1 Verification is not permitted; no additional search will be initiated by the receiving node.
	2	Bypass of directed Locate not allowed indicator: 0 Bypass (e.g., for topology nonverify) is permitted. 1 The directed Locate must be performed; processing may not be bypassed by using any local information (e.g., topology information).
	3	Search of branch uplink prohibited: 0 The search may be forwarded across a branch uplink. 1 The search may not be forwarded across a branch uplink.
	4	Reserved
	5– 6	Retired
	7	Verification not required indicator: 0 Verification is required. 1 Verification is not required; nonverify Locate processing may be used to locate the search argument.

Command Parameters (X′80′) Find Control Vector

Byte	Bit	Content
3	0	Subarea OLU indicator: 0 Within the local topology subnetwork, the search has not crossed an APPN-subarea network boundary between the OLU and the DLU. 1 Within the local topology subnetwork, the search has crossed an APPN-subarea network boundary between the OLU and the DLU.
	1	Reserved
	2	DLUS-served LU indicator: 0 The LU is not served by a dependent LU server. 1 The LU is served by a dependent LU server.
	3	Interchange node search (reserved for end nodes): 0 Do not restrict search to just the subarea network. 1 Restrict search to just the subarea network.
	4	Origin of Locate/Find (reserved for end nodes): 0 The origin is a network node. 1 The origin is a central directory server.
	5	Function of Locate/Find (reserved for end nodes): 0 This request is a referred Locate. 1 This request is to verify whether the resource identified by the Find search argument directory entry is currently available in the destination node.
	6	Owning CP respond Indicator: 0 The CP of the DLUS node owning the DLUS-served LU should reply to the Locate/Find. 1 The CP of the DLUR node owning the DLUS-served LU should reply to the Locate/Find.
	7	Surrogate resource owner indicator: 0 This Find is from the real owner of the OLU. 1 This Find may be from a surrogate owner of the OLU.

Command Parameters (X′80′) Find Control Vector

Byte	Bit	Content
4(=m)	0	Name change support indicator: 0 Target resource name change is not supported on Locate search reply. 1 Target resource name change is supported on Locate search reply.
	1	LU name resolution indicator: 0 Generic LU name resolution can be performed. 1 Generic LU name resolution cannot be performed.
	2	Inauthentic Net ID indicator: 0 The net ID of the DLU specified in Find CV X′82′ is authentic. 1 The net ID of the DLU specified in Find CV X′82′ is inauthentic.
	3	Larger reply allowed indicator. When set to a value of 1 (LRA), this field indicates that a request for a larger Locate reply length is allowed. If the destination LU requests a larger Locate reply length, a subsequent directed Locate request is sent by the origin over a path that can support a larger Locate reply (if one is available). LRAI is always set to LRA on a broadcast Locate request. When set to a value of 0 (¬ LRA), a request for a larger Locate reply length is not allowed. In this case, the origin will not send a subsequent Locate to obtain a larger reply; however, a subsequent Locate may be sent for other reasons (e.g., if the DLU indicated Resubmit-Directed). 0 Larger reply not allowed (¬ LRA) 1 Larger reply allowed (LRA)
	4– 6	Maximum Locate reply length - indicates the largest Locate reply (in 3-bit exponent form) that may be returned by the Locate reply sender. For directed Locate requests, this field indicates the smallest maximum Locate message length supported along the CP-CP session path traversed by the directed Locate. For broadcast Locates this field is always set to the largest Locate message length that is supported by every NN in the network. When a Locate request is sent by an EN, this field is set to the maximum Locate length value previously sent by the EN in the CP_Capabilities GDS variable. **3-bit exponent form** is a 3-bit field indicating the number of kilobytes (K bytes) with values 0-7 used as an exponent of 2. Value 0 (B′000′) indicates 1K bytes (default), 1 (B′001′) indicates 2K, 2 (B′010′) indicates 4K, 3 (B′011′) indicates 8K, 4 (B′100′) indicates 16K, 5 (B′101′) indicates 32K, 6 (B′110′) indicates 64K, and 7 (B′111′) indicates 128K.
	7	Reserved

Search Argument Directory Entry (X′82′) Find Control Vector

Search Argument Directory Entry (X′82′) Find Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′82′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – m		Vector Data
2– 3		Resource type: X′00F3′ logical unit X′00F4′ ENCP X′00F6′ NNCP

Search Argument Directory Entry (X′82′) Find Control Vector

Byte	Bit	Content
4 – m		Resource name: a 1- to 17-byte name consisting of an optional qualifier concatenated to a 1- to 8-byte type-1134 symbol-string name; when present, the qualifier contains a 1- to 8-byte type-1134 symbol-string network ID concatenated with a period (which is omitted if the network ID is omitted) *Note:* The network ID is always present when different from the network ID of the receiver.

Found Resource (X′12CB′) GDS Variable

The Found Resource (X′12CB′) GDS variable is a positive reply to a Find Resource (X′12CA′) GDS variable; it provides the requested data.

Found Resource (X′12CB′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12CB′
4 – n		GDS Variable Data
4 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” and in “Found Control Vectors” on page 13-36. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′80′ Command Parameters Found control vector (always present, always first) X′3C′ Associated Resource Entry control vector: used to identify the search destination end node CP or network node server CP to be saved at the search origin (e.g., in the server′s directory as a cache entry); hierarchical associations are indicated by the order of X′3C′ and X′3D′ control vectors, those appearing first being hierarchically above those that follow (an EN(DLU) generates one for its own CP; an EN(OLU) receives one for the DLU′s network node server CP and, if the DLU resides in an end node, one for the DLU′s ENCP) (this control vector is omitted if it would duplicate a following X′3D′ control vector that indicates LU=CP) X′3D′ Directory Entry control vector: identifies the requested destination directory entry (always present) X′3E′ Directory Entry Characteristic control vector (present when characteristics are associated with the destination directory entry; follows the associated X′3D′ control vector; precedes a X′40′ control vector, if one is present) X′40′ Real Associated Resource Entry control vector (always type X′F6′=NNCP): identifies the real network node server of the resource identified in a preceding Directory Entry control vector (present only when an Associated Resource Entry control vector in the hierarchy does not represent the network node server of the target resource, but rather is a surrogate) X′4A′ Real Owning Control Point control vector (present when the type X′00F4′ [ENCP] Associated Resource Entry control vector in the hierarchy identifies a surrogate for the real owning control point): specifies the true control point that owns the resource identified in the Directory Entry control vector preceding it X′26′ NCE Identifier control vector (present when the node sending the accompanying CD-Initiate GDS variable is HPR-capable and supports the RTP save option set): identifies the component within the node that does processing for the destination LU

Found Control Vectors

Command Parameters (X′80′) Found Control Vector

Command Parameters (X′80′) Found Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′80′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – m		Vector Data
2	0	Target information present indicator: 1 present (only value defined)
	1	Verification indicator: 0 verify performed 1 verify not performed
	2	DLUS-served LU indicator: 0 The LU is not served by a dependent LU server. 1 The LU is served by a dependent LU server.
	3– 4	Retired
	5	Reserved
	6	Owning CP responder indicator: 0 The CP of the DLUS node owning the DLUS-served LU has responded to the Locate/Find. 1 The CP of the DLUR node owning the DLUS-served LU has responded to the Locate/Find.
	7	Wild-card directory entry: 0 The directory entry for this located resource is an explicit or partially specified name. 1 The directory entry for this located resource is a wild-card entry.
3(=m)		Flag Extension Byte
	0	Subarea DLU indicator: 0 Within the local topology subnetwork, the search has not crossed an APPN-subarea network boundary between the DLU and the OLU. 1 Within the local topology subnetwork, the search has crossed an APPN-subarea network boundary between the DLU and the OLU.
	1	Resubmit directed Locate search indicator. 0 Do not resubmit a directed search 1 Resubmit a directed search because a larger Locate reply size (specified in byte 3, bits 2-4) is desired.
	2– 4	Maximum Locate reply length desired - indicates the maximum Locate reply length desired (in 3-bit exponent form) by the Locate reply sender. This length may be larger than the actual Locate reply length indicating that the reply sender would like to include more TGVs. **3-bit exponent form** is a 3-bit field indicating the number of kilobytes (K bytes) with values 0-7 used as an exponent of 2. Value 0 (B′000′) indicates 1K bytes (default), 1 (B′001′) indicates 2K, 2 (B′010′) indicates 4K, 3 (B′011′) indicates 8K, 4 (B′100′) indicates 16K, 5 (B′101′) indicates 32K, 6 (B′110′) indicates 64K, and 7 (B′111′) indicates 128K.
	5– 6	Reserved

Command Parameters (X′80′) Found Control Vector

Byte	Bit	Content
	7	Surrogate resource owner indicator: 0 This Found is from the real owner of the DLU. 1 This Found may be from a surrogate owner of the DLU.

Initiate-Other Cross-Domain (X′12CD′) GDS Variable

The Initiate-Other Cross-Domain GDS variable is sent to request an SLU in the destination node to initiate a session with a different PLU (not the SLU′s current PLU).

Note: For session initiation by a third party, the following values apply: Session Polarity = “DLU is PLU”; LIFO Queuing indicator = “LIFO”; DLU Queuing Conditions = “queue on session limit exceeded.”

Initiate-Other Cross-Domain (X′12CD′) GDS Variable

Byte	Bit	Content
0– 1		Length (p+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12CD′
4 – p		GDS Variable Data
4	0– 2	Session initiation status: 000 initiate 001 queued or proceeding 100 session started 111 procedure terminated
	3– 6	Session Polarity and Queuing Parameters (reserved unless Session Initiation Status = “initiate”)
	3	Session polarity: 0 DLU is PLU. 1 OLU is PLU.
	4– 6	Queuing Parameters (used by the ILU to specify how corresponding bits should be set in the CD-Initiate GDS variable that will be sent from the OLU to the DLU)
	4– 5	DLU Queuing Conditions (specifies certain conditions [at the DLU] for which OLU should be willing to have its forthcoming session initiation request queued — a value of 1 indicates the condition is selected)
	4	Queue on LU not enabled
	5	Queue on session limit exceeded
	6	LIFO queuing indicator — specifies the queuing position for the forthcoming session initiation request, should it become queued: 0 first-in, first-out (FIFO) queuing 1 last-in, first-out (LIFO) queuing
	7	Role of sender: 0 ILU 1 OLU

Initiate-Other Cross-Domain (X′12CD′) GDS Variable

Byte	Bit	Content
5	0– 1	Initiate type (reserved unless Session Initiation Status = “initiate”): 01 initiate only (I) 10 initiate or queue (I/Q)
	2	Authentic mode-name indicator (when set, mode-name translation is bypassed): 0 not authentic 1 authentic
	3	Name change support indicator: 0 new PLU name change not supported 1 new PLU name change supported
	4	LU name authenticity indicator: 0 generic LU name resolution can be performed 1 generic LU name resolution cannot be performed
	5	Net ID authenticity indicator: 0 New PLU net ID is authentic. 1 New PLU net ID is not authentic.
	6– 7	Reserved
6		Length of mode name
7 – n		Mode name: 0 to 8 type-1134 symbol-string characters with optional (but not significant) trailing space (X′40′) characters
n + 1 – p		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” and in “Initiate-Other Cross-Domain Control Vectors” below *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT. X′0E′ Network Name control vector (always present): name of the destination PLU X′34′ LU Definition Override subfield (present if model terminal support override values are available) X′5F′ Extended Fully Qualified PCID control vector (present unless information required from the application program is not available): identifies the session between the current PLU (the ILU) and the SLU (the OLU for the forthcoming session initiation request) X′80′ User Data control vector (present when the initiating application program includes such data in its initiating request; a second User Data control vector is present to carry any excess over 253 user data bytes — in this case, the first User Data control vector carries 253 bytes of user data, while the second carries up to two additional user data bytes)

Initiate-Other Cross-Domain Control Vectors

User Data (X′80′) Initiate-Other Cross-Domain Control Vector

User Data (X′80′) Initiate-Other Cross-Domain Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – p		User data field: user-specific data to be passed in the CINIT User Data field

Locate (X′12C4′) GDS Variable

The Locate (X′12C4′) GDS variable is used in conjunction with other GDS variables by the SEND_ and RECEIVE_NETWORK_SEARCH service transaction programs.

Locate (X′12C4′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12C4′
4		GDS Variable Data
	0	Locate chain indicator: 0 discard 1 keep
	1– 3	Request-reply status: 000 request 010 incomplete reply — a complete reply (bits 1–3 = 100|101|110) will follow 100 complete reply 101 chain flow: Any GDS variables that flow on a previously established Locate chain that has been maintained using the “keep” indicator (bit 0 set to 1), except the first flowing in each direction (which are the request and reply). 110 complete reply, but eligible resources may exist that could not be located because of an outage on the search route
	4	Locate chain keep support (reserved except on a directed Locate request, i.e., except when bits 1–3 = 000, control vector X′80′ is present, and control vector X′2B′ is not present): set to 1 by the NNS(OLU) if it can keep Locate chains after processing their replies; any node (i.e., a back-level implementation) along the directed search path that cannot support keeping Locate chains in this way sets this field to 0. Once set to 0, this field remains 0 for the remainder of the request path. 0 not supported (i.e., byte 4, bit 0 must be set to 0 on the reply) 1 supported (byte 4, bit 0 may be set to 0 or 1 according to need)
	5	Retired
	6	Resubmit on directed Locate search indicator: 0 Resubmission is not required. 1 The request must be resubmitted using a directed Locate search.
	7	Suppress subarea network search indicator: 0 Do not suppress search of subarea network. 1 Suppress search of subarea network.
5– 6		Retired
7– 8		Search number (always sent or received by NNCPs; sent or received by ENCPs implementing options such as Session Services Extensions and Extended Border Node; otherwise, reserved): a binary value used as a secondary key, in conjunction with the Fully Qualified PCID (X′60′) and PCID Modifier (X′81′) control vectors, to uniquely identify a search subprocedure (control block) of a Locate procedure; echoed in the search reply

Locate (X′12C4′) GDS Variable

Byte	Bit	Content
9 – n		Control vectors, as described in "Control Vectors" in Chapter 9, "Common Fields." *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. Control vectors may occur in any order. X′0E′ Network Name control vector: name of the destination control point (present in a request when the destination control point name is known) X′2B′ Route Selection control vector (present on a directed Locate search request exchanged between NNCPs to specify the CPs along a directed Locate procedure path) X′35′ Extended Sense Data control vector (present on a reply to indicate a Locate error) X′60′ Fully Qualified PCID control vector (always present) X′80′ Search Scope control vector (present between NNCPs to define the scope of a broadcast search request) X′81′ PCID Modifier control vector (present on searches initiated at current-level network nodes and at end nodes implementing options such as Session Services Extensions and Extended Border Node) X′82′ Intersubnetwork Search control vector (present when the Locate has been sent over an intersubnetwork TG) X′84′ Cross-Subnetwork Loop Prevention control vector (present to check for looping within a route that crosses multiple subnetworks)

Locate Control Vectors

Search Scope (X′80′) Locate Control Vector

Search Scope (X′80′) Locate Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′80′ (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields")
2 – n		Vector Data
2		Hop count: a binary value specifying the number of hops that may be traversed for a broadcast search (set by the broadcast origin CP and decremented, on the search request, by intermediate CPs participating in the broadcast search)

PCID Modifier (X′81′) Locate Control Vector

PCID Modifier (X′81′) Locate Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′81′ (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields")
2 – n		Vector Data

PCID Modifier (X′81′) Locate Control Vector

Byte	Bit	Content
2– 3		Procedure resubmit number: A 2-byte (always increasing) binary number that indicates the number of times a Locate search request has been resubmitted by any and all nodes
4		Last significant half-byte in the PCID Modifier List field (zero-origin index to the last half-byte in the list that has been claimed by a node on the procedure path)
5 – n		PCID modifier list: up to 20 half-bytes of list entries, with each list entry containing a binary count of the number of new search subprocedures created by a particular node. The Last Significant Half-byte field is used to determine the next available half-byte in the PCID Modifier List. The PCID modifier list has a minimum length of 1 byte, and must always contain an integral number of bytes. Thus, if byte 4 specifies an odd number of half-bytes (i.e., byte 4 is even), then the last byte in the list is padded with X′0′ to form a full byte. A node must be able to receive at least twenty entries in the list.

Intersubnetwork Search (X′82′) Locate Control Vector

The Intersubnetwork Search Locate control vector is present when the Locate has crossed an intersubnetwork TG. It contains search control information used by extended border nodes.

Intersubnetwork Search (X′82′) Locate Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′82′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – n		Vector Data
2		Subnetwork Controls
2	0	Limit search to local subnetwork indicator: 0 Search may span subnetwork boundaries. 1 Search may not exit the local subnetwork. Set by an EBN to limit a broadcast search that is part of a cross-subnetwork search to the local subnetwork.
	1	Limit search to local and adjacent subnetworks indicator: 0 Search is not limited to local and adjacent subnetworks. 1 Search is limited to local and adjacent subnetworks. Set by the EN image of a PBN after receiving or before sending a search across an ISTG; set by an EBN before sending a search across a peripheral ISTG.
	2– 3	Reserved
	4	Prevent subarea path indicator: 0 EBNs on the path of the search may set the subarea network search suppression indicator in the Locate GDS variable to 0. 1 Nodes on the search path may not set the subarea network search suppression indicator to 0. (Set by the DLUS or DLUR to ensure an APPN-only CP-SVR pipe between them.)
	5	DLUR search required indicator: 0 A search to obtain the DLUR TG vectors is not required. 1 A search to obtain the DLUR TG vectors is required.
	6– 7	Reserved

Intersubnetwork Search (X′82′) Locate Control Vector

Byte	Bit	Content
3		Maximum subnetwork-visit count (SNVC): The number, in binary, of subnetworks a search may traverse before being terminated with a complete negative reply. SNVC is 1 greater than the number of ISTG crossings permitted in the remainder of the search.
4		Intersubnetwork-TGs-crossed count (ICC): When nonzero, ICC is the number, in binary, of ISTGs a search has crossed. It is set to 0 at the origin EBN and incremented by 1 at each EBN after receiving a search across an ISTG, and before sending a search across a peripheral ISTG. When 0, the search has either crossed no ISTGs, or has crossed one peripheral ISTG.
5– 6		Reserved

Cross-Subnetwork Loop Prevention (X′84′) Locate Control Vector

The Cross-Subnetwork Loop Prevention control vector identifies the nodes to be used to enter or exit an APPN network (extended border nodes or interchange nodes) in the cross-subnetwork session path through an APPN network. The control vector is carried on the Locate GDS variable during the Locate search request and may be originated, modified, and/or used by border nodes and interchange nodes in the session path.

Cross-Subnetwork Loop Prevention (X′84′) Locate Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′84′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2 – n		Vector Data
2	0	Retired (set to 1)
	1– 7	Reserved
3		The number, in binary, of Network Name control vectors in this Cross-Subnetwork Loop Prevention Locate control vector
4		Retired (always contains the same value as byte 3)
5 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.” *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X′0E′ Network Name (type X′F6′=NNCP) control vectors, one for every entry and exit extended border node and interchange node on the path of the current Locate search procedure. The list may be incomplete if adding another CV X′0E′ would exceed the CV or Locate maximum length.

LU Name (X′1215′) GDS Variable

LU Name (X′1215′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of LU Name GDS variable
2– 3		GDS ID: X′1215′
4		Length of the remainder of the Network-Qualified LU Network Name subfield: values 2 to 18 (X′12′) are valid
5 – n		Network-Qualified LU network name *Note:* The network-qualified LU 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.

LU Names Position (X′1214′) GDS Variable

LU Names Position (X′1214′) GDS Variable

Byte	Bit	Content
0– 1		Length (8), in binary, of LU Names Position GDS variable
2– 3		GDS ID: X′1214′
4– 7		Binary value indicating the (1-origin) position of the receiving LU′s name in the list of LU names that enumerate the LUs involved in this branch of the sync point tree. The list of LU names follow this GDS variable in consecutive LU Name (X′1215′) GDS variables.

Node Address Service (X′1223′) GDS Variable

The Node Address Service (X′1223′) GDS variable is used to request from a service transaction program having the registered TP name X′23F0F0F7′ (referred to as the NODE_ADDRESS_SERVER), running on an APPN node′s control point, a list of one or more non-SNA addresses for the node′s SNMP agent. It is also used as the reply on which the addresses are returned; the reply is returned to the TP identified in the Sender TP Name subfield of the request.

This function is used by a program that knows an APPN node′s control point name, but needs to communicate with the node via some non-SNA protocol such as TCP/IP.

By default, the request GDS variable flows on a session with the SNASVCMG mode name. It can, however, flow on any session supported by an APPN control point, except for those using the CPSVCMG or CPSVRMGR modes. The reply returns on a session using the same mode as the request.

Node Address Service (X′1223′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including this length field
2– 3		GDS ID: X′1223′
4 – n		GDS variable data: one of the following context-dependent GDS structured fields. The first of these structured fields is included in a request for address information. The second structured field is included in the reply. X′FF00′ Node Address Service Side Information X′FF01′ Address List Each of these structured fields is described in zero-origin form below.

Node Address Service Side Information (X′FF00′) GDS Structured Field

Node Address Service Side Information (X′FF00′) GDS Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF00′
4 – n		The following subfield: X′00′ Sender TP Name subfield (always present) The format of this subfield is shown below.

Sender TP Name (X′00′) Subfield

Sender TP Name (X′00′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield
1		Type: X′00′
2 – m		Sender TP name: the SNA-registered or unregistered TP name of the sender of the request for address information. This will be the destination TP of the Node Address Service reply containing the addresses. By default, an application program requesting a node′s list of non-SNA addresses uses as its own name the registered TP name: X′23F0F0F8′ NODE_ADDRESS_REQUESTER. However, any registered or unregistered TP name may be sent in this field. All unregistered TP names are encoded using coded graphic character set 00640-00500.

Address List (X′FF01′) GDS Structured Field

Address List (X′FF01′) GDS Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF01′
4 – n		Zero or more pairs of the following two subfields. In each pair the first subfield indicates the type of address being returned, and the second contains the address itself. An Address List with no entries in it indicates that no SNMP agent exists at the APPN node and therefore no non-SNA address is available for it. Address Type (X′01′) subfield Address Type (X′02′) subfield

Address Type (X′01′) Address List Subfield

This subfield identifies the type of address contained in the X′02′ subfield that immediately follows it.

Address Type (X′01′) Address List Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Address Type subfield
1		Key: X′01′
2(=q)		Address Type: a code point indicating the type of address in the X′02′ subfield that immediately follows this subfield. Defined codes are: X′00′ IP version 4-address X′01′ IP version 6-address

Address (X′02′) Address List Subfield

This subfield transports a single non-SNA address. The encoding and, in some cases, the length of this subfield are determined by the value in the X′01′ subfield that immediately precedes it.

Address (X′02′) Address List Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Address subfield
1		Key: X′02′
2 – q		Address

Notify (X′12CC′) GDS Variable

The Notify GDS variable is sent by one node to another to inform it of a session termination or the enabling of a resource, or to request that a session be released.

Notify (X′12CC′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12CC′
4 – n		GDS Variable Data
4		Notify type: X′01′ CD-Terminate X′02′ resource enabled X′03′ session-release requested X′04′ third-party initiate failure
5– 6		Reserved
7 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” and in “Notify Control Vectors” *Note:* The following control vectors may be included; they are parsed according to sub-field parsing rule LT. X′60′ Extended Fully Qualified PCID control vector (present when CD-Terminate Parameters control vector Terminate Type field = orderly): contains the FQPCID of the session to be terminated (*Note:* When Terminate Type field = “cleanup,” Locate, rather than Notify, carries the Extended FQPCID control vector.) X′80′ CD-Terminate Parameters control vector (present when Notify Type = “CD-Terminate”) X′81′ Third-Party Initiate Failure control vector (present when Notify Type = “third-party initiate failure”)

Notify Control Vectors

CD-Terminate Parameters (X′80′) Notify Control Vector

CD-Terminate Parameters (X′80′) Notify Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′80′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)
2	0	Terminate type: 0 orderly 1 cleanup
	1– 7	Reserved

CD-Terminate Parameters (X′80′) Notify Control Vector

Byte	Bit	Content
3(=p)	0–1	Reasons
	0	0 network user
		1 network manager
	1	0 normal
		1 abnormal
	2	Reason code required indicator, i.e., when this Notify reports a session initiation or termination failure detected by a NAU other than the one that originated it (reserved if bit 1 = "normal"):
		0 reason code not required
		1 reason code required
	3	Failure type:
		0 session initiation failure
		1 session termination failure
	4–7	Reason code copied from CDSESSSF or BINDF (if bit 3 = "session initiation failure") or from CDSESSTF or UNBINDF (if bit 3 = "session termination failure") — a value of 1 indicates the condition is true:
	4	CINIT or CTERM error in reaching PLU
	5	BIND or UNBIND error in reaching SLU
	6	Initiation or termination reject at PLU
	7	Initiation reject at SLU

Third-Party Initiate Failure (X′81′) Notify Control Vector

Third-Party Initiate Failure (X′81′) Notify Control Vector

Byte	Bit	Content
0–1		Vector header; Key=X′81′ (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields")
2 – p		Vector Data *Note:* This vector data corresponds to the Setup/Takedown Procedure Error Reason Code associated with the NS(s) NOTIFY (format 3, third-party SSCP notification, status X′03′).
2(=p)		Reason (A value of 1 in any of the bits 0–3 and 5–7 means the associated condition is true.)
If bit 4 = 0, the Reason byte is encoded for a setup procedure error as follows:		
	0	CINIT error in reaching PLU
	1	BIND error in reaching the SLU
	2	Setup reject at the PLU
	3	Setup reject at the SLU
	4	0 setup procedure error
	5	Reserved
	6	Setup reject at the SSCP
	7	Reserved

Third-Party Initiate Failure (X′81′) Notify Control Vector

Byte	Bit	Content
If bit 4 = 1, the Reason byte is encoded for a takedown procedure error as follows:		
	0	CTERM error in reaching PLU
	1	UNBIND error in reaching the SLU
	2	Takedown reject at the PLU
	3	Takedown reject at the SLU
	4	1 takedown procedure error
	5	Takedown reject at the SSCP
	6	Reserved (see Note below)
	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.

Partner Restart (X′1218′) GDS Variable

The Partner Restart GDS variable indicates the loss of volatile data recording the state of Don′t Know protocols. It is sent at the first contact after restart, following the Exchange Log Name GDS variables. It is only sent if the partner supports Don′t Know processing.

Partner Restart (X′1218′) GDS Variable

Byte	Bit	Content
0– 1		Length (6), in binary, of Partner Restart GDS variable, including this Length field
2– 3		GDS ID: X′1218′
4– 5		Reserved

Register Resource (X′12C3′) GDS Variable

The Register Resource GDS variable is used to request that one or more entries be added to a network node server′s directory of network resources. The network node server returns a Register Resource reply GDS variable to report an error; no reply is sent if the registration is completely successful. A network node may send a Register Resource GDS variable on a directed Locate to a central directory server requesting that one or more entries be added to the central directory server′s directory. A Register Resource GDS variable is included on the Locate reply from the central directory server to report an error.

Register Resource (X′12C3′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12C3′
4 – n		GDS Variable Data

Register Resource (X′12C3′) GDS Variable

Byte	Bit	Content
4 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” and in “Register Control Vectors” on page 13-52. *Note:* The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. X′80′ Command Parameters control vector (always present, always first) **For a Register request**: X′37′ Directory Entry Correlator control vector (optionally present; paired with, and immediately preceding, a X′3C′ or X′3D′ control vector to provide error correlation of the reply data to the corresponding request data; if paired with a control vector that causes an error, the correlator is returned to provide a pointer to the data in error) X′3C′ Associated Resource Entry control vector (optionally present to indicate a hierarchical relationship to the following X′3D′ control vectors, e.g., the ENCP for the LUs to be registered) X′3D′ Directory Entry control vector (present one or more times, not to exceed the length limit of 1024 bytes for the full Register GDS variable) X′3E′ Directory Entry Characteristic control vector (present on a Register from an end node when both it and its network node server support the Nonverify [1108] option set and the end node has the information to send; propagated on a Register to a central directory server if the entry is centrally registered; follows a X′3D′ control vector [or a X′3C′ control vector in the case where LU=CP and the X′3D′ control vector is omitted] to register a characteristic of the resource) X′4A′ Real Owning Control Point control vector: present when the type X′00F4′ (ENCP) Associated Resource Entry control vector in the hierarchy identifies a surrogate for the real owning control point; specifies the true control point that owns the resource identified in the Directory Entry control vector preceding it **For a Register reply**: X′36′ Directory Error control vector (always present) X′37′ Directory Entry Correlator control vector (present when provided in the Register request; returned in a reply to indicate the directory entry on which the error occurred)

Register Control Vectors

Command Parameters (X′80′) Register Control Vector

Command Parameters (X′80′) Register Control Vector

Byte	Bit	Content
0– 1		Vector header; Key = X′80′ (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”)

Command Parameters (X′80′) Register Control Vector

Byte	Bit	Content
2	0	Request/reply indicator: 0 Register request 1 Register (negative) reply
	1	Reserved
	2	Central resource registration request indicator: 0 central resource registration not requested 1 central resource registration requested
	3–4	Retired
	5	DLUS-served LU indicator: 0 These resources are not DLUS-served LUs. 1 These resources are DLUS-served LUs (in which case, byte 2, bit 2 is always set to 0).
	6–7	Reserved

Service Flow Authentication Token Data (X′12F8′) GDS Variable

The Service Flow Authentication Token Data (X′12F8′) GDS variable is used to convey authentication tokens by the Distributed Authentication service TP (X′06F3F0F2′).

Service Flow Authentication Token Data (X′12F8′) GDS Variable

Byte	Bit	Content
0– 1		Length (m+1), in binary, of Service Flow Authentication Token Data GDS variable, including this Length field
2– 3		GDS ID: X′12F8′
4– 1 1		Conversation correlator for the conversation the authentication token is for — padded with X′00′ bytes, if needed
12– 1 5		Sense data
16– 1 7		Length of SNA-specific header
18 – n		SNA-specific header (reserved)
n + 1 – n + 2		Length of the GSS-API authentication token (valid values are 0 to 24,576 — a length of 0 indicates a “null token”)
n + 3 – m		A string of bytes containing the GSS-API authentication token

Sign-Off (X′1220′) GDS Variable

The Sign-Off (X′1220′) GDS variable flows if an active session exists to the partner LU whenever:

- A user′s entry has been removed from the sender′s signed-on-from list.
- A sign-on Attach fails to sign on the user.
- A signed-on Attach fails because the user is not in the sender′s signed-on-from list and was not already verified.

Sign-Off (X′1220′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field.
2– 3		GDS ID: X′1220′
4 – n		Zero to two Sign-Off subfields, each of which has the following format:
0		Length (values 1–11), in binary, of remainder of subfield, not including this byte
1		Subfield type:
		X′00′ profile
		X′01′ reserved

Sign-Off (X′1220′) GDS Variable

Byte	Bit	Content
		X′02′ user ID
2 – i		Profile name or user ID, depending on the subfield type: a 1- to 10-byte symbol string of a type (A, AE, GR, DB, 1134) constrained by the receiver

Notes:

1. Only one of each subfield type is included.
2. If the subfield type is X′00′ (profile) and no profile name follows, only the list entries *lacking* a profile for the specified user ID are to be removed. Omission of the entire profile subfield (length and X′00′ subfield type as well as the profile name) means *all* entries for the specified user ID, *regardless of* profile, are to be removed.
3. Omission of the full user ID subfield (length, X′02′ subfield type, and user ID) implies *all* user IDs in the receiving LU′s signed-on-to list that are signed on to the sending LU are to be removed.

Sign-On (X′1221′) GDS Variable

The Sign-On (X′1221′) GDS variable is used to convey user ID, password, and optional profile to a sign-on server, and to request a sign-on, or a sign-on with change password, or to reply to these requests by the server.

Sign-On (X′1221′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including this length field
2– 3		GDS ID: X′1221′
4 – n		GDS variable data: one of the following context-dependent GDS structured fields: X′FF00′ Sign-On Request Data, password in the clear X′FF01′ Sign-On/Change-Password Request Data, password in the clear X′FF02′ Sign-On Reply Data X′FF03′ Sign-On Request Data, Substituted password X′FF04′ Sign-On/Change-Password Request Data, Substituted Passwords. The X′FF00′, X′FF01′, X′FF03′, and X′FF04′ structured fields flow requester to server; the X′FF02′ structured field flows server to requester. Each of these structured fields is described in zero-origin form below.

Sign-On Request Data, Passwords in the Clear, (X′FF00′) Structured Field

Sign-On Request Data, Passwords in the Clear, (X′FF00′) Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF00′
4 – n		The following subfields (order unspecified): X′00′ Profile subfield (optional) X′01′ User ID subfield (always present) X′02′ Password subfield, in the clear (always present) The formats of these subfields are shown in “Sign-On/Change-Password Common Subfields” on page 13-58.

Sign-On/Change-Clear-Password Request Data, (X′FF01′) Structured Field

Sign-On/Change-Password Request Data, Password in the Clear, (X′FF01′) Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field

Sign-On/Change-Password Request Data, Password in the Clear, (X′FF01′) Structured Field

Byte	Bit	Content
2– 3		GDS ID: X′FF01′
4 – n		The following subfields (order unspecified): X′00′ Profile subfield (optional) X′01′ User ID subfield (always present) X′02′ Password subfield, in the clear (always present) X′06′ New Password subfield, in the clear (always present) The formats of these subfields are shown in “Sign-On/Change-Password Common Subfields” on page 13-58.

Sign-On Request Data, substituted passwords, (X′FF03′) Structured Field

Sign-On Request Data, substituted passwords, (X′FF03′) Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF03′
4 – n		The following subfields (order unspecified): X′00′ Profile subfield (optional) X′01′ User ID subfield (always present) X′03′ Password subfield, substituted password (always present) X′07′ Sequence number (always present) The formats of these subfields are shown in “Sign-On/Change-Password Common Subfields” on page 13-58.

Sign-On/Change-Substituted-Password Request Data, (X′FF04′) Structured Field

Sign-On/Change-Substituted-Password Request Data, (X′FF04′) Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF04′
4 – n		The following subfields (order unspecified): X′00′ Profile subfield (optional) X′01′ User ID subfield (always present) X′03′ Password subfield, substituted password (always present) X′04′ Protected Old Password subfield (always present) X′05′ Protected New Password subfield (always present) X′07′ Sequence number (always present)

Sign-On/Change-Substituted-Password Request Data, (X′FF04′) Structured Field

Byte	Bit	Content
		The formats of these subfields are shown in “Sign-On/Change-Password Common Subfields” on page 13-58.

Sign-On/Change-Password Common Subfields

Profile (X′00′) Subfield

Profile (X′00′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield
1		Type: X′00′
2 – m		Profile: a 1- to 10-byte symbol string of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

User ID (X′01′) Subfield

User ID (X′01′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield
1		Type: X′01′
2 – m		User ID: a 1- to 10-byte symbol string of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

Clear Password (X′02′) Subfield

Clear Password (X′02′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield
1		Type: X′02′
2 – m		Clear Password: a 1- to 10-byte symbol string (in the clear) of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

Substituted Password (X′03′) Subfield

Substituted Password (X′03′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield — always 10 (X′0A′)
1		Type: X′03′
2 – m		Substituted Password: an 8-byte binary value that is calculated the same way that the substituted password is calculated in the Attach

Protected Old Password (X′04′) Subfield

Protected Old Password (X′04′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield — always 10 (X′0A′) or 18 (X′12′)
1		Type: X′04′
2 – m		Protected Old Password: an 8- or 16-byte binary value that is the old password exclusive-ORed with tokens that are calculated the same way that the substituted password is calculated in the Attach

Protected New Password (X′05′) Subfield

Protected New Password (X′05′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield — always 10 (X′0A′) or 18 (X′12′)
1		Type: X′05′
2 – m		Protected New Password: an 8- or 16-byte binary value that is the new password exclusive-ORed with tokens that are calculated the same way that the substituted password is calculated in the Attach

Clear New Password (X′06′) Subfield

Clear New Password (X′06′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield
1		Type: X′06′
2 – m		Clear New Password: a 1- to 10-byte symbol string (in the clear) of a type (A, AE, GR, DB, or 1134) acceptable to the receiver

Sequence Number (X′07′) Subfield

Sequence Number (X′07′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield, always 10 (X′0A′)
1		Type: X′07′
2 – m		Sequence Number: an 8-byte binary value specifying the sequence number to be used in creating the substituted passwords

Sign-On Reply Data (X′FF02′) GDS Structured Field

Sign-On Reply Data (X′FF02′) GDS Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF02′
4 – n		One or more of the following subfields (order unspecified): X′00′ Sign-On Completion Status subfield (always present) X′01′ Sign-On Request Formatting Error subfield (present only when completion status [in the X′00′ subfield] is set to X′06′ [incorrect data format]) X′02′ Date/Time of Current Successful Sign-On subfield (optionally present only when completion status [in the X′00′ subfield] is set to X′00′) X′03′ Date/Time of Last Successful Sign-On subfield (optionally present only when completion status [in the X′00′ subfield] is set to X′00′) X′04′ Date/Time That Password Will Expire subfield (optionally present only when completion status [in the X′00′ subfield] is set to X′00′) X′05′ Number of Unsuccessful Sign-On Requests subfield (optionally present only when completion status′ [in the X′00′ subfield] is set to X′00′) X′06′ Substituted Password Verification Token subfield (optionally present only when completion status [in the X′00′ subfield] is set to X′00′) X′07′ Password Update Failure Reason subfield (optionally present only when completion status [in the X′00′ subfield] is set to X′04′) The formats of these subfields are shown below.

Sign-On Completion Status (X′00′) Subfield

Sign-On Completion Status (X′00′) Subfield

Byte	Bit	Content
0		Length (3), in binary, of this subfield
1		Type: X′00′
2		Completion Status: X′00′ Successful completion: • user ID valid • optional profile valid • password valid • password not expired unless new password specified • new password valid, if specified and therefore set • persistent verification processing complete, if supported X′01′ user ID unknown X′02′ user ID valid, password incorrect X′03′ user ID valid, password correct but expired, requiring new password be sent X′04′ user ID valid, password correct, new password not acceptable to receiving security system, subfield X′07′, if present, provides additional details X′05′ security function failure, function not performed X′06′ incorrect data format, subfield X′01′ provides additional error information X′07′ general security error: user ID unknown or password or optional profile incorrect X′08′ password changed completed, but persistent verification sign-on failed X′09′ user ID valid, password correct and not expired, new password valid if specified, profile invalid X′0A′ sequence number out of range X′0B′ user ID valid, but revoked

Sign-On Request Formatting Error (X′01′) Subfield

Sign-On Request Formatting Error (X′01′) Subfield

Byte	Bit	Content
0		Length (4), in binary, of this subfield
1		Type: X′01′
2– 3		Error code: one of the values also defined for use in bytes 2-3 of sense code X′100B′

Date/Time (X′02′, X′03′, X′04′) Subfields

Date/Time (X′02′, X′03′, X′04′) Subfields

Byte	Bit	Content
0		Length (10), in binary, of this subfield
1		Type: X′02′ date/time of current successful Sign-On X′03′ date/time of last successful Sign-On X′04′ date/time that password will expire
2–9		Date/Time Fields (Values formatted in hex)
2–3		Year (Example: 1989 = X′07C5′)
4		Month (Example: January = X′01′)
5		Day (Example: First day = X′01′; 31st day = X′1F′)
6		Hour (Example: Midnight = X′00′; 23rd hour = X′17′)
7		Minute (Example: On the hour = X′00′; 59th minute = X′3B′)
8		Second (Example: On the minute = X′00′; 59th second = X′3B′)
9		One-hundredth of a second (Example: On the second = X′00′; maximum = X′63′)

Note: On a given day, the maximum time is 23 hours, 59 minutes, and 59.99 seconds. (Midnight is zero hours, zero minutes, and zero seconds on the following day.)

Number of Unsuccessful Sign-On Requests (X′05′) Subfield

Number of Unsuccessful Sign-On Requests (X′05′) Subfield

Byte	Bit	Content
0		Length (4), in binary, of this subfield
1		Type: X′05′
2–3		Number, in binary, of unsuccessful Sign-On requests since the last successful one

Verification Token (X′06′) Subfield

Verification token (X′06′) Subfield

Byte	Bit	Content
0		Length (10), in binary, of this subfield
1		Type: X′06′
2–9		An 8-byte binary verification token

Password Update Failure Reason (X′07′) Subfield

Password Update Failure Reason (X′07′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield (valid values: 3 to 255)
1		Type: X′07′
2		A one-byte field identifying why the new password was not accepted by the security system. Values 128–191 are reserved for implementation-specific reasons. Values 192–255 are reserved for installation-specific reasons. Assigned values are: X′00′ Password was rejected by user installation code. X′01′ New password is longer than maximum accepted length. X′02′ New password is shorter than minimum accepted length. X′03′ New password contains a character used more than once. X′04′ New password has adjacent digits. X′05′ New password contains a character repeated consecutively. X′06′ New password was previously used. X′07′ New password uses an installation disallowed character. X′08′ New password must contain at least one numeric. X′09′ New password must contain at least one alphabetic. X′0A′ New password matches old password in one or more character positions. X′0B′ New password exists in a dictionary of disallowed passwords. X′0C′ New password contains the user ID as part of the password. X′0D′ – X′7F′ Reserved for future assignment. X′80′ – X′BF′ Reserved for implementation-specific reasons. X′C0′ – X′FF′ Reserved for installation-specific reasons.
3– 6		A 4-byte unsigned binary field containing the Coded Character Set Identifier (CCSID) for the message text in the following string field (present only if the Message String field is present)
7 – m		Message string: a type-G symbol string describing the reason the new password was not accepted by the security system

SNMP-over-SNA (X′1222′) GDS Variable

The SNMP-over-SNA (X′1222′) GDS variable is used to transport an SNMP PDU on an LU 6.2 session. In addition to the SNMP PDU, the GDS variable conveys the sender′s transaction program name, so that the receiver can send its responses back to the sender.

SNMP-over-SNA (X′1222′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including this length field
2– 3		GDS ID: X′1222′
4 – n		GDS variable data: both of the following context-dependent GDS structured fields, in this order: X′FF00′ SNMP-over-SNA Side Information X′FF01′ SNMP-over-SNA Protocol Data Unit Each of these structured fields is described in zero-origin form below.

SNMP-over-SNA Side Information (X′FF00′) GDS Structured Field

SNMP-over-SNA Side Information (X′FF00′) GDS Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF00′
4 – n		The following subfield: X′00′ Sender TP Name subfield (always present) The format of this subfield is shown below.

Sender TP Name (X′00′) Subfield

Sender TP Name (X′00′) Subfield

Byte	Bit	Content
0		Length (m+1), in binary, of this subfield
1		Type: X′00′

Sender TP Name (X′00′) Subfield

Byte	Bit	Content
2 – m		Sender TP name: the SNA-registered or unregistered TP name of the sender of the SNMP PDU By default, SNMP-over-SNA uses the following four registered TP names: X′23F0F0F3′ The TP at an SNMP manager that sends an SNMP request to an agent. This will be the destination TP for the response. It is identified in the registry as the “SNMP_PORT_00160_registered” transaction program. X′23F0F0F4′ The agent TP to which a manager sends an SNMP request. This TP corresponds to the well-known UDP port 161. It is identified in the registry as the “SNMP_PORT_00161_registered transaction program.” X′23F0F0F5′ The TP at an SNMP manager to which an agent sends a trap. This TP corresponds to the well-known UDP port 162. It is identified in the registry as the “SNMP_PORT_00162_registered” transaction program. X′23F0F0F6′ The TP at an SNMP agent that sends a trap. This will be the destination TP for the inform confirming receipt of the trap. It is identified in the registry as the “SNMP_PORT_00163_registered” transaction program. In addition to these four registered TP names, a format for unregistered names is recommended for use in situations where the registered TP names are not sufficient. (One such situation is when multiple SNMP managers or agents are installed on the same LU for development, testing, or migration.) These names have the following form: `SNMP_PORT_ppppp[_ccc...ccc]` where “ppppp” represents in EBCDIC a five-digit decimal port number, padded with EBCDIC 0′s (X′F0′ characters) on the left (if necessary), and “ccc...ccc” indicates an optional EBCDIC character string of up to 48 characters. Examples of names with this structure include “SNMP_PORT_01234_testing” and “SNMP_PORT_01234_production.” All unregistered TP names are encoded in EBCDIC using coded graphic character set 00640-00500.

SNMP-over-SNA Protocol Data Unit (X′FF01′) GDS Structured Field

SNMP-over-SNA Protocol Data Unit (X′FF01′) GDS Structured Field

Byte	Bit	Content
0– 1		Length, in binary, of this GDS structured field, including this length field
2– 3		GDS ID: X′FF01′
4 – n		One SNMP protocol data unit, as specified in *SNMP over SNA using APPC*. This document is available on the Internet using the following URL: `ftp://www.raleigh.ibm.com/pub/standards/aiw/snasnmp/snmpsna.txt`

Topology Database Update (X′12C2′) GDS Variable

A Topology Database Update (TDU) GDS variable is used to transport topology data between APPN network nodes. The TDU may also be sent from an end node to its network node server to register or update session endpoint TGs, if both nodes support registration of session endpoint TGs.

Each Topology Database Update GDS variable has a maximum length of 1024 bytes.

Topology Database Update (X′12C2′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1, ≤ 1024), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12C2′
4 – n		GDS Variable Data Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields” and in “Topology Database Update Control Vectors” on page 13-67 *Note:* The following control vectors may be included. They are parsed according to sub-field parsing rule LT (see “Substructure Encoding/Parsing Rules” in Chapter 9, “Common Fields”). X′80′ Flow-Reduction Sequence Number control vector (one Flow-Reduction Sequence Number control vector is present regardless of the number of resources being reported; it always appears first.) X′44′ Node Descriptor control vector (always present): All control vectors that follow a Node Descriptor (X′44′) control vector are assumed to be associated with the node identified in the X′44′ control vector until another X′44′ control vector is encountered. *Note*: Multiple topology updates may be blocked within a single Topology Database Update GDS variable. Each block of updates (those associated with a single node) begins with a Node Descriptor (X′44′) control vector X′45′ Node Characteristics control vector (present when node characteristics are being reported; when present, the Node Characteristics control vector immediately follows the associated Node Descriptor control vector) X′46′ TG Descriptor control vector (present when TG characteristics are being reported) *Note:* The X′46′ and X′47′ control vectors always appear in ordered pairs. X′47′ TG Characteristics control vector (present when TG characteristics are being reported) X′48′ Topology Resource Descriptor control vector (present only when the sender and all its adjacent network nodes support the Garbage Collection Enhancements function set [087], in which case it may follow the X′44′ and X′45′ control vector ordered pair or the X′46′ and X′47′ control vector ordered pair) *Note:* Unrecognized control vectors are stored in the topology database (TDB) and propagated as follows: • Those received between the X′80′ and the X′44′ control vectors (CVs) are simply discarded, not stored or propagated. • Those received between the X′45′ and the X′46′ control vectors are stored in the TDB with the node record corresponding to that X′45′ CV and propagated as received. • Those received following a X′47′ control vector are stored in the TDB with the TG record corresponding to that X′47′ CV and propagated as received.

Topology Database Update Control Vectors

Flow-Reduction Sequence Numbers (X′80′) TDU Control Vector

Successive Flow-Reduction Sequence Number control vectors use monotonically increasing values to identify the ordered sending of Topology Database Update GDS variables. Each update includes flow reduction sequence numbers (FRSNs) in it. This allows a node that has become reconnected to a network to specify (on the CP Capabilities exchange) the last update that it received, or (on TDUs) to specify that gaps exist in the sequence of FRSNs sent. FRSN values begin at 1, are incremented by 1, and wrap to 1 when the end of the range is reached.

Flow-Reduction Sequence Numbers (X′80′) TDU Control Vector

Byte	Bit	Content
0– 1		Vector header; Key=X′80′ (see “Substructure Encoding/Parsing Rules” and “Control Vector Usage” table in Chapter 9, “Common Fields.”)
2 – n		Vector Data
2– 5		Current FRSN of the sender: the binary value that the sender maintains locally for each update that was included in this TDU GDS variable
6– 9 (= n)		Last FRSN sent by the sender to allow the receiver to compute the numbering gap between the previously sent FRSN and the current one

GDS Variables for HPR Control Flows

Route Setup (X′12CE′) GDS Variable

The Route Setup (X′12CE′) GDS variable, may be included in either the data portion of an NLP or in the ROUTE SETUP network control (NC) request RU within a FID2 PIU. The Type field indicates whether the GDS variable is a request or reply. A reply is positive unless an Extended Sense Data (X′35′) control vector is included.

When being built in an NLP, the NHDR and THDR fields are set as follows:

- NHDR:
  - Switching mode is set to *ANR*.
  - Transmission priority is set to *network*.
  - The ANR Routing field contains the NCE identifier for the route setup component in the adjacent node (obtained during the XID3 exchange).
- THDR:
  - The TCID contains the identifier for the Route Setup RTP connection.
  - The rest of the fields are set based on the current state of the RTP connection.
  - The Route Setup request (or reply) may be sent along with the Connection Setup segment for the Route Setup RTP connection.

As is generally true in APPN, intermediate nodes pass on unrecognized control vectors unchanged, while the end points ignore them; a node generating a reply never returns (echoes) unrecognized control vectors it received on a Route Setup request.

Route Setup (X′12CE′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of the GDS variable, including the Length field
2– 3		Key: X′12CE′
4 – n		GDS Variable Data
4	0	Type: 0 request 1 reply (positive or negative)
	1	Path switch indicator (set on a request; reserved on a reply): 0 request not triggered by a path switch 1 request triggered by a path switch
	2– 3	ARB mode - indicates the mode of ARB protocol supported by the Route Setup reply sender. This field is only meaningful on a positive Route Setup reply. 00 Base mode ARB 01 Responsive mode ARB
	4– 7	Reserved

Route Setup (X′12CE′) GDS Variable

Byte	Bit	Content
5		Destination hop index: the index, in binary, of the last TG Descriptor in the Route Selection (X′2B′) control vector describing the TG leading to the destination node; when equal to the current hop count in the RSCV, indicates the receiving node is the intended destination node: • initially set in a Route Setup request by the origin node but modified by intermediate CNNs (if any) when RSCV pruning is done. (Pruning occurs when multiple consecutive VRTG RSCV entries are compressed into one VRTG entry; the hop count fields in the RSCV are also updated so they remain in sync with the destination hop count field.) • set on a positive Route Setup reply to the index of the actual destination node; in this case, the field is not used by intermediate nodes or the origin node—it is set only for network management (e.g., for debugging). **Note:** If the intended destination node does not support the RTP 1401 option set, a negative reply is sent with sense data X′08550020′; this is the “backout” sense data and allows another node to act as the destination if possible.
6–11		Destination Node Information Returned on a Positive Reply—to a request not triggered by a path switch (i.e., that had byte 4, bit 1 = 0); otherwise, reserved
6	0	Directory-search-required-for-path-switch indicator: 0 directory search not required 1 directory search required Notes: • The *destination node* sets this bit to 1 when it is an EN. • The *HPR border node* always sets this bit to 1 in all Route Setup replies passing through it. • If this bit is 1 or the *origin node* is an EN, then it (the origin node) sets the corresponding bit in the Switching Information (X′83′) control vector in the Switching Information segment (sent along with the Connection Setup segment) to 1.
	1	Destination mobility indicator: 0 destination not mobile (i.e., the destination is stationary) 1 destination mobile
	2	NCE scope indicator: indicates whether the NCE included on this reply (in control vector X′26′) is used for all LUs (if the NCE is for an LU) or all BFs (if the NCE is for a BF). If the NCE is used for all LUs (or BFs) in the destination node, the origin node remembers the NCE so that when establishing subsequent RTP connections to other LUs (or BFs) in the destination node, a Route Setup request may not be required to get the NCE. 0 The NCE is not used for all LUs (or BFs) in the destination node. 1 The NCE is used for all LUs (or BFs) in the destination node.
	3	Dedicated RTP connections (i.e., one session per RTP connection) supported indicator: 0 dedicated RTP connections not supported 1 dedicated RTP connections supported
	4–7	Reserved
7		Reserved
8–11		Path switch time: maximum time, in milliseconds, that the destination requires for a path switch; used by the origin node in conjunction with its own path switch time to determine the maximum allowed time for doing a path switch on an RTP connection established with this destination
12–15		Reserved
16 – n		Control vectors, as described in “Control Vectors” in Chapter 9, “Common Fields.”

Route Setup (X′12CE′) GDS Variable

Byte	Bit	Content
		Note: The following control vectors are included as indicated; they are parsed according to subfield parsing rule LT. Only control vectors from the following list are included on a **Route Setup request**: X′80′ Route Information control vector (always present, always last): used to accumulate information about the forward route; the origin node and intermediate nodes add information about the next hop (the next link the Route Setup request is to be sent over). This control vector is always the last one on a Route Setup request so as to minimize data movement by each intermediate node when adding information about the next hop to the end of this control vector. If this control vector is not present, a negative Route Setup reply is sent with sense data X′088C8000′. X′0E′ Network Name (Type X′F3′) control vector (present when the Route Setup request is not for a path switch and the destination node contains the destination LU): contains the network-qualified destination LU name. If the LU specified in this control vector is not known or not ready to accept sessions, the destination sends a negative reply with the appropriate sense data: X′08550002′ or X′08550001′, respectively. X′2B′ Route Selection control vector (always present to direct the flow of the Route Setup request). Its processing along the route is identical to that used to direct the flow of a BIND (i.e., the hop count field in the RSCV is incremented by the origin node and each intermediate node along the path). The RSCV represents the route from the origin node to the destination node with the following exception. When the Route Setup is not for a path switch (byte 4 bit 1 is 0) and the destination LU resides in a node beyond the destination node, then the RSCV includes one additional entry for the destination BF link (the link from the destination node to the node housing the LU). In this case, the destination node will return the NCE identifier associated with the destination BF link on the positive Route Setup reply (in control vector X′26′). If the current hop count in the RSCV exceeds the destination hop index (byte 5) a negative reply is sent with sense data X′0855000E′. If the BF link is identified in the RSCV but is not known or is inactive and not activatable, the destination node sends a negative reply with the appropriate sense data: X′08550004′ or X′08550003′, respectively. If the Route Setup request is not for a path switch and the X′0E′ control vector (with an LU name) and X′2B′ RSCV BF link identifier are both not present, the receiver returns a negative reply with sense data X′088C0E00′. If both are present, the receiver assumes that the LU resides in the destination node and processes it accordingly (no error is indicated). If this control vector is not present, a negative Route Setup reply is sent with sense data X′088C2B00′. X′60′ Fully Qualified PCID control vector (always present to correlate the Route Setup reply to the Route Setup request): always identifies the CP name of the origin node If this control vector is not present, the receiver sends a negative Route Setup reply with sense data X′088C6000′; if this control vector′s length exceeds 28 bytes, the receiver sends a negative Route Setup reply with sense data X′086F600C′.

Route Setup (X′12CE′) GDS Variable

Byte	Bit	Content
		X′2C′ COS/TPF control vector (always present to indicate the COS and TPF for this route): may be used by subnets such as CNN for internal routing (within the subnet); border nodes map the received COS/TPF for the incoming subnet to the COS/TPF for the outgoing subnet (i.e., the COS/TPF control vector on the Route Setup request may be modified by HPR BNs). If the receiver (intermediate or destination node) needs to use this control vector and it is not present, it sends a negative Route Setup reply with sense data X′088C2C00′. X′26′ NCE Identifier control vector (present when the Route Setup request is for a path switch): identifies the destination NCE for a BF or an LU (This control vector may be used in a destination composite network node to identify the particular node that contains the RTP endpoint associated with this RTP connection whose path is being switched.) X′49′ MNPS LU Name control vector (optionally present when the origin LU supports MNPS and the Route Setup request is not for a path switch; contains the MNPS LU name of the origin) Only control vectors from the following list are included on a **positive Route Setup reply**: X′80′ Route Information control vector (always present): contains all information accumulated about the forward route X′80′ Route Information control vector (always present, always last): used to accumulate information about the reverse route; the destination node and the intermediate nodes provide information about the next hop (i.e., the next link that the positive Route Setup reply is to be sent over). This control vector is always the last one on a positive Route Setup reply so as to minimize data movement by each intermediate node when adding information about the next hop to the end of this control vector. X′0E′ Network Name (Type X′F4′) control vector (always present): contains the network-qualified CP name of the destination node. This field is saved by the origin node and used later for subsequent path switches. It is also used to determine if a “backout” has occurred. If the destination CP name is different from the CP name the Route Setup request was originally intended for, then a “backout” has occurred. X′60′ Fully Qualified PCID control vector (always present to correlate the Route Setup reply to the Route Setup request) X′2B′ Route Selection control vector (always present): used to accumulate the TG descriptors of the reverse route. The reverse RSCV sent on an RTP Connection Setup is built on the Route Setup reply path. Each node receiving the reply appends an identifier of the next hop (the hop from the local node to the next node to receive the RS reply) to the RSCV and increments the maximum hop count field in the control vector. (Both the maximum and current hop count fields are initialized to 0 by the destination of the Route Setup request. The current hop count is not incremented and thus is always 0.) If the next hop is through a connection network, both connection network hops—local node to virtual routing node (VRN), and VRN to real adjacent node—are appended by the local node. (Some of the information needed to create the “next hop TG” is remembered from the forward hop describing the same link of the RSCV received on the Route Setup request; the *HPR Architecture Reference* describes further how each field is set in the “next-hop TG” Descriptor Entry (X′46′) control vector that is built and appended to the accumulated reverse RSCV.)

Route Setup (X′12CE′) GDS Variable

Byte	Bit	Content
		X′2C′ COS/TPF control vector (always present to indicate the COS and TPF for this route): The destination always puts the COS/TPF control vector received on the Route Setup request on the Route Setup reply. The origin node always uses this COS/TPF, when present on a received positive reply, in the Topic Identifier field of the RTP Connection Setup segment. If COS/TPF is not present on a positive reply, the origin uses the COS/TPF that was sent on the Route Setup request. The COS/TPF received on the Route Setup request is always understood by the destination, even when the Route Setup has passed through multiple subnets; this is because HPR border nodes translate it at each subnet boundary. This COS/TPF is communicated to the origin and is used on the connection setup so that the destination node can understand the COS/TPF associated with the RTP connection.
		X′26′ NCE Identifier control vector (present when the Route Setup reply is not for a path switch): identifies the NCE of either the destination LU or the destination BF link as follows: • When the destination LU resides in the destination node, this control vector identifies the NCE associated with the destination LU. The destination LU name was specified in the control vector X′0E′ on the Route Setup request. • When the destination LU resides beyond the destination node, this control vector identifies the NCE associated with the destination BF link. The RSCV on the Route Setup request specified the destination BF link. The (LU) Network Name (X′0E′) control vector was not included.
		X′39′ NCE Instance Identifier control vector (present when the Route Setup reply is not for a path switch): identifies the instance of the NCE component identified in the preceding X′26′ control vector
		X′49′ MNPS LU Name control vector (optionally present when the destination LU supports MNPS and the Route Setup reply is not for a path switch; contains the MNPS LU name of the destination)
		Control vectors on a **negative Route Setup reply:** X′35′ Extended Sense Data control vector (always present): indicates the error that has occurred and the node that detected the error: RU Information is not included (i.e., byte 6 bit 0 in the control vector is always set to 0); also, the Related Resource field may optionally be present to carry additional product-specific information to aid in problem determimation. X′60′ Fully Qualified PCID control vector (always present): used to correlate the Route Setup reply to the Route Setup request **Note:** It is recommended, but not required, that any control vectors available from the Route Setup request that is being rejected be echoed in the negative reply. It is also recommended that the X′35′ control vector appear before any other control vectors.

Route Information (X′80′) Control Vector

The Route Information (X′80′) control vector contains information about either the forward or reverse route (path).

Route Information (X′80′) Control Vector

Byte	Bit	Content
0–1		Vector header; Key = X′80′ (see "Substructure Encoding/Parsing Rules" in Chapter 9, "Common Fields")
2	0	Route direction: 0 Forward—information about the forward route is collected in this control vector. 1 Reverse—information about the reverse route is collected in this control vector.
	1	Resequence ("REFIFO") indicator: indicates whether or not RTP connection traffic can, as part of normal operation (i.e., with no errors occurring), get out of order in this direction on the path (i.e., NLPs may not arrive at the receiver in the order that they were sent) and require resequencing. If an RTP end point expects traffic may be received out of order, it will allow sufficient time to receive missing (i.e., delayed) packets before asking the sender RTP end point to resend them. 0 The RTP endpoint receiving this control vector on a request or a positive reply need not do resequencing. 1 The RTP endpoint receiving this control vector on a request or a positive reply must do resequencing.
	2–7	Reserved
3		Reserved
4–7		Maximum packet size: used to establish the smallest link packet size in this direction along the RTP connection. When a node receives a Route Setup request or positive reply, it checks to see if the maximum packet size for the next hop (the TG that the Route Setup request or reply will be sent out on) is less than the current value in the received Route Setup. If it is, the value of the maximum packet size for the next hop is stored in this field. Thus, the maximum packet size for the next hop is the minimum of this node′s maximum send packet size (locally defined) and the adjacent node′s maximum receive packet size (obtained from the partner in the "Maximum BTU length that the XID sender can receive" field of the XID3). The final value in this field is the maximum packet size that can be sent by the RTP end point over this route. The purpose of this field is to avoid segmenting in the intermediate nodes.
8–11		Accumulated transmission time (in microseconds for 1200 bits): The relevant transmission time for the next hop is added to the current value in the received Route Setup. The final value indicates the total transmission time for the entire path and is used for calculating the sensitivity threshold for ARB. The next-hop transmission time is derived from the effective capacity value specified for that TG in the topology database, which is encoded there as a floating-point number (see the *SNA APPN Architecture Reference* TRS chapter for details). The following is an example of how this time is computed: Assume the effective capacity value is encoded in the database as X′45′ (i.e., as the binary floating-point representation 01000101, where the exponent is 01000 and the 3-bit normalized mantissa is 101). Converting the floating-point value to decimal yields 208 units. The effective capacity converted to bits-per-second then equals 300 times 208 (since each unit is 300 bps), or 62400 bps. The transmission time in microseconds of 300 bits is 1,000,000 microseconds (=1 sec) divided by 208, giving 4807 microseconds. Multiplying this by 4 (since 1200 bits is 4 times 300 bits)—or shifting left by 2—gives the final result of 19230 microseconds, which is added to this field, bytes 8–11.
12–15		Minimum link capacity (in Kbits per second): If the link capacity for the next hop is less than the value in the received Route Setup, this lesser value is stored in this field. The final value indicates what the capacity is for the slowest link along the path and is used to calculate the initial send rate for ARB.

Route Information (X′80′) Control Vector

Byte	Bit	Content
16–19		Limited-resource liveness timer value (in seconds): Each limited-resource link along the path has a liveness timer value associated with it. The purpose of this field is to obtain the smallest liveness timer value of all the limited-resource links along the path. The field is initialized to 0. When a Route Setup is received and the next hop is a limited-resource link, a check is made to see if the locally defined liveness timer value is less than the value currently in this field. If it is, the locally defined value is stored in this field. The final value of this field will be either 0 (indicating that no limited-resource links exist along the path) or positive (indicating that at least one limited-resource link exists along the path). If positive, it will be used to set the RTP connection ALIVE timer value, which is described further in the *HPR Architecture Reference*.
20 – n		ANR Path (X′67′) control vector: This control vector contains the accumulated ANR information for the path. The control vector is updated at each hop by appending the ANR label entry for the next hop. NCE identifiers are never appended.

GDS Variables for Management Services

MDS Message Unit (X′1310′) GDS Variable

Multiple-Domain Support Message Unit (MDS-MU) transports routing and control information and data for management services application programs.

Multiple-Domain Support Message Unit (X′1310′) GDS Variable

Byte	Bit	Content
0– 1		Length (m+1), in binary, of the MDS-MU
2– 3		GDS ID: X′1310′
4 – m		The following general data stream (GDS) variables as indicated:

General Data Stream (GDS) Variables	Presence in MDS-MU (X′1310′) GDS variable	
MDS Routing Information (X′1311′)	P	Note 1
Agent Unit of Work Correlator (X′1549′)	P	Note 2
SNA Condition Report (X′1532′)	CP	Note 3
CP-MSU (X′1212′)	CP	Note 4
MDS-MU (X′1310′)	CP	Note 5

Key:

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

Notes:

1. MDS Routing Information is always the first structure in the MDS-MU.
2. Agent Unit of Work Correlator is always the second structure in the MDS-MU. See Appendix B, “Common Structures” for the format of this GDS variable.
3. SNA Condition Report is always present if the MDS message type is X′02′ (MDS error message), as indicated in byte 2 of the Flags (X′90′) MDS Routing Information subvector. It is optionally present for other MDS message types (see next Note). See Appendix B, “Common Structures” for the format of this GDS variable.
4. CP-MSU is optionally present for MDS message types X′00′ (MDS request) and X′01′ (MDS reply). It may not be present for message type X′02′ (MDS error message).

   For MDS requests and replies, a single GDS variable may be included after the Agent Unit of Work Correlator. This GDS variable, which is supplied by the origin MS application program, may be one of the following:

a. CP-MSU
b. SNA Condition Report
c. Some other GDS variable, not currently defined by management services architecture.

Multiple-domain support considers this GDS variable to be application data, with no restrictions except the following:

a. At most, one application GDS variable may be present.
b. The length of the application GDS variable may not exceed 31743 (X′7BFF′) bytes.

5. Another MDS-MU is included within the MDS-MU only under the following conditions:

a. The MDS message type is X′02′ (MDS error message)
b. The Application ID (X′03′) SF of both the Origin Location Name (X′81′) SV and the Destination Location Name (X′82′) SV contains the value X′23F0F1F0′ (MDS_ROUTER).
c. The SNA Condition Report is present, and the SNA Report Code (X′7D′) SV contains the value X′08A8 0009′ (Destination not supported by reporting network node).

The presence of this GDS variable is explicitly prohibited in all other circumstances.

MDS Routing Information (X′1311′) GDS Variable

Multiple-Domain Support (MDS) Routing Information contains routing and control information for the Multiple-Domain Support Message Unit (MDS-MU) containing it.

MDS Routing Information (X′1311′) GDS Variable

Byte	Bit	Content
0– 1		Length (m+1), in binary, of the MDS Routing Information GDS variable
2– 3		GDS ID: X′1311′
4 – m		The following MDS subvectors: X′81′ Origin Location Name (always first) X′82′ Destination Location Name (always second) X′90′ Flags (always third)

Origin Location Name (X′81′) MDS Routing Information Subvector

This subvector identifies the origin NAU and application program for the Multiple-Domain Support Message Unit (MDS-MU) that contains it.

Origin Location Name (X′81′) MDS Routing Information Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the Origin Location Name subvector
1		Key: X′81′
2 – p		Three subfields containing data identifying the origin of the record, as described below. X′01′ NETID (always first) X′02′ NAU Name (always second) X′03′ Application ID (always third)

NETID (X′01′) Origin Location Name Subfield

This subfield contains the network identifier portion of the network-qualified name of the NAU that originated the management services record containing it.

NETID (X′01′) Origin Location Name Subfield

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

NETID (X′01′) Origin Location Name Subfield

Byte	Bit	Content
1		Key: X′01′
2 – q		NETID: a 1- to 8-byte type-1134 symbol string name; trailing space (X′40′) characters may be present, but are insignificant; leading or embedded space characters are not permitted.

NAU Name (X′02′) Origin Location Name Subfield

This subfield contains the unqualified name of the NAU that originated the management services record containing it. This is a CP or LU name.

NAU Name (X′02′) Origin Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the NAU Name subfield
1		Key: X′02′
2 – q		NAU name: a 1- to 8-byte type-1134 symbol string name; trailing space (X′40′) characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Application ID (X′03′) Origin Location Name Subfield

This subfield contains either a 4-byte application program name defined by the management services architecture or a 1- to 8-byte installation-defined name.

Application ID (X′03′) Origin Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Application Identification subfield
1		Key: X′03′
2 – q		Application identification: Either a 1- to 8-byte type-1134 symbol string name, or one of the 4-byte architecturally defined values for management services application programs, listed in *SNA/Management Services Reference*. Trailing space (X′40′) characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Destination Location Name (X′82′) MDS Routing Information Subvector

This subvector identifies the destination NAU and application program for the Multiple-Domain Support Message Unit (MDS-MU) that contains it.

Destination Location Name (X′82′) MDS Routing Information Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the Destination Location Name subvector
1		Key: X′82′
2 – p		Three subfields containing data identifying the destination of the record, as described below. X′01′ NETID (always first) X′02′ NAU Name (always second) X′03′ Application ID (always third)

NETID (X′01′) Destination Location Name Subfield

This subfield contains the network identifier portion of the network-qualified name of the NAU to which the management services record containing it is being sent.

NETID (X′01′) Destination Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the NETID subfield
1		Key: X′01′
2 – q		NETID: a 1- to 8-byte type-1134 symbol string name; trailing space (X′40′) characters may be present, but are insignificant; leading or embedded space characters are not permitted.

NAU Name (X′02′) Destination Location Name Subfield

This subfield contains the unqualified name of the NAU to which the management services record containing it is being sent. This is a CP or LU name.

NAU Name (X′02′) Destination Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the NAU Name subfield
1		Key: X′02′

NAU Name (X′02′) Destination Location Name Subfield

Byte	Bit	Content
2 – q		NAU name: a 1- to 8-byte type-1134 symbol string name; trailing space (X′40′) characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Application ID (X′03′) Destination Location Name Subfield

This subfield contains either a 4-byte application program name defined by the management services architecture or a 1- to 8-byte installation-defined name.

Application ID (X′03′) Destination Location Name Subfield

Byte	Bit	Content
0		Length (q+1), in binary, of the Application Identification subfield
1		Key: X′03′
2 – q		Application identification: Either a 1- to 8-byte type-1134 symbol string name, or one of the 4-byte architecturally defined values for management services application programs, listed in *SNA/Management Services Reference*. Trailing space (X′40′) characters may be present, but are insignificant; leading or embedded space characters are not permitted.

Flags (X′90′) MDS Routing Information Subvector

This subvector contains various flags related to the transport of data between management services application programs.

Flags (X′90′) MDS Routing Information Subvector

Byte	Bit	Content
0		Length (p+1), in binary, of the Flags subvector
1		Key: X′90′
2		MDS message type: X′00′ MDS request X′01′ MDS reply X′02′ MDS error message

Flags (X′90′) MDS Routing Information Subvector

Byte	Bit	Content
3– 4 (= p)		Flags:
	0	First MDS message indicator: 0 MDS message is not the first message for the current unit of work 1 MDS message is the first message for the current unit of work. This value is required for an MDS error message. If the last MDS message indicator is also 1, then the message is the only one for the current unit of work.
	1	Last MDS message indicator: 0 MDS message is not the last message for the current unit of work 1 MDS message is the last (or only) message for the current unit of work. This value is required for an MDS error message.
	2	Routing verification indicator: 0 MDS-MU was routed based on unverified directory information 1 MDS-MU was routed based on verified directory information
	3– 4	Application-level UOW context
	5– 1 5	Reserved

GDS Variables for LU 6.2 Application Programs

Figure 13-3 indicates (using an “X”) every GDS variable code point with first byte = X′12′.

The following subsections describe the details of GDS variables that are specific to LU 6.2 application transaction programs.

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

Figure 13-3. LU Type 6.2 GDS Variable Code Points

The code points used by LU 6.2, the APPN CP, and by HPR are:

X′1210′	Change Number of Sessions (see Note 1)
X′1211′	Exchange Log Name (see Note 1)
X′1212′	Control Point—Management Services Unit (see Note 1)
X′1213′	Compare States (see Note 1)
X′1214′	LU Names Position (see Note 1)
X′1215′	LU Name (see Note 1)

X′1217′	Do Know (see Note 1)
X′1218′	Partner Restart (see Note 1)
X′1219′	Don′t Know (see Note 1)
X′1220′	Sign Off (see Note 1)
X′1221′	Sign On (see Note 1)
X′1222′	SNMP over SNA (see Note 1)
X′1223′	Node Address Service (see Note 1)
X′12A0′	Workstation Display Passthrough (set aside for AS/400 use and will not otherwise be defined by SNA)
X′12A1′	Shared Object Coupling / Communications Transport (SOC/CT) (set aside for AS/400 use and will not otherwise be defined by SNA)
X′12C1′	CP Capabilities (see Note 1)
X′12C2′	Topology Database Update (see Note 1)
X′12C3′	Register Resource (see Note 1)
X′12C4′	Locate (see Note 1)
X′12C5′	Cross-Domain Initiate (see Note 1)
X′12C9′	Delete Resource (see Note 1)
X′12CA′	Find Resource (see Note 1)
X′12CB′	Found Resource (see Note 1)
X′12CC′	Notify (see Note 1)
X′12CD′	Initiate-Other Cross-Domain (see Note 1)
X′12CE′	Route Setup (see Note 2)
X′12DF′	Compressed Application Data (retired)
X′12E1′	Error Log (see below)
X′12E2′	PIP Structured Field (see Note 3)
X′12F1′	Null Data (see below)
X′12F2′	User Control Data (see below)
X′12F3′	Map Name (see below)
X′12F4′	Error Data (see below)
X′12F5′	PIP (see Note 3)
X′12F6′	Authentication Token Data (see below)
X′12F8′	Service Flow Authentication Token Data (see Note 1)
X′12FF′	Application Data (see below)

Notes:

1. See “GDS Variables for SNA Service Transaction Programs (STPs)” on page 13-9 for the formats and meanings of these GDS variables.
2. See “GDS Variables for HPR Control Flows” on page 13-69 for the format and meaning of this GDS variable.
3. See Chapter 11, “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.

Authentication Token Data (X′12F6′) GDS Variable

The Authentication Token Data GDS variable, ID X′12F6′, is used to convey authentication tokens on a conversation before user data. When associated with

the Attach (LU 6.2) FMH5, it follows the PIP (X′12F5′) GDS variable (if present). Its format is:

Authentication Token Data (X′12F6′) GDS Variable

Byte	Bit	Content
0– 1		Length (p+1), in binary, of the GDS variable, including this Length field
2– 3		GDS ID: X′12F6′
4– 5		Length of SNA-specific header (valid values are 0 to n-5)
6 – n		SNA-Specific Header
6		Header Byte
	0	Start of deferred token exchanges: 0 Token exchanges are to continue using the conversation′s session. 1 Additional token exchanges for this conversation are to be performed using the Distributed Authentication service TP. In this case, the associated FMH5 must contain a valid conversation correlator.
	1	Mutual authentication requested: 0 Mutual authentication was not requested on the originating system. 1 Mutual authentication was requested (set only in initial flow from the Attach sender; otherwise reserved).
	2– 7	Reserved
7		Length of Security mechanism′s object identifier (valid values are 0 to 32)
8 – m		BER-encoded form of the Security mechanism′s object identifier (required only in initial flow from the Attach sender; if omitted, the associated length field is set to 0)
m + 1 – n		Rest of SNA-specific header (reserved)
n + 1 – n + 2		Length of the GSS-API authentication token (valid values are 0 to 24,576; a length of 0 indicates a “null token”)
n + 3 – p		A string of bytes containing the GSS-API authentication token

Error Data (X′12F4′) GDS Variable

The Error Data GDS variable, ID X′12F4′, is used to convey information about mapping errors. It is sent using the SEND_DATA verb following a SEND_ERROR verb. Its format is:

Error Data (X′12F4′) GDS Variable

Byte	Bit	Content
0– 1		Length (n+1), in binary, of Error Data GDS variable, including this Length field
2– 3		GDS ID: X′12F4′
4– 7		Error code: X′00010000′ Invalid GDS ID: The mapped conversation verb component encountered a GDS ID that it did not recognize. X′00030001′ Map Not Found: The specified map was not available at the target, or access to the referenced map could not be completed. X′00030002′ Map Execution Failure: The map program was not able to process the data stream.
8		Length (n-8), in binary, of error parameter

Error Data (X′12F4′) GDS Variable

Byte	Bit	Content
9 – n		Error parameter: for a mapping failure, the map name carried in the GDS variable for which the error occurred; for an invalid GDS ID, the 2-byte GDS ID that was not recognized

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 9, “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

Map Name (X′12F3′) GDS Variable

The Map Name GDS variable, ID X′12F3′, is followed by a 0- to 64-byte map name.

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.

End of Chapter 13

Chapter 14. SNA/DS FS1 Encodings

Introduction

This chapter contains the format descriptions of the FS1 message units. The format descriptions are comprised of two parts: *header description tables* and *structure descriptions*. A header description table contains the header information for each structure associated with a particular message unit. A structure description contains a prose description of the structure, bit-level representations, any presence rules or length restrictions associated with a particular structure, and any special notes required to understand the differences between FS1 and FS2 encodings.

The definition of SNA/Distribution Services (SNA/DS) requires a byte-accurate description of the formats that must be understood by all DSUs. The SNA/DS formats are described in terms of encoded fields referred to as ″structures″ and the hierarchical relationship between these structures. In this chapter, the header description tables show each structure and its header.

Refer to Appendix B, “Common Structures” for a complete definition and classification of the encoding structures used in the following tables.

Header Description Tables for FS1 Message Units

DISTRIBUTION MESSAGE UNIT (DIST_MU)

Figure 14-1 (Page 1 of 2). Distribution Message Unit (DIST_MU)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_MU	14-8	Del-IDF	pfx	≥148	1	N	Y	3-4	—
Prefix	14-8	IDF/pfx	C00102	5-21	1	—	—	—	—
Dist_Command	14-8	IDF/seg	C10502	138-32511	1	N	Y	2-3	—
Service_Desc_Operands	14-8	Imp-IDF	idc	58-774	1	N	N	2-5	—
Dist_ID	14-8	IDF/idc	C34041	28-107	1	N	N	5-7	—
Origin_RGN	14-8	T	01	3-10	0-1	—	—	—	—
Origin_REN	14-9	T	02	3-10	1	—	—	—	—
Origin_DGN	14-9	T	03	2-10	1	—	—	—	—
Origin_DEN	14-9	T	04	2-10	1	—	—	—	—
Origin_Seqno	14-10	T	05	6	1	—	—	—	—
Origin_DTM	14-10	T	06	10	1	—	—	—	—
Agent_Correl	14-10	T	07	3-46	0-1	—	—	—	—
Dist_Gen_Options	14-11	IDF	C33D41	30-58	1	N	N	5	—
Dist_Flags (FS1)	14-11	T	01	3	1	—	—	—	—
Hop_Count	14-11	T	02	4	1	—	—	—	—
Service_Parms	14-12	T	03	11-32	1	—	—	—	—
Server_Object_Ind	14-15	T	04	4	1	—	—	—	—
Origin_Agent	14-15	T	05	3-10	1	—	—	—	—
Report-To_Address	14-15	IDF	C36041	14-45	0-1*	N	N	3-4	—
Report-To_RGN	14-15	T	01	3-10	0-1	—	—	—	—
Report-To_REN	14-16	T	02	3-10	1	—	—	—	—
Report-To_DGN	14-16	T	03	3-10	1	—	—	—	—
Report-To_DEN	14-17	T	04	3-10	1	—	—	—	—
Report-To_Options	14-17	IDF	C34341	8-47	0-1*	N	N	1-2	—
Report_Service_Parms	14-18	T	01	11-32	0-1	—	—	—	—
Report-To_Agent	14-21	T	02	3-10	0-1	—	—	—	—
Agent_Object	14-21	IDF	C32D01	6-517	0-1	—	—	—	—
Destination_Operands	14-21	Imp-IDF	idc	≥75	1	N	Y	3	—
Begin_Dest_Operands	14-23	IDF/idc	C35001	8	1	—	—	—	—
Dest_RGN_List	14-23	Imp-IDF	idc	≥62	≥1	N	Y	4	—
Dest_RGN	14-23	IDF/idc	C35201	5-13	1	—	—	—	—
Begin_REN_List	14-23	IDF	C35001	8	1	—	—	—	—
Dest_REN_List	14-23	Imp-IDF	idc	≥44	≥1	N	Y	4	—
Dest_REN	* *	IDF/idc	C35301	6-13	1	—	—	—	—
Begin_DGN_List	14-24	IDF	C35001	8	1	—	—	—	—
Dest_DGN_List	14-24	Del-IDF	pfx	≥25	≥1	N	Y	4	—

Figure 14-1 (Page 2 of 2). Distribution Message Unit (DIST_MU)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dest_DGN	* *	IDF/pfx	C35401	6-13	1	—	—	—	—
Begin_DEN_List	14-24	IDF	C35001	8	1	—	—	—	—
Dest_DEN	14-25	IDF	C35501	6-13	≥1	—	—	—	—
End_DEN_List	14-25	IDF/sfx	C35101	5	1	—	—	—	—
End_DGN_List	14-25	IDF	C35101	5	1	—	—	—	—
End_REN_List	14-25	IDF	C35101	5	1	—	—	—	—
End_Dest_Operands	14-25	IDF	C35101	5	1	—	—	—	—
Dist_Report_Operands	14-27	Imp-IDF	idc	≥63	0-1*	N	Y	2-4	14-6
Dist_Server_Operands	14-25	Imp-IDF	idc	≥14	0-1*	N	Y	2	—
Server_Prefix	14-25	IDF/idc	C90A41	8-280	1	N	N	1-3	—
Server_Obj_Byte_Count	14-26	T	01	10	0-1	—	—	—	—
Server	14-26	T	02	3-10	1	—	—	—	—
Server_Parms	14-26	T	03	3-255	0-1	—	—	—	—
Server_Object	14-26	IDF/seg	C90801	≥6*	1	—	—	—	—
DS_Suffix (FS1)	14-27	IDF	CF0100	5	1	—	—	—	—

Notes:

- * Refer to FS1 Structure Descriptions starting on page 14-8 for presence rules and length restrictions.
- * * Refer to Chapter 15, "SNA/DS FS2 Encodings."
- *Dist_Report_Operands* does not occur for Dist_MU *type* TRANSPORT.
- *Agent_Correl*, *Report-To_Address*, *Report-To_Options*, *Agent_Object*, and *Dist_Server_Operands* do not occur for Dist_MU *type* REPORT.
- *Dest_RGN_List*, *Dest_REN_List*, *Dest_DGN_List*, and *Dest_DEN* occur only one time for Dist_MU *type* REPORT.

DIST REPORT OPERANDS

Figure 14-2. Distribution Report Operands

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Report_Operands	14-27	Imp-IDF	idc	≥63	0-1	N	Y	2-4	—
Report_Operands	14-27	Imp-IDF	idc	27-112	1	N	N	1-2	—
Report_Correlation	14-27	IDF/idc	C34041	27-87	1	N	N	4-5	—
Reported-On_Origin_DGN	14-27	T	03	3-10	1	—	—	—	—
Reported-On_Origin_DEN	14-28	T	04	3-10	1	—	—	—	—
Reported-On_Seqno	14-28	T	05	6	1	—	—	—	—
Reported-On_DTM	14-29	T	06	10	1	—	—	—	—
Reported-On_Agent_Correl	14-29	T	07	3-46	0-1	—	—	—	—
Receiving_DSU	14-29	IDF	C36141	8-25	0-1	N	N	1-2	—
Receiving_RGN	14-30	T	01	3-10	0-1	—	—	—	—
Receiving_REN	14-30	T	02	3-10	1	—	—	—	—
Gen_SNADS_Report	14-30	Imp-IDF	idc	16	0-1*	N	Y	2	—
Gen_SNADS_Type	14-31	IDF/idc	C35601	7	1	—	—	—	—
Gen_SNADS_Contents	14-31	IDF	C35741	9	1	N	Y	1	—
Gen_SNADS_Cond_Code	14-31	T	01	4	1	—	—	—	—
Gen_DIA_Report	14-32	Imp-IDF	idc	14-524	0-1*	N	Y	2	—
Gen_DIA_Type	14-32	IDF/idc	C35601	7	1	—	—	—	—
Gen_DIA_Contents	14-32	IDF	C35741	7-517*	1	—	—	—	—
Specific_Report	14-32	Imp-IDF	idc	≥36	1	N	Y	3	—
Begin_Report_DGN_List	14-32	IDF/idc	C35001	8	1	—	—	—	—
Report_DGN_List	14-33	Imp-IDF	idc	≥23	≥1	N	Y	4	—
Reported-On_Dest_DGN	14-33	IDF/idc	C35401	5-13	1	—	—	—	—
Begin_Report_DEN_List	14-33	IDF	C35001	8	1	—	—	—	—
Report_DEN_List	14-33	Imp-IDF	idc	5-553	≥1	N	Y	1-3	—
Reported-On_Dest_DEN	14-34	IDF/idc	C35501	5-13	1	—	—	—	—
Spec_SNADS_Report	14-34	Imp-IDF	idc	16	0-1*	N	Y	2	—
Spec_SNADS_Type	14-34	IDF/idc	C35601	7	1	—	—	—	—
Spec_SNADS_Cont	14-34	IDF	C35741	9	1	N	Y	1	—
Spec_SNADS_CC	14-35	T	01	4	1	—	—	—	—
Spec_DIA_Report	14-35	Imp-IDF	idc	14-524	0-1*	N	Y	2	—
Spec_DIA_Type	14-36	IDF/idc	C35601	7	1	—	—	—	—
Spec_DIA_Contents	14-36	IDF	C35741	7-517*	1	—	—	—	—
End_Report_DEN_List	14-36	IDF	C35101	5	1	—	—	—	—
End_Report_DGN_List	14-36	IDF	C35101	5	1	—	—	—	—

Note: * Refer to FS1 Structure Descriptions starting on page 14-8 for presence rules and length restrictions.

SENDER EXCEPTION MESSAGE UNIT (TYPE FS1)

Figure 14-3. Sender Exception Message Unit (type FS1)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Sender_Exception_MU (FS1)	14-37	IDF	CF0201	8	1	—	—	—	—

RECEIVER EXCEPTION MESSAGE UNIT (TYPE FS1)

Figure 14-4. Receiver Exception Message Unit (type FS1)

Structure Name	Struct Ref Pg	Struct Class	IDF/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Receiver_Exception_MU (FS1)	14-37	Del-IDF	pfx	59-863	1	N	Y	3	—
Prefix	14-8	IDF/pfx	C00102	5	1	—	—	—	—
Receiver_Exception_Command	14-37	IDF	C10101	49-853	1	N	Y	2	—
Receiver_Exception_Correl	14-38	IDF	C32801	7-23	1	—	—	—	—
Exception_And_Reply_Data	14-38	Imp-IDF	idc	37-825	1	N	N	2	—
Receiver_Exception_Code	14-39	IDF/idc	C32201	8-255	1	—	—	—	—
Reply_Data	14-40	IDF	C34501	29-570	1	N	Y	2-3	—
Receiving_DSU	14-29	IDF	C36141	8-25	1	N	N	1-2	—
Receiving_RGN	14-30	T	01	3-10	0-1	—	—	—	—
Receiving_REN	14-30	T	02	3-10	1	—	—	—	—
SNADS_Report	14-40	Imp-IDF	idc	16	1	N	Y	2	—
SNADS_Report_Type	14-40	IDF/idc	C35601	7	1	—	—	—	—
SNADS_Report_Cont	14-40	IDF	C35741	9	1	N	Y	1	—
SNADS_Report_CC	14-41	T	01	4	1	—	—	—	—
DIA_Report	14-41	Imp-IDF	idc	14-524	0-1	N	Y	2	—
DIA_Report_Type	14-41	IDF/idc	C35601	7	1	—	—	—	—
DIA_Report_Cont	14-42	IDF	C35741	7-517	1	—	—	—	—
DS_Suffix (FS1)	14-27	IDF/sfx	CF0100	5	1	—	—	—	—

FS1 Structure Descriptions

Dist_MU

Description: The *distribution_message_unit* transports user information to one or more distribution service users. A Dist_MU can be one of two types based on the value of *dist_flags* (type FS1): TRANSPORT or REPORT. A Dist_MU *type* TRANSPORT transports agent and/or server objects. A Dist_MU *type* REPORT transports information reporting on the state of the distribution.

Prefix

Description: The *prefix* identifies the beginning of a message unit and may contain a message-unit identifier.

Format: Undefined byte string

Dist_Command

Description: The *distribution_command* contains all information used by each DSU to transport the distribution for a Dist_MU *type* TRANSPORT. For a Dist_MU *type* REPORT, the *distribution_command* contains the control information for the distribution report.

Service_Desc_Operands

Description: The *service_description_operands* contain all the information, except for the destination list, required by each DSU to transport the distribution.

Dist_ID

Description: The *distribution_identifier* contains information corresponding to the distribution originator.

Origin_RGN

Description: The *origin_RGN* is the first part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Origin_REN

Description: The *origin_REN* is the second part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Origin_DGN

Description: The *origin_DGN* is the first part of the user name of the distribution originator.

Note: For FS1, when the Dist_MU is of type REPORT and the distribution report was generated by SNA/DS, null user names will occur.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Origin_DEN

Description: The *origin_DEN* is the second part of the user name of the distribution originator.

Note: For FS1, when the Dist_MU is of type REPORT and the distribution report was generated by SNA/DS, null user names will occur.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Origin_Seqno

Description:	The *origin_sequence_number* is the number assigned to the distribution by the *origin_DSU*. The value ranges from 1 to 9999 for a Dist_MU *type* TRANSPORT, and is always 0 for a Dist_MU *type* REPORT.
Format:	Character string; each character is the EBCDIC representation of one digit of the sequence number.

Byte	Content
0-1	LT header
2-5	Sequence number

Notes:

- For Dist_MU *type* TRANSPORT, values range from X′F0F0F0F1′ to X′F9F9F9F9′.
- For Dist_MU *type* REPORT, value is X′F0F0F0F0′.

Origin_DTM

Description:	The *origin_date-time* is the date and time the distribution was originated by the origin DSU. Time is assumed to be local.
Note:	FS1 supports neither the GMT format nor the offset time format supported by FS2.
Format:	Byte string

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X′07C5′)
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)

Note: Refer to Chapter 15, “SNA/DS FS2 Encodings” for a complete discussion of the encoding and interpretation of date and time.

Agent_Correl

Description:	The *agent_correlation* is a string supplied by the origin agent. SNA/DS is not aware of its contents.
Format:	Undefined byte string

Dist_Gen_Options

Description: The *distribution_general_options* contains structures used by SNA/DS to condition its processing of the distribution.

Dist_Flags (type FS1)

Description: The *distribution_flags* indicate reporting services requested by the origin agent.

Format:

Bit	Content
0	Exception Report bit: 0 SNA/DS is requested to generate a report in case of an exception. 1 A report will not be generated by SNA/DS for this distribution.
1	Distribution Message Unit type bit: 0 Distribution is of type TRANSPORT. 1 Distribution is of type REPORT.
2-7	Reserved

Byte	Content
0-1	LT header
2	X′00′ Dist_MU *type* TRANSPORT with report requested X′80′ Dist_MU *type* TRANSPORT with no report requested X′C0′ Dist_MU *type* REPORT with no report requested Note: All other values are reserved.

Hop_Count

Description: The *hop_count* is the remaining number of hops that may be traversed by a SNA/DS distribution on its way toward its destination DSUs. The *hop_count* is set by the origin DSU in the Dist_MUs *type* TRANSPORT and by the reporting DSUs for the Dist_MUs *type* REPORT. The *hop_count* is decremented by 1 in every DSU through which the distribution passes. If the *hop_count* reaches 0 at an intermediate DSU, exception processing is invoked.

Format: Signed binary integer (1-origin)

Range of Values: Valid values range from 0 to 2^{15}-1.

Service_Parms

Description: The *service_parameters* structure describes the types and levels of service requested for the distribution. The parameters in this structure are provided by the origin agent. In FS1, the *service_parameters* are specified by the origin agent in Dist_MU *type* TRANSPORT. The specification for deriving the *service_parameters* for Dist_MU *type* REPORT is found in the description of *report_service_parameters* on page 14-18.

Note: The differences between FS1 and FS2 service parameter support are discussed below and throughout the SP descriptions.

Format: Special format consisting of ordered SP triplets of the following general structure:

Byte	Bit	Content
0		Parameter type: All parameter type byte values < X′80′ are defined by or reserved for SNA/DS. In FS1, all other parameter type byte values are reserved.
1		Comparison operator: 1100 REQUIRE_LEVEL_GE 1110 REQUIRE_SUPPORT_FOR Note: All other values for bits 0-3 are reserved.

Notes:

- FS2 supports architecturally defined SP comparison operators and values beyond those defined for FS1.
- FS2 tolerates customer-defined service parameters. Customer-defined service parameters have a parameter type byte value > X′80′.
- FS2 supports defaulted service parameters. Defaults are assigned for the SP comparison operator and SP value for each architecturally defined service parameter not specified for a given message unit.
- FS2 does not restrict the combination of SP triplets to the degree that FS1 does.

Byte	Content
0-1	LT header
2-31	Up to 10 different *service_parameter* (SP) triplets may be carried in one distribution. Each triplet appears in ascending sequence of parameter type. The architecturally defined service parameters are given below:

Note:

- Service parameters beyond the four defined below have been architecturally defined for FS2.

Priority SP Triplet

Byte	Content
0	X'01'
1	X'C0' REQUIRE_LEVEL_GE
2	X'F0' FAST X'D0' CONTROL X'80' DATA_16 (can be treated as DATAHI) X'78' DATA_15 (can be treated as DATAHI) X'70' DATA_14 (can be treated as DATAHI) X'68' DATA_13 (can be treated as DATAHI) X'60' DATA_12 (DATAHI) X'58' DATA_11 (can be treated as DATAHI) X'50' DATA_10 (can be treated as DATAHI) X'48' DATA_9 (can be treated as DATAHI) X'40' DATA_8 (can be treated as DATALO) X'38' DATA_7 (can be treated as DATALO) X'30' DATA_6 (can be treated as DATALO) X'28' DATA_5 (can be treated as DATALO) X'20' DATA_4 (DATALO) X'18' DATA_3 (can be treated as DATALO) X'10' DATA_2 (can be treated as DATALO) X'08' DATA_1 (can be treated as DATALO) Note: All other values are reserved.

Protection SP Triplet

Byte	Content
0	X'02'
1	X'C0' REQUIRE_LEVEL_GE
2	X'10' LEVEL1: safe store may be performed. X'30' LEVEL2: safe store must be performed. Note: All other values are reserved.

Capacity SP Triplet

Byte	Content
0	X′03′
1	X′C0′ REQUIRE_LEVEL_GE 1 X′E0′ REQUIRE_SUPPORT_FOR
2	Capacity value is the exponent of the power of 2 that represents the value of the required capacity for the *server_object* in the DTMU: X′00′ ZERO X′OC′ 4KB (4 kilobytes) X′FF′ INDEFINITE Note: All other values are reserved.

Notes:

- The capacity requirement is for the *server_object*, and does not include the capacity needed to store and handle the other structures of the DTMU.
- FS2 implementations may accept other capacity levels (including 4KB) as long as they can route the distribution responsibly.
- Capacity SP values beyond the three defined above for FS1 have been architecturally defined for FS2 (e.g., 1MB).
- In FS2, the capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified.
- Receiving FS2 DSUs are always able to receive a capacity level of INDEFINITE; although originating FS2 DSUs never generate that capacity level. The FS2 equivalent to INDEFINITE is 16MB (X′18′).

Security RSP Triplet

Byte	Content
0	X′04′
1	X′C0′ REQUIRE_LEVEL_GE
2	X′01′ LEVEL1: security is not required. X′20′ LEVEL2: security is required. Note: All other values are reserved.

Server_Object_Ind

Description: The *server_object_indicator* indicates whether a *server_object* is present or not. The only values supported are 0 and 1.

Presence Rule: Contains X′0001′ only for Dist_MU *type* TRANSPORT.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2-3	X′0000′ no *server_object* present in this MU X′0001′ a *server_object* present in this MU Note: All other values are reserved.

Origin_Agent

Description: The *origin_agent* is the transaction program at the DSU at which the distribution originated.

Format: Character string, except for first byte

CGCSGID	String Conventions
01130-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Report-To_Address

Description: The *report-to_address* contains the name of the DSU and user to which any distribution reports are sent.

Presence Rule: This information may be present only in Dist_MU *type* TRANSPORT.

Report-To_RGN

Description: The *report-to_RGN* is the first part of the DSU name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT. This is typically, but not necessarily, the network ID.

Note: In FS2, the *report-to_RGN* occurs in both the DTMU and DRMU.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_REN

Description: The *report-to_REN* is the second part of the DSU name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT. This is typically, but not necessarily, the LU name.

Note: In FS2, the *report-to_REN* occurs in both the DTMU and DRMU.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_DGN

Description: The *report-to_DGN* is the first part of the user name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT.

Note: In FS2, the *report-to_DGN* occurs in both the DTMU and DRMU.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_DEN

Description: The *report-to_DEN* is the second part of the user name to which distribution reports are to be sent. This information is valid only if Dist_MU is of type TRANSPORT.

Note: In FS2, the *report-to_DEN* occurs in both the DTMU and DRMU.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_Options

Description: The *report-to_options* contains information involved in processing any reports generated as part of the distribution.

Presence Rule: This information may be present only in Dist_MU *type* TRANSPORT.

Report_Service_Parms

Description: The *report_service_parameters* structure describes the service requested for the distribution report by the origin agent when the agent wants to override the *service_parameters* that would be routinely generated by the reporting DSU for the report MU. If *report_service_parameters* are specified, they are used as the *service_parameters* in any Dist_MU *type* REPORT that is generated as part of the distribution. If the origin agent does not specify one or more of the *report_service_parameters*, a DSU that generates a report derives appropriate *service_parameters* for the Dist_MU *type* REPORT from the *service_parameters* in the Dist_MU *type* TRANSPORT. The comparison operators and values derived for the protection, capacity, and security parameters are the same as those specified in the Dist_MU *type* TRANSPORT.

For the priority service parameter, the value derived is either FAST or CONTROL. FAST is used if the Dist_MU *type* TRANSPORT specified FAST priority; CONTROL is used if the Dist_MU *type* TRANSPORT specified a DATA_N priority. CONTROL priority is used only in Dist_MUs *type* REPORT; it may not be specified for the priority service parameter in a Dist_MU *type* TRANSPORT. If the origin agent explicitly specifies a value for the priority report service parameter, the value may be FAST, CONTROL, or DATA_N. The comparison operator for the priority service parameter is always REQUIRE_LEVEL_GE.

Notes: Following are RSP notes specific to FS2:

- For FS2, the comparison operators and values derived for the protection, security, and acceptable delay parameters are the same as those specified (explicitly or implicitly) in the DTMU. The FS2 values for the priority service parameter in the DRMU are derived using the same logic as defined above.
- FS2 tolerates customer-defined service parameters. Customer-defined service parameters have a parameter type byte value > X′80′.
- FS2 supports defaulted report service parameters. Defaults are assigned for the SP comparison operator and SP value for each architecturally defined service parameter not specified for a given message unit.
- FS2 supports SP comparison operators and SP values beyond those defined for FS1.

Format: Special format consisting of ordered, optional *report_service_parameter* triplets of the same general structure as for *service_parameters*. See *service_parameters* on page 14-12.

Byte	Content
0-1	LT header
2-31	Up to 10 different *report_service_parameter* (RSP) triplets may be carried in one distribution. Each triplet, when present, appears in ascending sequence of parameter type. The first three parameters—priority, protection, and capacity—are present if report service parameters are to be specified.

Notes:

- In FS2, all service parameters are optional in both the DTMU and DRMU.
- In FS2, the capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified. Note that the capacity RSP is specified in FS1.

Priority RSP Triplet

Byte	Content
0	X'01'
1	X'C0' REQUIRE_LEVEL_GE
2	X'F0' FAST X'D0' CONTROL X'80' DATA_16 (can be treated as DATAHI) X'78' DATA_15 (can be treated as DATAHI) X'70' DATA_14 (can be treated as DATAHI) X'68' DATA_13 (can be treated as DATAHI) X'60' DATA_12 (DATAHI) X'58' DATA_11 (can be treated as DATAHI) X'50' DATA_10 (can be treated as DATAHI) X'48' DATA_9 (can be treated as DATAHI) X'40' DATA_8 (can be treated as DATALO) X'38' DATA_7 (can be treated as DATALO) X'30' DATA_6 (can be treated as DATALO) X'28' DATA_5 (can be treated as DATALO) X'20' DATA_4 (DATALO) X'18' DATA_3 (can be treated as DATALO) X'10' DATA_2 (can be treated as DATALO) X'08' DATA_1 (can be treated as DATALO) Note: All other values are reserved.

Protection RSP Triplet

Byte	Content
0	X′02′
1	X′C0′ REQUIRE_LEVEL_GE
2	X′10′ LEVEL1: safe store may be performed. X′30′ LEVEL2: safe store must be performed. Note: All other values are reserved.

Capacity RSP Triplet

Byte	Content
0	X′03′
1	X′C0′ REQUIRE_LEVEL_GE
1	X′E0′ REQUIRE_SUPPORT_FOR
2	X′00′ ZERO
	Notes: All other values are reserved. Also, all FS1 implementations are able to receive distribution reports of 4KB capacity (X′0C′). New FS1 implementations always send distribution reports of ZERO capacity.

Notes:

- The capacity requirement is for the *server_object*, and does not include the capacity needed to store and handle the other structures of the DTMU.
- FS2 implementations accept other capacity levels (including 4KB) as long as they can route the distribution responsibly.
- Capacity SP values beyond the three defined above for FS1 have been architecturally defined for FS2 (e.g., 1MB).
- In FS2, the capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified.
- Receiving FS2 DSUs are always able to receive a capacity level of INDEFINITE; although originating FS2 DSUs never generate that capacity level. The FS2 equivalent to INDEFINITE is 16MB (X′18′).

Security RSP Triplet

Byte	Content
0	X′04′
1	X′C0′ REQUIRE_LEVEL_GE
2	X′01′ LEVEL1: security is not required. X′20′ LEVEL2: security is required. Note: All other values are reserved.

Report-To_Agent

Description:	The *report-to_agent* is the name of the application transaction program to be started after the report is queued for delivery. If *report-to_agent* is absent in the Dist_MU *type* TRANSPORT, the value specified in the Dist_MU *type* TRANSPORT for *origin_agent* is used in the Dist_MU *type* REPORT for *origin_agent*.
Presence Rule	This information may be present only in Dist_MU *type* TRANSPORT.
Format:	Character string, except for first byte.

CGCSGID	String Conventions
01130-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Agent_Object

Description:	The *agent_object* is directly supplied by the origin agent. It is never parsed by the distribution service and is directly delivered, unchanged, to the agent at each destination.
Format:	Undefined byte string

Destination_Operands

Description:	The *destination_operands* are the list of destinations for the distribution. Up to 256 destinations are allowed if the distribution is of type TRANSPORT; exactly one destination, if the distribution is of type REPORT. The destinations are encoded as a fully factored, partially factored, or unfactored list of users and DSUs (see the following example).

Example: The following is a list of destinations (qualified by RGN.REN.DGN.DEN):
A.K.DA.U1, A.K.DA.U2, A.K.DB.U3, A.K.DB.U4,
A.L.DC.U5, A.L.DC.U6, A.L.DD.U7, A.L.DD.U8,
B.M.DE.U9, B.M.DE.U10, B.M.DF.U11, B.M.DF.U12,

B.N.DG.U13, B.N.DG.U14, B.N.DH.U15, and B.N.DH.U16.

The list may appear factored in *destination_operands* as follows:

- Fully factored:

```
A(K(DA(U1
       U2)
     DB(U3
       U4))
   L(DC(U5
       U6)
     DD(U7
       U8)))
 B(M(DE(U9
       U10)
     DF(U11
       U12))
   N(DG(U13
       U14)
     DH(U15
       U16))))
```

- Partially factored:

```
(A(K(DA(U1)
     DA(U2)
     DB(U3
       U4))
   L(DC(U5
       U6))
   L(DD(U7
       U8)))
 B(M(DE(U9
       U10)
     DF(U11
       U12))
   N(DG(U13))
   N(DG(U14))
   N(DH(U15
       U16))))
```

- Unfactored, equivalent to the initial list:

```
(A(K(DA(U1)))
 A(K(DA(U2)))
 A(K(DB(U3)))
 A(K(DB(U4)))
 A(L(DC(U5)))
 A(L(DC(U6)))
 A(L(DD(U7)))
 A(L(DD(U8)))
 B(M(DE(U9)))
 B(M(DE(U10)))
 B(M(DF(U11)))
 B(M(DF(U12)))
```

B(N(DG(U13)))
B(N(DG(U14)))
B(N(DH(U15)))
B(N(DH(U16))))

In the above lists, *"("* represents *begin_dest_operands*, *begin_REN_list*, *begin_DGN_list*, or *begin_DEN_list*. *")"* represents *end_DEN_list*, *end_DGN_list*, *end_REN_list*, or *end_dest_operands*. (Inner parentheses have precedence over outer parentheses.)

Begin_Dest_Operands

Description: The *beginning_of_the_destination_operands* marks the beginning of the *destination_list*.

Format: Constant byte string; value is X'C35201'

Dest_RGN_List

Description: The *destination_RGN_list* associates one destination RGN with at least one destination REN.

Dest_RGN

Description: The *destination_RGN* is the first part of a destination DSU name. This is typically, but not necessarily, the network ID.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X'40') characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X'40') characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_REN_List

Description: The *beginning_of_the_destination_REN_list* marks the beginning of a list of one or more *dest_REN*(s).

Format: Constant byte string; value is X'C35301'

Dest_REN_List

Description: The *destination_REN_list* associates one destination REN with at least one destination DGN.

Dest_REN

Description: The *destination_REN* is the second part of a destination DSU name. This is typically, but not necessarily, the LU name.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_DGN_List

Description: The *beginning_of_the_destination_DGN_list* marks the beginning of a list of one or more *dest_DGN*(s).

Format: Constant byte string; value is X′C35401′

Dest_DGN_List

Description: The *destination_DGN_list* associates one *dest_DGN* with at least one *dest_DEN*.

Dest_DGN

Description: The *destination_DGN* is the first part of the name of a destination user.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_DEN_List

Description: The *beginning_of_the_destination_DEN_list* marks the beginning of a list of one or more *dest_DEN*(s).

Format: Constant byte string; value is X′C35501′

Dest_DEN

Description: The *destination_DEN* is the second part of the name of a destination user.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

End_DEN_List

Description: The *end_destination_DEN_list* marks the end of the list begun by the corresponding *begin_DEN_list*.

End_DGN_List

Description: The *end_destination_DGN_list* marks the end of the list begun by the corresponding *begin_DGN_list*.

End_REN_List

Description: The *end_destination_REN_list* marks the end of the list begun by the corresponding *begin_REN_list*.

End_Dest_Operands

Description: The *end_destination_operands* marks the end of the *destination_list*.

Dist_Server_Operands

Description: The *distribution_server_operands* structure contains the *server_prefix* and the *server_object*.

Presence Rule: This information occurs only in Dist_MU *type* TRANSPORT when *server_object_ind* = X′0001′.

Server_Prefix

Description: The *server_prefix* contains information associated with the *server_object*.

Server_Obj_Byte_Count

Description:	The *server_object_byte_count* is the number of bytes of all the segments of the *server_object*.
Note:	In FS1, the byte count need not be accurate. In FS2, the originating DSU either supplies a correct byte count or omits the field completely.
Presence Rule:	Optional when the *server_object* is present; otherwise, precluded.
Format:	Unsigned binary integer (1-origin)
Range of Values:	Valid values range from 1 to 2^{64}-2.

Server

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

CGCSGID	String Conventions
01130-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Server_Parms

Description:	The *server_parameters* structure contains parameters passed by SNA/DS to the destination server.
Note:	This structure is never sent, and is retired in FS2.
Format:	Undefined byte string

Server_Object

Description:	The *server_object* is identified by the origin agent and is fetched by the origin server during transmission of the Dist_MU *type* TRANSPORT. At each destination, the *server_object* is stored by the destination server and a notification of its receipt is delivered to the destination agent.
Length Restriction:	The maximum segment size for FS1 is 32511.
Format:	Undefined byte string

DS_Suffix (FS1)

Description: The *distribution_services_suffix* contains no information and marks the end of the message unit.

Dist_Report_Operands

Description: The *distribution_report_operands* structure contains all the report information describing the condition of a particular distribution.

Presence Rule: This information occurs only when Dist_MU is of type REPORT.

Report_Operands

Description: The *report_operands* structure contains all information pertaining to the originator of the distribution and the detector of an exception.

Report_Correlation

Description: The *report_correlation* contains information that uniquely identifies a distribution being reported on.

Reported-On_Origin_DGN

Description: The *reported-on_origin_DGN* is the first part of the name of the user that originated the distribution.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Reported-On_Origin_DEN

Description: The *reported-on_origin_DEN* is the second part of the name of the user that originated the distribution.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Reported-On_Seqno

Description: The *reported-on_origin_sequence_number* is the sequence number of the distribution being reported on.

Format: Character string; each character represents the EBCDIC representation of one digit of the sequence number.

Byte	Content
0-1	LT header
2-5	Sequence number Note: Values range from X′F0F0F0F1′ to X′F9F9F9F9′.

Reported-On_DTM

Description: The *reported-on_date-time* is the date and time the distribution was originated.

Note: FS1 supports neither the GMT format nor the offset time format supported by FS2.

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X′07C5′)
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)

Note: Refer to Chapter 15, “SNA/DS FS2 Encodings” for a complete discussion of the encoding and interpretation of date and time.

Reported-On_Agent_Correl

Description: The *reported-on_agent_correlation* is a string that was supplied by the origin agent at the origin DSU.

Format: Undefined byte string

Receiving_DSU

Description: The *receiving_DSU* is the name of the DSU to which a distribution was being sent.

Receiving_RGN

Description: The *receiving_RGN* is the first part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the network ID.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Receiving_REN

Description: The *receiving_REN* is the second part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the LU name.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Gen_SNADS_Report

Description: The *general_SNADS_report* contains the SNA/DS report applicable to each user specified in *specific_report* for which a *spec_SNADS_report* is not supplied.

Note: Older DSUs may generate both *gen_SNADS_report* and *gen_DIA_report* in a single MU. All DSUs are able to receive such MUs. However, DSUs may ignore *gen_DIA_report* if *gen_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.

Presence Rule: This information occurs when *gen_SNADS_type* = X′0001′.

Gen_SNADS_Type

Description: The *general_SNADS_type* indicates that a SNA/DS condition is being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' SNA/DS report Note: Any other value indicates that this is not a *gen_SNADS_report*.

Gen_SNADS_Content

Description: The *general_SNADS_contents* contains information describing the condition being reported on.

Gen_SNADS_Cond_Code

Description: The *general_SNADS_condition_code* is the particular condition being reported on.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2-3	X'0001' routing exception X'0002' unknown user name X'0003' hop count exhausted X'0004' format exception X'0005' function not supported X'0006' specific-server exception X'0007' unknown resource name (specific server) X'0008' invalid server parameters X'0009' unknown resource name (destination agent) X'000C' operator intervention (purging) X'000D' user names lost X'000E' resource not available X'000F' system exception X'0010' insufficient resource X'0011' storage-medium exception X'0012' REMU exception X'0013' server object size incompatible with capacity level Note: All other values are reserved.

Gen_DIA_Report

Description: The *general_DIA_report* describes an application-layer condition. The *gen_DIA_report* applies to all users specified in *specific_report*. The interaction between *gen_DIA_report* and *spec_DIA_report* is defined by DIA.

Note: Older DSUs may generate both *gen_SNADS_report* and *gen_DIA_report* in a single MU. All DSUs can receive such MUs. However, DSUs may ignore *gen_DIA_report* if *gen_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.

Presence Rule: This information occurs when *gen_DIA_type* ≠ X′0001′.

Gen_DIA_Type

Description: The *general_DIA_type* indicates the type of DIA condition being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X′0001′ indicates this is not a *gen_DIA_report* X′0200′ DIA application exceptions X′FEFF′ reserved for 5520 migration Note: All other values are reserved.

Gen_DIA_Contents

Description: The *general_DIA_contents* structure contains a DIA-defined byte string.

Length Restriction: Older DSUs may generate MUs with length of up to 517. All DSUs receive such MUs without generating an exception. However, DSUs may modify such MUs to force the length to be 69 or less. For *gen_DIA_type* of X′0200′ (DIA application exceptions), the truncation algorithm is given in the *DIA Transaction Programmer's Guide*. The length is at least 7, since *gen_DIA_contents* contains at least a null LT (an LT of length 2).

Format: Undefined byte string

Specific_Report

Description: The *specific_report* contains the portion of the destination users that are being reported on. Any specific SNA/DS and/or DIA reports are also specified within this structure.

Begin_Report_DGN_List

Description: The *beginning_of_report_DGN_list* marks the beginning of the *specific_report*.

Format: Constant byte string; value is X′C35401′

Report_DGN_List

Description: The *report_DGN_list* associates one *reported-on_dest_DGN* with at least one *reported-on_dest_DEN*.

Reported-On_Dest_DGN

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

Note: In FS1, for a SNA/DS condition code of X′000D′ (lost user names), user names will be null.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Begin_Report_DEN_List

Description: The *beginning_of_report_DEN_list* marks the beginning of a list of one or more *reported-on_dest_DEN*s.

Format: Constant byte string; value is X′C35501′

Report_DEN_List

Description: The *report_DEN_list* associates one *reported-on_dest_DEN* with a specific SNA/DS and/or DIA report.

Reported-On_Dest_DEN

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

Note: In FS1, for a SNA/DS condition code of X′000D′ (lost user names), user names will be null.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	00961-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Spec_SNADS_Report

Description: The *specific_SNADS_report* is a report on one particular user. This report overrides the *gen_SNADS_report*, if one exists, for that particular user.

Note: Older DSUs may generate both *spec_SNADS_report* and *spec_DIA_report* in a single MU. All DSUs can receive such MUs. However, DSUs may ignore *spec_DIA_report* if *spec_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.

Presence Rule: This information occurs when *spec_SNADS_type* = X′0001′.

Spec_SNADS_Type

Description: The *specific_SNADS_type* indicates that a SNA/DS condition is being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X′0001′ SNA/DS report Note: Any other value indicates that this is not a *spec_SNADS_report*.

Spec_SNADS_Cont

Description: The *specific_SNADS_contents* contains information describing a condition being reported on.

Spec_SNADS_CC

Description: The *specific_SNADS_condition_code* describes the particular condition being reported on.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2-3	X'0001' routing exception X'0002' unknown user name X'0003' hop count exhausted X'0004' format exception X'0005' function not supported X'0006' specific-server exception X'0007' unknown resource name (specific server) X'0008' invalid server parameters X'0009' unknown resource name (destination agent) X'000C' operator intervention (purging) X'000D' user names lost X'000E' resource not available X'000F' system exception X'0010' insufficient resource X'0011' storage-medium exception X'0012' REMU exception X'0013' server object size incompatible with capacity level Note: All other values are reserved.

Spec_DIA_Report

Description: The *specific_DIA_report* describes a DIA-specific report on one particular user.

Note: Older DSUs may generate both *spec_SNADS_report* and *spec_DIA_report* in a single MU. All DSUs can receive such MUs. However, DSUs may ignore *spec_DIA_report* if *spec_SNADS_report* is present. A sending DSU never generates both a DIA report and a SNA/DS report for multiple destinations.

Presence Rule: This information occurs when *spec_DIA_type* ≠ X'0001'.

Spec_DIA_Type

Description: The *specific_DIA_type* indicates the type of DIA condition being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X′0001′ indicates this is not a *spec_DIA_report* X′0200′ DIA application exceptions X′FEFF′ reserved for 5520 migration Note: All other values are reserved.

Spec_DIA_Contents

Description: The *specific_DIA_contents* structure contains a DIA-defined byte string.

Length Restriction: Older DSUs may generate MUs with length of up to 517. All DSUs receive such MUs without generating an exception. However, DSUs may modify such MUs to force the length to be 69 or less. For *spec_DIA_type* of X′0200′ (DIA application exceptions), the truncation algorithm is given in the *DIA Transaction Programmer's Guide*. The length is at least 7, since *spec_DIA_contents* contains at least a null LT (an LT of length 2).

Format: Undefined byte string

End_Report_DEN_List

Description: The *end_report_DEN_list* marks the end of the list begun by *begin_report_DEN_list*.

End_Report_DGN_List

Description: The *end_report_DGN_list* marks the end of the *specific_report.*

Sender_Exception_MU (Type FS1)

Description: The *sender_exception_MU* (type FS1) is sent from the sender to the receiver when the sender detects an exception while sending a Dist_MU.

Format: Byte string

Byte	Bit	Content
0-4		LLIDF header
5		Severity:
	0-1	11 catastrophic
		Class:
	2-7	000101 sender
6		Exception condition code:
		X'06' execution terminated
		X'0B' I/O error
		X'0F' length invalid
		X'18' content error
7		Exception object:
		X'01' IU prefix
		X'07' command
		X'0C' document unit
		X'13' IU suffix
		X'17' unknown subfield
		X'1A' distribution object prefix
		X'1B' distribution object data

Note: Other values and their corresponding meanings are represented under *receiver_exception_code*.

Receiver_Exception_MU (Type FS1)

Description: The *receiver_exception_MU* (type FS1) is sent from the receiver to the sender when the receiver detects an exception while receiving a Dist_MU.

Receiver_Exception_Command

Description: The *receiver_exception_command* contains all information used for identifying the exception that occurred.

Receiver_Exception_Correl

Description: The *receiver_exception_correlation* contains the *prefix* ID value from the rejected Dist_MU.

Format: Byte string

Byte	Content
0-4	LLIDF header
5	Correlation field: X′00′ Note: All other values are reserved.
6	Command sequence number: X′01′ Note: All other values are reserved.
7-22	Correlation MU ID; value from the *prefix* of the Dist_MU

Exception_And_Reply_Data

Description: The *exception_and_reply_data* contains information pertaining to the exception causing the rejection of the Dist_MU.

Receiver_Exception_Code

Description: The *receiver_exception_code* identifies the type of exception encountered and, conditionally, the portion of the Dist_MU containing the exception.

Format: Byte string

Byte	Bit	Content
0-4		LLIDF header
5		Severity:
	0-1	11 catastrophic Note: All other values for bits 0-1 are reserved.
		Class:
	2-7	000010 syntactic 000011 semantic 000100 process Note: All other values for bits 2-7 are reserved or defined elsewhere.
6		Exception condition code (indicates reason for exception): X′01′ function not supported X′02′ data not supported X′04′ resource not available X′06′ execution terminated X′07′ data not found X′08′ segmentation X′0A′ sequence X′0B′ I/O error X′0C′ ID invalid X′0E′ format invalid X′0F′ length invalid X′10′ indicator invalid X′11′ range exceeded X′15′ subfield length invalid X′16′ subfield type invalid X′17′ invalid parameters X′18′ content error Note: All other values are reserved.
7		Exception object (indicates the syntactical entity in error): X′01′ IU prefix X′02′ IU identifier X′07′ command X′08′ command operand X′09′ operand value X′0C′ document unit X′0D′ document unit identifier

Byte	Bit	Content
		X′0E′ document profile X′0F′ document profile parameter X′10′ document content introducer X′11′ document content control X′12′ document content data X′13′ IU suffix X′14′ segment X′16′ unsupported subfield X′17′ unknown subfield X′1A′ distribution object prefix X′1B′ distribution object data Note: All other values are reserved.
8-254		Exception data contains the Dist_MU structures in error

Reply_Data

Description: The *reply_data* describes which DSU rejected the Dist_MU and why the Dist_MU was rejected.

SNADS_Report

Description: The *SNADS_report* contains information describing the particular SNA/DS exception that caused the Dist_MU to be rejected.

SNADS_Report_Type

Description: The *SNADS_report_type* indicates that a SNA/DS exception is being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X′0001′ SNA/DS report Note: Any other value indicates that this is not a *SNADS_report*.

SNADS_Report_Cont

Description: The *SNADS_report_contents* structure contains information describing the type of SNA/DS condition in the Dist_MU.

SNADS_Report_CC

Description: The *SNADS_report_condition_code* describes the particular SNA/DS condition that caused the Dist_MU to be rejected.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2-3	X'0001' routing exception X'0002' unknown user name X'0003' hop count exhausted X'0004' format exception X'0005' function not supported X'0006' specific-server exception X'0007' unknown resource name (specific server) X'0008' invalid server parameters X'0009' unknown resource name (destination agent) X'000E' resource not available X'000F' system exception X'0010' insufficient resource X'0011' storage-medium exception X'0013' server object size incompatible with capacity level Note: All other values are reserved.

DIA_Report

Description: The *DIA_report* describes a DIA condition being reported.

Note: When generating a Dist_MU *type* REPORT with report information supplied by a REMU (type FS1), the reporting DSU may ignore *DIA_report*.

Presence Rule: This information occurs when *gen_DIA_type* ≠ X'0001'.

DIA_Report_Type

Description: The *DIA_report_type* indicates the type of DIA condition being reported.

Format: Hexadecimal code

Byte	Content
0-4	LLIDF header
5-6	X'0001' indicates this is not a *DIA_report* X'0200' DIA application exceptions X'FEFF' reserved for 5520 migration Note: All other values are reserved.

DIA_Report_Cont

Description:	The *DIA_report_contents* structure contains a DIA-defined byte string.
Format:	Undefined byte string

Transaction Program and Server Names

Following is a list of all transaction program and server names defined for SNA/DS in the FM header 5 (Attach), in the Distribution MU, or used internally in the distribution service unit (DSU).

Code	Meaning
X′20**F0F0F0**′	DIA process destination transaction program name
X′20**F0F0F1**′	DIA server name
X′20**F0F0F2**′	DIASTATUS transaction program name
X′21**F0F0F1**′	DS_SEND transaction program name (FS1)
X′21**F0F0F2**′	DS_RECEIVE transaction program name (FS1)
X′21**F0F0F3**′	DS_ROUTER_DIRECTOR transaction program name
X′21**F0F0F6**′	SNA/DS general server name
X′30**F0F0F2**′	Object Distribution transaction program.
X′30**F0F0F3**′	Object Distribution server transaction program.

Code Points Used by SNA/DS FS1

The values of the ID component of the LLIDF structure as used for SNA/DS GDS variables are shown below:[1]

ID	Structure Name
C001*	In DIA, MU PREFIX; in SNA/DS, Prefix within DIST_MU or within REMU (type FS1)
C101*	in DIA, MU CMD NO REPLY ACKNOWLEDGE; in SNA/DS, Command within REMU (type FS1)
C105	Command, DIST_MU
C322*	in DIA, MU OPERAND IMM DATA EXCEPTION-CODE; in SNA/DS, Exception Code, within REMU (type FS1)
C328*	in DIA, MU OPERAND IMM DATA DATA CORRELATION; in SNA/DS, Correlation, within REMU (type FS1)
C32D*	in DIA, MU OPERAND IMM DATA USER-DATA; in SNA/DS, Agent Object within DIST_MU
C33D*	in DIA, MU OPERAND IMM DATA STATUS-INFORMATION; in SNA/DS, Distribution General Options, within DIST_MU
C340*	in DIA, MU OPERAND IMM DATA DISTRIBUTION-IDENTIFIER; in SNA/DS, Distribution Identifier, within DIST_MU
C343*	in DIA, MU OPERAND IMM DATA GENERAL-ROUTING-DATA; in SNA/DS, Report-To Options within DIST_MU
C345*	in DIA, MU OPERAND IMM DATA REPLY DATA; in SNA/DS, Reply Data, within REMU (type FS1)
C350	Beginning of Destination Operand Lists, of the Specific Report Lists, within DIST_MU
C351	End of Destination Operands Lists, of the Specific Report Lists, within DIST_MU
C352	Routing Group Name (RGN) of Destination Operands, within DIST_MU
C353	Routing Element Name (REN) of REN List, within DIST_MU
C354	Distribution Group Name (DGN) of DGN List, within DIST_MU
C355	Distribution Element Name (DEN) of DEN List, within DIST_MU
C356	Report Type, within DIST_MU
C357	Report Contents, within DIST_MU
C360	Report-To Address, within DIST_MU
C361	Receiving DSU, within DIST_MU or within REMU (type FS1)
C908	Server Object, within DIST_MU

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

C90A Server Prefix, within DIST_MU

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

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

Terminology Mappings

Figure 14-5 (Page 1 of 3). Terminology Mappings

FS2 TERMINOLOGY	Current FS1 TERMINOLOGY	Old FS1 TERMINOLOGY
Dist_Transport_MU	Dist_MU (type Transport)	Dist_IU (type Data)
Transport_Prefix	Prefix	Prefix
Hop_Count	Hop_Count	Dist_Dest_Hops
MU_ID	—	—
Transport_Command	Dist_Command	Dist_CMD
Dist_Flags	Dist_Flags (FS1)	Dist_Flags
Service_Parms	Service_Parms	DSL
Server_Obj_Byte_Count	Server_Obj_Byte_Count	Data_Size
Origin_Agent	Origin_Agent	Dest_TPN
Server	Server	Server_Name
Origin_DSU	—	—
Origin_RGN	Origin_RGN	Orig_RGN
Origin_REN	Origin_REN	Orig_REN
Origin_User	—	—
Origin_DGN	Origin_DGN	Orig_DGN
Origin_DEN	Origin_DEN	Orig_DEN
Seqno_DTM	Origin_Seqno, Origin_DTM	Orig_Seqno, Orig_DTM
Ext_Net	—	—
Agent_Correl	Agent_Correl	Orig_Correl
Report-To_DSU	—	—
Report-To_RGN	Report-To_RGN	Fdbk_RGN
Report-To_REN	Report-To_REN	Fdbk_REN
Report-To_User	—	—
Report-To_DGN	Report-To_DGN	Fdbk_DGN
Report-To_DEN	Report-To_DEN	Fdbk_DEN
Report_Service_Parms	Report_Service_Parms	Fdbk_DSL
Report-To_Agent	Report-To_Agent	Fdbk_TPN
Dest_Agent	—	—
Unrecognized_Reserve	—	—
Dest_List	Destination_Operands	Destination_Operands
Dest	—	—
Dest_DSU	—	—
Dest_RGN	Dest_RGN	Dest_RGN
Dest_REN	Dest_REN	Dest_REN
Dest_User	—	—
Dest_DGN	Dest_DGN	Dest_DGN
Dest_DEN	Dest_DEN	Dest_DEN
Agent_Object	Agent_Object	Dest_Appl_Parms
Server_Object	Server_Object	Distrib_Object_Data
Ext_Net_Correl	—	—

Figure 14-5 (Page 2 of 3). Terminology Mappings

FS2 TERMINOLOGY	Current FS1 TERMINOLOGY	Old FS1 TERMINOLOGY
Ext_Net_Object	—	—
DS_Suffix	DS_Suffix	Suffix
Dist_Report_MU	Dist_MU (type Report)	Dist_IU (type Status)
Report_Prefix	—	—
Report_Command	—	—
Reporting_DSU	—	—
Reporting_RGN	—	—
Reporting_REN	—	.—
Report_DTM	—	—
Report-To_DSU_User	—	—
Report_Information	—	—
Reported-On_Origin_DSU	—	—
Reported-On_Origin_RGN	—	—
Reported-On_Origin_REN	—	—
Reported-On_Origin_User	—	—
Reported-On_Origin_DGN	Reported-On_Origin_DGN	Orig_DGN
Reported-On_Origin_DEN	Reported-On_Origin_DEN	Orig_DEN
Reported-On_Seqno_DTM	Reported-On_Seqno, Reported-On_DTM	Orig_Seqno, Orig_DTM
Reported-On_Ext_Net	—	—
Reported-On_Ext_Net_Correl	—	—
Reported-On_Agent_Correl	Reported-On_Agent_Correl	Orig_Correl
Reported-On_Dest_Agent	—	—
Reported-On_Hop_Count	—	—
SNA_Condition_Report	—	—
SNA_Report_Code	—	—
Structure_Report	—	—
Structure_State	—	—
Structure_Contents	—	—
Parent_Spec	—	—
Parent_ID_Or_T	—	—
Parent_Class	—	—
Parent_Position	—	—
Parent_Instance	—	—
Structure_Spec	—	—
Structure_ID_Or_T	—	—
Structure_Class	—	—
Structure_Position	—	—
Structure_Instance	—	—
Structure_Segment_Num	—	—
Structure_Byte_Offset	—	—
Sibling_List	—	—
Reported-On_Dest_List	Specific_Report	Specific_Status

Figure 14-5 (Page 3 of 3). Terminology Mappings

FS2 TERMINOLOGY	Current FS1 TERMINOLOGY	Old FS1 TERMINOLOGY
Reported-On_Dest_Pfx	—	—
Reported-On_Dest	—	—
Reported-On_Dest_DSU	—	—
Reported-On_Dest_RGN	—	—
Reported-On_Dest_REN	—	—
Reported-On_Dest_User	—	—
Reported-On_Dest_DGN	Reported-On_Dest_DGN	Stat_DGN
Reported-On_Dest_DEN	Reported-On_Dest_DEN	Stat_DEN
Reported-On_Dest_Sfx	—	—
Supplemental_Report	—	—
Dist_Continuation_MU	—	—
Continuation_Prefix	—	—
Restarting_Byte_Position	—	—
Sender_Exception_MU	Sender_Exception_MU	Suffix (type 2)
Receiver_Exception_MU	Receiver_Exception_MU	Ack_IU
Receiver_Exception_Command	Receiver_Exception_Command	Ack_Cmd
Sender_Retry_Action	—	—
Receiving_DSU	Receiving_DSU	Rcv_DSUN
Receiving_RGN	Receiving_RGN	Rcv_DSUN_RGN
Receiving_REN	Receiving_REN	Rcv_DSUN_REN
Completion_Query_MU	—	—
Completion_Report_MU	—	—
Indicator_Flags	—	—
Last_Structure_Received	—	—
Last_Byte_Received	—	—
Purge_Report_MU	—	—
Reset_Request_MU	—	—
Reset_DTM	—	—
Reset_Accepted_MU	—	—

End of Chapter 14

Chapter 15. SNA/DS FS2 Encodings

Introduction

This chapter contains the format descriptions of the FS2 message units. The format descriptions are comprised of two parts: *header description tables* and *structure descriptions*. A header description table contains the header information for each structure associated with a particular message unit. A structure description contains a prose description of the structure, bit-level representations, and any presence rules or length restrictions associated with a particular structure.

The definition of SNA/Distribution Services (SNA/DS) requires a byte-accurate description of the formats that must be understood by all DSUs. The SNA/DS formats are described in terms of encoded fields referred to as ″structures″ and the hierarchical relationship between these structures. In this chapter, the header description tables show each structure and its header.

Refer to Appendix B, “Common Structures” for a complete definition and classification of the encoding structures used in the following tables.

Header Description Tables for FS2 Message Units

DISTRIBUTION TRANSPORT MESSAGE UNIT (DTMU)

Figure 15-1 (Page 1 of 2). Distribution Transport Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Transport_MU	15-11	Del-ID	pfx	≥53*	1	Y	Y	≥4	—
Transport_Prefix	15-11	ID/pfx	1570	8-18	1	N	Y	1-3	—
Hop_Count	15-11	T	01	4	1	—	—	—	—
MU_ID	15-11	T	03	6	0-1*	—	—	—	—
MU_Instance_Number	15-11	T	06	4	0-1*	—	—	—	—
Transport_Command	15-12	ID/seg	1571	29-4096*	1	Y	Y	≥3	—
Dist_Flags	15-12	T	01	5	0-1	—	—	—	—
Service_Parms	15-13	T	02	5-32	0-1	—	—	—	—
Server_Obj_Byte_Count	15-16	T	03	10	0-1*	—	—	—	—
Origin_Agent	15-16	T	04	3-10	1	—	—	—	—
Server	15-17	T	05	3-10	0-1*	—	—	—	—
Origin_DSU	15-17	T	06	8-22	1	N	Y	2	—
Origin_RGN	15-17	T	01	3-10	1	—	—	—	—
Origin_REN	15-17	T	02	3-10	1	—	—	—	—
Origin_User	15-17	T	07	8-22	0-1	N	Y	2	—
Origin_DGN	15-18	T	01	3-10	1	—	—	—	—
Origin_DEN	15-18	T	02	3-10	1	—	—	—	—
Seqno_DTM	15-18	T	08	14-17*	1	—	—	—	—
Supplemental_Dist_Info1	15-19	T	09	3-10	0-1	—	—	—	—
Agent_Correl	15-19	T	0A	3-130	0-1	—	—	—	—
Report-To_DSU	15-20	T	0B	8-22	0-1	N	Y	2	—
Report-To_RGN	15-20	T	01	3-10	1	—	—	—	—
Report-To_REN	15-20	T	02	3-10	1	—	—	—	—
Report-To_User	15-20	T	0C	8-22	0-1	N	Y	2	—
Report-To_DGN	15-21	T	01	3-10	1	—	—	—	—
Report-To_DEN	15-21	T	02	3-10	1	—	—	—	—
Report_Service_Parms	15-22	T	0D	5-32	0-1	—	—	—	—
Report-To_Agent	15-24	T	0E	3-10	0-1	—	—	—	—
Dest_Agent	15-25	T	0F	3-10	0-1	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-3728	—	—	—	—	—
Dest_List	15-25	ID/seg	1572	12-11268	1	N	Y	1	—
Dest	15-25	Imp-T	idc	8-5654	≥1	N	Y	1-2	—
Dest_DSU	15-25	T/idc	01	8-22	1	N	Y	2	—
Dest_RGN	15-26	T	01	3-10	1	—	—	—	—
Dest_REN	15-26	T	02	3-10	1	—	—	—	—

Figure 15-1 (Page 2 of 2). Distribution Transport Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dest_User	15-26	T	02	8-22	≥0	N	Y	2	—
Dest_DGN	15-26	T	01	3-10	1	—	—	—	—
Dest_DEN	15-27	T	02	3-10	1	—	—	—	—
Agent_Object	15-27	ID/seg	1573	5-32767	0-1	—	—	—	—
Server_Object	15-27	ID/seg	1574	≥5	0-1	—	—	—	—
Supplemental_Dist_Info2	15-27	ID/seg	1580	5-32767	0-1	—	—	—	—
Unrecognized_Reserve	15-39	ID/seg	—	4-32767	—	—	—	—	—
DS_Suffix	15-27	ID/sfx	157F	4	1	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-11 for presence rules and length restrictions.

DISTRIBUTION REPORT MESSAGE UNIT (DRMU)

Figure 15-2 (Page 1 of 2). Distribution Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Report_MU	15-27	Del-ID	pfx	≥77*	1	Y	Y	≥6	—
Report_Prefix	15-28	ID/pfx	157C	8-18	1	N	Y	1-3	—
Hop_Count	15-11	T	01	4	1	—	—	—	—
MU_ID	15-11	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-11	T	06	4	0-1*	—	—	—	—
Report_Command	15-28	ID/seg	1575	25-4096*	1	Y	Y	≥3	—
Service_Parms	15-13	T	02	5-32	0-1	—	—	—	—
Report-To_Agent	15-24	T	04	3-10	1	—	—	—	—
Reporting_DSU	15-28	T	06	8-22	1	N	Y	2	—
Reporting_RGN	15-28	T	01	3-10	1	—	—	—	—
Reporting_REN	15-28	T	02	3-10	1	—	—	—	—
Report_DTM	15-28	T	09	10-13*	1	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-4015	—	—	—	—	—
Report-To_DSU_User	15-29	ID	1583	12-48	1	N	Y	1-2	—
Report-To_DSU	15-20	T	01	8-22	1	N	Y	2	—
Report-To_RGN	15-20	T	01	3-10	1	—	—	—	—
Report-To_REN	15-20	T	02	3-10	1	—	—	—	—
Report-To_User	15-20	T	02	8-22	0-1	N	Y	2	—
Report-To_DGN	15-21	T	01	3-10	1	—	—	—	—
Report-To_DEN	15-21	T	02	3-10	1	—	—	—	—
Report_Information	15-29	ID/seg	1576	18-4096	1	Y	Y	≥1	—
Reported-On_Origin_DSU	15-30	T	06	8-22	0-1*	N	Y	2	—
Reported-On_Origin_RGN	15-30	T	01	3-10	1	—	—	—	—
Reported-On_Origin_REN	15-30	T	02	3-10	1	—	—	—	—
Reported-On_Origin_User	15-30	T	07	8-22	0-1*	N	Y	2	—
Reported-On_Origin_DGN	15-31	T	01	3-10	1	—	—	—	—
Reported-On_Origin_DEN	15-31	T	02	3-10	1	—	—	—	—
Reported-On_Seqno_DTM	15-31	T	08	14-17	1	—	—	—	—
Reported-On_Supp_Dist_Info1	15-32	T	09	3- 10	0-1	—	—	—	—
Reported-On_Agent_Correl	15-32	T	0A	3-130	0-1	—	—	—	—
Reported-On_Origin_Agent	15-33	T	0B	3-10	0-1*	—	—	—	—
Reported-On_Dest_Agent	15-33	T	0C	3-10	0-1*	—	—	—	—
Receiving_DSU	15-34	T	10	8-22	0-1	N	Y	2	—
Receiving_RGN	15-35	T	01	3-10	1	—	—	—	—
Receiving_REN	15-35	T	02	3-10	1	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-3849	—	—	—	—	—
SNA_Condition_Report	* *	ID/seg	1532	10-32767	1	Y	Y	≥1	* *
Reported-On_Supp_Dist_Info2	15-33	ID/seg	1582	5-32767	0-1*	—	—	—	—
Unrecognized_Reserve	15-39	ID/seg	—	4-32767	—	—	—	—	—

Figure 15-2 (Page 2 of 2). Distribution Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
DS_Suffix	15-27	ID/sfx	157F	4	1	—	—	—	—

Notes:

- * Refer to FS2 Structure Descriptions starting on page 15-11 for presence rules and length restrictions.
- * * Refer to Appendix B, "Common Structures."

DISTRIBUTION CONTINUATION MESSAGE UNIT (DCMU)

Figure 15-3. Distribution Continuation Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Dist_Continuation_MU	15-33	Del-ID	pfx	≥18	1	Y	Y	≥2	—
Continuation_Prefix	15-33	ID/pfx	157B	14-24	1	N	Y	2-3	—
MU_ID	15-11	T	03	6	1	—	—	—	—
MU_Instance_Number	15-11	T	06	4	1	—	—	—	—
Restarting_Byte_Position	15-34	T	02	10	0-1	—	—	—	—
Agent_Object	15-27	ID/seg	1573	5-32767	0-1	—	—	—	—
Server_Object	15-27	ID/seg	1574	≥5	0-1	—	—	—	—
Supplemental_Dist_Info2	15-27	ID/seg	1580	5-32767	0-1	—	—	—	—
Unrecognized_Reserve	15-39	ID/seg	—	4-32767	—	—	—	—	—
DS_Suffix	15-27	ID/sfx	157F	4	1	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-11 for presence rules.

SENDER EXCEPTION MESSAGE UNIT (SEMU)

Figure 15-4. Sender Exception Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Sender_Exception_MU	15-34	ID	1578	10-256	1	Y	Y	≥1	—
SNA_Report_Code	* *	T	7D	6	1	—	—	—	—
MU_ID	15-11	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-11	T	06	4	0-1*	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-236	—	—	—	—	—

Notes:

- * Refer to FS2 Structure Descriptions starting on page 15-11 for presence rules.
- * * Refer to Appendix B, "Common Structures."

RECEIVER EXCEPTION MESSAGE UNIT (REMU)

Figure 15-5. Receiver Exception Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Receiver_Exception_MU	15-34	Del-ID	pfx	≥25	1	Y	Y	≥2	—
Receiver_Exception_Command	15-34	ID/pfx	1577	15-512	1	Y	Y	≥2	—
Sender_Retry_Action	15-34	T	01	3	1	—	—	—	—
MU_ID	15-11	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-11	T	06	4	0-1*	—	—	—	—
Receiving_DSU	15-34	T	16	8-22	1	N	Y	2	—
Receiving_RGN	15-35	T	01	3-10	1	—	—	—	—
Receiving_REN	15-35	T	02	3-10	1	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-473	—	—	—	—	—
Unrecognized_Reserve	15-39	ID	—	≥4	—	—	—	—	—
SNA_Condition_Report	* *	ID/sfx	1532	10-1024	1	Y	Y	≥1	* *

Notes:

- * Refer to FS2 Structure Descriptions starting on page 15-11 for presence rules.
- * * Refer to Appendix B, “Common Structures.”

COMPLETION QUERY MESSAGE UNIT (CQMU)

Figure 15-6. Completion Query Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Completion_Query_MU	15-35	ID	1579	14-256	1	Y	Y	≥2	—
MU_ID	15-11	T	03	6	1	—	—	—	—
MU_Instance_Number	15-11	T	06	4	1	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-242	—	—	—	—	—

COMPLETION REPORT MESSAGE UNIT (CRMU)

Figure 15-7. Completion Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Completion_Report_MU	15-35	ID	157A	7-256	1	Y	Y	≥1	—
Indicator_Flags	15-35	T	01	3	1	—	—	—	—
MU_ID	15-11	T	03	6	0-1	—	—	—	—
MU_Instance_Number	15-11	T	06	4	0-1*	—	—	—	—
Last_Structure_Received	15-36	T	04	4	0-1*	—	—	—	—
Last_Byte_Received	15-36	T	05	10	0-1*	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-225	—	—	—	—	—

Note: * Refer to FS2 Structure Descriptions starting on page 15-11 for presence rules.

PURGE REPORT MESSAGE UNIT (PRMU)

Figure 15-8. Purge Report Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Purge_Report_MU	15-36	ID	157E	10-256	1	Y	Y	≥1	—
MU_ID	15-11	T	03	6	1	—	—	—	—
Unrecognized_Reserve	15-39	T	—	2-246	—	—	—	—	—

RESET REQUEST MESSAGE UNIT (RRMU)

Figure 15-9. Reset Request Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Reset_Request_MU	15-37	ID	1585	21-4096	1	—	Y	≥2	—
MU_ID	15-11	T	03	6	1	—	—	—	—
Reset_DTM	15-37	T	09	11-13	1	—	—	—	—

RESET ACCEPTED MESSAGE UNIT (RAMU)

Figure 15-10. Reset Accepted Message Unit

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
Reset_Accepted_MU	15-38	ID	1586	21-4096	1	—	Y	≥2	—
MU_ID	15-11	T	03	6	1	—	—	—	—
Reset_DTM	15-37	T	09	11-13	1	—	—	—	—

FS2 Structure Descriptions

Dist_Transport_MU

Description: The *distribution_transport_message_unit* transports agent and/or server objects for distribution to one or more users or application programs.

Length Restriction: The minimum length of a *dist_transport_MU* originated by an FS2 DSU is 54 bytes. This is due to the length restriction on the *Seqno_DTM*.

Transport_Prefix

Description: The *transport_prefix* identifies the beginning of the *dist_transport_MU*. This structure carries information that changes from DSU to DSU.

Hop_Count

Description: The *hop_count* is the remaining number of hops that may be traversed by a SNA/DS distribution on its way toward its destination DSUs. The *hop_count* is set by the origin DSU in the DTMUs and by the reporting DSUs for the DRMUs. The *hop_count* is decremented by 1 in every DSU through which the distribution passes. If the *hop_count* reaches 0 at an intermediate DSU, exception processing is invoked.

Format: Signed binary integer

Range of Values: Valid values range from 1 to 2^{15}-1.

MU_ID

Description: The *message_unit_identifier* is a number that uniquely identifies a distribution MU throughout its existence. An MU exists for only one hop, from one DSU to the adjacent DSU. In REMUs and SEMUs, the *MU_ID* refers to a distribution MU. An *MU_ID* is unique only for a particular *LU name*, *mode name* combination.

Presence Rule: If the *MU_ID* is absent, exception reporting may not be requested.

Format: Signed binary integer

Range of Values: Valid values range from 1 to 2^{31}-1.

MU_Instance_Number

Description: The *message_unit_instance_number* identifies the instance of a particular distribution message unit and its corresponding *MU_ID*.

Presence Rule: Precluded if an *MU_ID* is not present; otherwise, required.

Format: Signed binary integer

Range of Values: Valid values range from 1 to 2^{15}-1.

Transport_Command

Description:	The *transport_command* contains the control information used by the distribution service to transport the distribution.
Length Restriction:	The minimum length of a *transport_command* originated by an FS2 DSU is 30 bytes. This is due to the length restriction on the *seqno_DTM*.

Dist_Flags

Description:	The *distribution_flags* indicate services requested by the origin agent.
Note:	If exception reporting is requested, the *MU_ID* is always present.
Format:	Bit string

Byte	Bit	Content
0-1		LT header
2		Flags (bits 0-7) that must be understood and honored by all DSUs
	0	Exception report flag indicating whether an exception report is to be sent if the distribution is aborted: 0 no exception report to be sent (default) 1 exception report to be sent
	1 -7	Reserved
3		Flags (bits 0-7) that must be understood and honored by destination DSUs, but that can be ignored by intermediate DSUs
	0 -7	Reserved
4		Flags (bits 0-7) that are ignored by DSUs if not understood
	0- 7	Reserved

Service_Parms

Description: The *service_parameters* structure describes the types and levels of service requested for the distribution. The parameters in this structure are provided by the origin agent. The *service_parameters* used in the DTMU and the DRMU are similar; the differences in such usage and the default values used for absent *service_parameter* (SP) triplets are discussed under the individual triplets below. The default values specified below are assumed for absent *service_parameter* (SP) triplets. The specification for deriving the *service_parameters* for the DRMU is found in the description of *report_service_parameters* on page 15-22.

Format: Special format consisting of ordered, optional, SP triplets of the following general structure:

Byte	Bit	Content
0		Parameter type: All parameter type byte values less than X'80' are defined by or reserved for SNA/DS. All parameter type byte values greater than X'80' may be customer defined.
1		Comparison operator:
	0-3	1100 REQUIRE_LEVEL_GE 1110 REQUIRE_SUPPORT_FOR Note: All other values for bits 0-3 are reserved.
	4-7	Reserved
2		Value: The meaning of this byte depends on the parameter type.

Byte	Content
0-1	LT header
2-31	Up to 10 different *service_parameter* (SP) triplets may be carried in one distribution. Each triplet, when present, appears in ascending sequence of parameter type. The capacity triplet is not used in the DRMU. All FS2 service parameters are optional in both the DTMU and the DRMU. The architecturally defined service parameters are given below:

Priority SP Triplet

Byte	Content
0	X′01′
1	X′C0′ REQUIRE_LEVEL_GE
2	X′F0′ FAST (default) X′D0′ CONTROL X′80′ DATA_16 (can be treated as DATAHI) X′78′ DATA_15 (can be treated as DATAHI) X′70′ DATA_14 (can be treated as DATAHI) X′68′ DATA_13 (can be treated as DATAHI) X′60′ DATA_12 (DATAHI) X′58′ DATA_11 (can be treated as DATAHI) X′50′ DATA_10 (can be treated as DATAHI) X′48′ DATA_9 (can be treated as DATAHI) X′40′ DATA_8 (can be treated as DATALO) X′38′ DATA_7 (can be treated as DATALO) X′30′ DATA_6 (can be treated as DATALO) X′28′ DATA_5 (can be treated as DATALO) X′20′ DATA_4 (DATALO) X′18′ DATA_3 (can be treated as DATALO) X′10′ DATA_2 (can be treated as DATALO) X′08′ DATA_1 (can be treated as DATALO) Note: All other values are reserved.

Notes:

1. The Priority SP value X′D0′ (CONTROL) occurs in a DRMU only.
2. The Priority SP range for DATALO is X′01′ to X′40′. The Priority SP range for DATAHI is X′41′ to X′80′.
3. Implementations may accept other priority levels as long as they can route the distribution responsibly.

Protection SP Triplet

Byte	Content
0	X′02′
1	X′C0′ REQUIRE_LEVEL_GE
2	X′10′ LEVEL1 (default when Priority SP is GE X′E0′): safe store may be performed. X′30′ LEVEL2 (default when Priority SP is LT X′E0′): safe store must be performed. Note: All other values are reserved.

Capacity SP Triplet

Byte	Content
0	X′03′
1	X′C0′ REQUIRE_LEVEL_GE
2	Capacity value is the exponent of the power of 2 that represents the value of the required capacity for the *server_object* in the DTMU: X′00′ ZERO (default when Priority SP is GE X′E0′) used if there is no *server_object* in *dist_transport_MU*. X′14′ 1MB X′16′ 4MB X′18′ 16MB (default when Priority SP is LT X′E0′) Note: All other values are reserved.

Notes:

1. The Capacity SP triplet occurs only in a DTMU.
2. Receiving FS2 DSUs are always able to receive a capacity level of INDEFINITE (designated by X′E0FF′ in bytes 1-2). Originating FS2 DSUs never generate the capacity level of INDEFINITE. The level replacing INDEFINITE is 16MB (X′C018′).
3. The capacity requirement is for the *server_object*, and does not include the capacity needed to store and handle the other structures of the DTMU.
4. Implementations may accept other capacity levels as long as they can route the distribution responsibly.

Security SP Triplet

Byte	Content
0	X′04′
1	X′C0′ REQUIRE_LEVEL_GE
2	X′01′ LEVEL1 (default): security is not required. X′20′ LEVEL2: security is required. Note: All other values are reserved.

Acceptable Delay SP Triplet

Byte	Content
0	X′05′
1	X′A0′ REQUIRE_LEVEL_LE
2	X′FF′ INDEFINITE Note: All other values are reserved.

Server_Obj_Byte_Count

Description:	The *server_object_byte_count* is the number of bytes of all the segments of the *server_object*. An FS2-capable DSU originating a distribution either supplies a correct byte count, or omits the field completely.
Presence Rule:	Optional when the *server_object* is present; otherwise, precluded.
Format:	Unsigned binary integer
Range of Values:	Valid values range from 1 to 2^{64}-2.

Origin_Agent

Description:	The *origin_agent* is the transaction program at the DSU at which the distribution originated.
Format:	Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Server

Description: The *server* is the name to be used to store the *server_object* at the destination.

Presence Rule: Optional when the *server_object* is present; otherwise, precluded. If optional and absent, the general server TP name is the default.

Format: Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Origin_DSU

Description: The *origin_DSU* is the name of the DSU at which the distribution originated.

Origin_RGN

Description: The *origin_RGN* is the first part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Origin_REN

Description: The *origin_REN* is the second part of the name of the DSU at which the distribution originated. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Origin_User

Description: The *origin_user* is the user name of the originator of the distribution.

Origin_DGN

Description: The *origin_DGN* is the first part of the user name of the distribution originator.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Origin_DEN

Description: The *origin_DEN* is the second part of the user name of the distribution originator.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Seqno_DTM

Description: The *sequence_number/date-time*, in combination with the *origin_agent*, *origin_user*, and *origin_DSU*, uniquely identifies the distribution. The sequence number is the number assigned to the distribution by the origin agent. The date of the distribution is assigned by the origin agent; the time of the distribution is assigned by the origin DSU. The offset from local time to GMT is included.

Length Restriction: Originating FS2 DSUs never generate a local-only time. The minimum length for *seqno_DTM* is therefore 15 bytes (including its header).

Format: Byte string

Range of Values: Valid values for the sequence number portion of the *seqno_DTM* range from 1 to 2^{31}-1. FS2 tolerates sequence numbers with value 0. However, sequence numbers with value 0 are never originated from within an FS2 network.

Byte	Content
0-1	LT header
	SEQNO
2-5	Signed binary integer ranging from 1 to 2^{31}-1
	DATE
6-7	Year, in binary (e.g., 1989 is encoded as X′07C5′)
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 13 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
14	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X′E9′ (Z) TIME is GMT and no offset required. X′4E′ (+) TIME is local, OFFSET VALUE is required, and encoder′s local time is ahead of GMT. X′60′ (-) TIME is local, OFFSET VALUE is required, and encoder′s local time trails GMT. Note: All other values are reserved.
	OFFSET VALUE
15	Hour offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 13 are valid)
16	Minute offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 59 are valid)

Note: Refer to “Representing Date and Time” on page 15-40 for a complete discussion of the encoding and interpretation of date and time.

Supplemental_Dist_Info1

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

Format: Undefined byte string

Agent_Correl

Description: The *agent_correlation* is a string supplied by the origin agent. SNA/DS is not aware of its contents.

Format: Undefined byte string

Report-To_DSU

Description: The *report-to_DSU* is the name of the DSU to which distribution reports are to be sent. If both *report-to_DSU* and *report-to_user* are absent in the DTMU, the values generated in the DRMU for these structures default to the origin. If only *report-to_DSU* is present in the DTMU, then any report is sent to that DSU. If only *report-to_user* is present in the DTMU, then the reporting DSU will refer to its directory to determine *report-to_DSU*.

Report-To_RGN

Description: The *report-to_RGN* is the first part of the DSU name to which distribution reports are to be sent. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Report-To_REN

Description: The *report-to_REN* is the second part of the DSU name to which distribution reports are to be sent. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. If a product chooses to implement DGN=REN, the enhanced character string (ECS) subset is implemented in a particular network, and a DGN exists that contains an ECS character that is not an element of CGCSGID 01134-0500, then ECS characters may occur in this structure.

Report-To_User

Description: The *report-to_user* is the name of the user to which distribution reports are to be sent. If both *report-to_user* and *report-to_DSU* are absent in the DTMU, the values generated in the DRMU for these structures default to the origin. If only *report-to_user* is present in the DTMU, the reporting DSU refers to its directory to determine *report-to_DSU*.

Report-To_DGN

Description: The *report-to_DGN* is the first part of the user name to which distribution reports are to be sent.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report-To_DEN

Description: The *report-to_DEN* is the second part of the user name to which distribution reports are to be sent.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Report_Service_Parms

Description: The *report_service_parameters* structure describes the service requested for the distribution report by the origin agent when the agent wants to override the *service_parameters* that would be routinely generated by the reporting DSU for the report MU. If *report_service_parameters* are specified, they are used as the *service_parameters* in any DRMUs that are generated as part of the distribution. If the origin agent does not specify one or more of the *report_service_parameters*, a DSU that generates a report derives appropriate *service_parameters* for the DRMU from the *service_parameters* in the DTMU.

The comparison operators and values derived for the protection, security, and acceptable delay parameters are the same as those specified (explicitly or implicitly) in the DTMU.

For the priority service parameter, the value derived is either FAST or CONTROL. FAST is used if the DTMU specified FAST priority; CONTROL is used if the DTMU specified a DATA_N priority. CONTROL priority is used only in DRMUs; it may not be specified for the priority service parameter in a DTMU. If the origin agent explicitly specifies a value for the priority report service parameter, the value may be FAST, CONTROL, or DATA_N. The comparison operator for the priority service parameter is always REQUIRE_LEVEL_GE.

Format: Special format consisting of ordered, optional *report_service_parameter* triplets of the same general structure as for *service_parameters*. See *service_parameters* on page 15-13.

Byte	Content
0-1	LT header
2-31	Up to 10 different *report_service_parameter* (RSP) triplets may be carried in one distribution. Each triplet, when present, appears in ascending sequence of parameter type. The capacity triplet is not used in the DRMU, and therefore the capacity RSP is never specified. All service parameters are optional in both the DTMU and the DRMU.

Priority RSP Triplet

Byte	Content
0	X'01'
1	X'C0' REQUIRE_LEVEL_GE
2	X'F0' FAST X'D0' CONTROL X'80' DATA_16 (can be treated as DATAHI) X'78' DATA_15 (can be treated as DATAHI) X'70' DATA_14 (can be treated as DATAHI) X'68' DATA_13 (can be treated as DATAHI) X'60' DATA_12 (DATAHI) X'58' DATA_11 (can be treated as DATAHI) X'50' DATA_10 (can be treated as DATAHI) X'48' DATA_9 (can be treated as DATAHI) X'40' DATA_8 (can be treated as DATALO) X'38' DATA_7 (can be treated as DATALO) X'30' DATA_6 (can be treated as DATALO) X'28' DATA_5 (can be treated as DATALO) X'20' DATA_4 (DATALO) X'18' DATA_3 (can be treated as DATALO) X'10' DATA_2 (can be treated as DATALO) X'08' DATA_1 (can be treated as DATALO) Note: All other values are reserved.

Protection RSP Triplet

Byte	Content
0	X'02'
1	X'C0' REQUIRE_LEVEL_GE
2	X'10' LEVEL1: safe store may be performed. X'30' LEVEL2: safe store must be performed. Note: All other values are reserved.

Security RSP Triplet

Byte	Content
0	X′04′
1	X′C0′ REQUIRE_LEVEL_GE
2	X′01′ LEVEL1: security is not required. X′20′ LEVEL2: security is required. Note: All other values are reserved.

Acceptable Delay RSP Triplet

Byte	Content
0	X′05′
1	X′A0′ REQUIRE_LEVEL_LE
2	X′FF′ INDEFINITE Note: All other values are reserved.

Report-To_Agent

Description:	The *report-to_agent* is the name of the application transaction program to be started after the report is queued for delivery. If *report-to_agent* is absent in the DTMU, the value specified in the DTMU for *origin_agent* is used in the DRMU for *report-to_agent*.
Format:	Character string, except for first byte.

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Dest_Agent

Description: The *destination_agent* is the transaction program at the destination DSU to which the distribution is to be delivered. If *dest_agent* is absent in the DTMU, the value specified for *origin_agent* is assumed to be the *dest_agent*.

Format: Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Dest_List

Description: The *destination_list* is the list of destinations for the distribution, which can contain up to 256 destinations. Each destination is a *dest_DSU* with or without a *dest_user*, expressed as (*dest_DSU* (,*dest_user*)). For single-destination distributions and distribution reports, the *dest_list* contains only one destination.

Either a flat destination list, of the form

(*dest_DSU* (*dest_user*)), ..., (*dest_DSU* (*dest_user*)), ...

or a factored destination list, of the form

(*dest_DSU* (*dest_user*, *dest_user*, ...)), (*dest_DSU* (*dest_user*, ...))

may be present. For example, a flat destination list might contain

(DSU_A USER_1), (DSU_A USER_2), (DSU_A), (DSU_B USER_3), (DSU_B USER_4)

whereas a factored destination list would contain

(DSU_A (USER_1, USER_2)), (DSU_A), (DSU_B (USER_3, USER_4)).

Dest

Description: The *destination* associates *dest_users* with a *dest_DSU*. For flat destination lists, there are zero or one user names per *dest*. For factored destination lists, there can be multiple user names per *dest*.

Dest_DSU

Description: The *destination_DSU* is the name of one of the DSUs to which the distribution is to be sent.

Dest_RGN

Description: The *destination_RGN* is the first part of a *dest_DSU* name. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Dest_REN

Description: The *destination_REN* is the second part of a *dest_DSU* name. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. If a product chooses to implement DGN=REN, the enhanced character string (ECS) subset is implemented in a particular network, and a DGN exists that contains an ECS character that is not an element of CGCSGID 01134-0500, then ECS characters may occur in this structure.

Dest_User

Description: The *destination_user* is the name of one of the users to which the distribution is to be sent.

Dest_DGN

Description: The *destination_DGN* is the first part of the name of a *dest_user*.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Dest_DEN

Description: The *destination_DEN* is the second part of the name of a *dest_user*.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Agent_Object

Description: The *agent_object* is directly supplied by the origin agent. It is never parsed by the distribution service and is directly delivered, unchanged, to the agent at each destination.

Format: Undefined byte string

Server_Object

Description: The *server_object* is identified by the origin agent and is fetched by the origin server when sending the *dist_transport_MU*. At each destination, the *server_object* is stored by the destination server and a notification of its receipt is delivered to the destination agent.

Format: Undefined byte string

Supplemental_Dist_Info2

Description: The *supplemental_dist_info2* structure is reserved for future use.

Format: Undefined byte string

DS_Suffix

Description: The *distribution_services_suffix* contains no information and marks the end of the *dist_transport_MU*, *dist_report_MU*, or *dist_continuation_MU*.

Dist_Report_MU

Description: The *distribution_report_message_unit* carries information reporting on the state of the distribution. Typically, for a multiple destination distribution, a *dist_report_MU* will report on only a portion of the distribution. The report is delivered to the report-to destination if one was specified in the reported-on DTMU; otherwise, it is delivered to the distribution originator.

Length Restriction: The minimum length of a *dist_report_MU* originated by an FS2 DSU is 78 bytes. This is due to the length restriction on the *Report_DTM*.

Report_Prefix

Description: The *report_prefix* identifies the beginning of *dist_report_MU*. This structure carries information that changes from DSU to DSU.

Report_Command

Description: The *report_command* contains the control information for the distribution report.

Length Restriction: The minimum length of a *dist_report_MU* originated by an FS2 DSU is 26 bytes (including its header). This is due to the length restriction on the *report_DTM*.

Reporting_DSU

Description: The *reporting_DSU* is the name of the DSU that generated the report.

Reporting_RGN

Description: The *reporting_RGN* is the first part of the name of the DSU that generated the report. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Reporting_REN

Description: The *reporting_REN* is the second part of the name of the DSU that generated the report. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Report_DTM

Description: The *report_date-time* contains the date and time at which the reporting DSU generated the report. FS2 implementations support the offset from local time to GMT.

Length Restriction: Originating FS2 DSUs never generate a local-only time (implying a minimum length of 11 bytes - including its header). However, if the value within Report_DTM had been mapped from an FS1 subnetwork, it would have a length of 10 bytes (including its header).

Format: Byte string

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X′07C5′)
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
10	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X′E9′ (Z) TIME is GMT and no offset required. X′4E′ (+) TIME is local, OFFSET VALUE is required, and encoder′s local time is ahead of GMT. X′60′ (-) TIME is local, OFFSET VALUE is required, and encoder′s local time trails GMT. Note: All other values are reserved.
	OFFSET VALUE
11	Hour offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 13 are valid)
12	Minute offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 59 are valid)

Note: Refer to “Representing Date and Time” on page 15-40 for a complete discussion of the encoding and interpretation of date and time.

Report-To_DSU_User

Description: The *report-to_DSU_user* is the DSU or user to which the distribution report is being sent.

Report_Information

Description: The *report_information* identifies the distribution (or portion thereof) being reported on.

Reported-On_Origin_DSU

Description: The *reported-on_origin_DSU* is the name of the DSU at which the distribution was originated.

Presence Rules: If *reported-on_origin_DSU* is present, and *reported-on_origin_user* is absent, then the distribution was originated by a DSU; if *reported-on_origin_user* is present and *reported-on_DSU* is absent, then the report either originated in or passed through an FS1 subnetwork. If both *reported-on_origin_DSU* and *reported-on_origin_user* are present, then the report is not going to the originator of the distribution; if both *reported-on_origin_DSU* and *reported-on_origin_user* are absent, then they default to *report-to_DSU* and, if applicable, *report-to_user*.

Reported-On_Origin_RGN

Description: The *reported-on_origin_RGN* is the first part of the DSU name at which the distribution originated. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Reported-On_Origin_REN

Description: The *reported-on_origin_REN* is the second part of the DSU name at which the distribution originated. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Reported-On_Origin_User

Description: The *reported-on_origin_user* is the name of the user that originated the distribution.

Presence Rules: If *reported-on_origin_DSU* is present, and *reported-on_origin_user* is absent, then the distribution was originated by a DSU; if *reported-on_origin_user* is present and *reported-on_DSU* is absent, then the report either originated in or passed through an FS1 subnetwork. If both *reported-on_origin_DSU* and *reported-on_origin_user* are present, then the report is not going to the originator of the distribution; if both *reported-on_origin_DSU* and *reported-on_origin_user* are absent, then they default to *report-to_DSU* and, if applicable, *report-to_user*.

Reported-On_Origin_DGN

Description: The *reported-on_origin_DGN* is the first part of the name of the user that originated the distribution.

Format: Character string

Support Option	CGCSGID	String Conventions
Base	01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.
Enhanced Character Strings	00930-00500	Leading space (X′40′) characters are not allowed, trailing space characters are not significant, and imbedded space characters are significant.

Reported-On_Origin_DEN

Description: The *reported-on_origin_DEN* is the second part of the name of the user that originated the distribution.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Reported-On_Seqno_DTM

Description: The *reported-on_sequence_number/date-time*, in combination with the origin agent, origin DSU, and origin user, is the unique identifier of the distribution. The origin agent, origin DSU, and origin user are specified in the appropriate reported-on or report-to structures. The sequence number is the number assigned to the distribution by the origin agent. The date-time is the date and time generated at the origin of the distribution. FS2 implementations support the offset from local time to GMT.

Length Restriction: Originating FS2 DSUs never generate a local-only time. The minimum length for *reported-on_seqno_DTM* is therefore 15 bytes (including its header).

Format: Byte string

Range Of Values: Valid values for sequence number portion of the *reported-on_seqno_DTM* range from 1 to 2^{31}-1.

Byte	Content
0-1	LT header
	SEQNO
2-5	Signed binary integer ranging from 1 to 2^{31}-1
	DATE
6-7	Year, in binary (e.g., 1989 is encoded as X′07C5′)
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 23 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
14	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X′E9′ (Z) TIME is GMT and no offset required. X′4E′ (+) TIME is local, OFFSET VALUE is required, and encoder′s local time is ahead of GMT. X′60′ (-) TIME is local, OFFSET VALUE is required, and encoder′s local time trails GMT. Note: All other values are reserved.
	OFFSET VALUE
15	Hour offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 13 are valid)
16	Minute offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 59 are valid)

Note: Refer to “Representing Date and Time” on page 15-40 for a complete discussion of the encoding and interpretation of date and time.

Reported-On_Supp_Dist_Info1

Description: The *reported-on_supp_dist_info1* structure is reserved for future use.

Format: Character string

Reported-On_Agent_Correl

Description: The *reported-on_agent_correlation* is a string that was supplied by the origin agent at the origin DSU.

Format: Undefined byte string

Reported-On_Origin_Agent

Description: The *reported-on_origin_agent* is the name of the transaction program at the origin DSU that originated the distribution that is being reported on.

Presence Rule: Occurs when *report-to_agent* is different from *origin_agent*. If third-party reporting has been requested and a report was generated in or flowed through an FS1 subnetwork, the *reported-on_origin_agent* structure is discarded.

Format: Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Reported-On_Dest_Agent

Description: The *reported-on_destination_agent* is the name of the transaction program at the destination DSU that was specified for the reported-on distribution.

Presence Rule: Occurs when *dest_agent* was specified in the reported-on DTMU.

Format: Character string, except for first byte

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00′ to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA registered. X′40′ is not a valid first-byte value.

Reported-On_Supp_Dist_Info2

Description: The *reported-on_supp_dist_info2* structure is reserved for future use.

Format: Undefined byte string

Dist_Continuation_MU

Description: The *distribution_continuation_message_unit* is used by a sending DSU to continue transmission of a suspended MU.

Continuation_Prefix

Description: The *continuation_prefix* identifies the beginning of a DCMU.

Restarting_Byte_Position

Description: The *restarting_byte_position* indicates where the sender is beginning retransmission of the first structure being re-sent. The byte count begins with the first byte of atomic data (i.e., no LLs included) within the encompassing structure. Absence of this structure is equivalent to the presence of a 1 in this structure, implying that the first structure present in the DCMU is being re-sent in its entirety. 0 is not allowed.

Format: Unsigned binary integer

Range of Values: Valid values range from 1 to 2^{64}-2.

Sender_Exception_MU

Description: The *sender_exception_MU* is sent from the sender to the receiver when the sender detects an exception while sending a *dist_transport_MU*, a *dist_report_MU*, or a *dist_continuation_MU*.

Receiver_Exception_MU

Description: The *receiver_exception_MU* is sent from the receiver to the sender when the receiver detects an exception while receiving a *dist_transport_MU*, a *dist_report_MU*, or a *dist_continuation_MU*.

Receiver_Exception_Command

Description: The *receiver_exception_command* is the prefix identifying the *receiver_exception_MU*.

Sender_Retry_Action

Description: The *sender_retry_action* is the receiver's recommendation to the sender as to whether to retry the transmission of the MU.

Format: Hexadecimal code

Byte	Content
0-1	LT header
2	X'01' RETRY_PRECLUDED X'02' RETRY_ALLOWED X'03' RETRY_EXPECTED_USING_DCMU Note: All other values are reserved.

Receiving_DSU

Description: The *receiving_DSU* is the name of the DSU to which a distribution was being sent.

Receiving_RGN

Description: The *receiving_RGN* is the first part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Receiving_REN

Description: The *receiving_REN* is the second part of the name of the DSU to which a distribution was being sent. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID	String Conventions
01134-00500	Leading, imbedded, and trailing space (X′40′) characters are not allowed. If a product chooses to implement DGN=REN, the enhanced character string (ECS) subset is implemented in a particular network, and a DGN exists that contains an ECS character that is not an element of CGCSGID 01134-0500, then ECS characters may occur in this structure.

Completion_Query_MU

Description: The *completion_query_message_unit* is sent by the sending DSU to query the completion status of a particular MU at the receiving DSU.

Completion_Report_MU

Description: The *completion_report_message_unit* is sent by the receiving DSU to report on the completion status of a particular MU or to control traffic flow on a conversation.

Indicator_Flags

Description: The *indicator_flags* structure contains a 1-byte flag, to indicate the completion status of the *MU_ID* identified in a *completion_report_MU*, or to control traffic flow on a conversation.

Format: Bit string

Note: Conversation control flags (bits 2 and 3) may be used in conjunction with flow control flags (Not Received, In Transit, Suspended, Terminated, Completed, Purged).

Bit Map								Architecturally-Defined Value
0	1	2	3	4	5	6	7	
x	x	0	0	x	x	x	x	Default—Normal SNA/DS flow
x	x	0	1	x	x	x	x	Terminate Conversation
0	x	x	x	0	0	0	0	Not Received
0	x	x	x	0	0	0	1	In Transit
0	x	x	x	0	0	1	0	Suspended
0	x	x	x	0	0	1	1	Completed
0	x	x	x	0	1	0	1	Terminated
1	x	x	x	x	x	x	x	Purged

Note: x = any value

Last_Structure_Received

Description: The *last_structure_received* is the codepoint of the structure the receiving DSU identifies as the last structure received before the MU was suspended. This structure must be a length-bounded LLID structure at the highest level of the MU.

Presence Rule: If *indicator_flags* = SUSPENDED, then *last_structure_received* is present.

Format: Hexadecimal code

Last_Byte_Received

Description: The *last_byte_received* is the last byte received by the receiving DSU before the MU was suspended. The byte count begins with the first byte of atomic data within the encompassing structure. The byte count contains only atomic data and does not contain the segmenting LLs for segmented structures. A byte count of X′0000000000000000′ indicates that only the LLID of the structure was received (i.e., that any following atomic data was either not received or lost). A byte count of X′FFFFFFFFFFFFFFFF′ indicates that the structure was fully received.

Presence Rules: If *indicator_flags* = SUSPENDED, *last_structure_received* is present, and *last_byte_received* is absent, then the structure was received.

Format: Unsigned binary integer

Range of Values: Valid values range from 0 to 2^{64}-1, where the values 0 and 2^{64}-1 have the meanings defined above.

Purge_Report_MU

Description: The *purge_report_message_unit* indicates to the receiving DSU that the sending DSU has marked a particular *MU_ID* PURGED, and that the receiving DSU may flag that *MU_ID* as PURGED.

Reset_Request_MU

Description: The *reset_request_message_unit* is sent from DS_Send to DS_Receive. DS_Send issues the *reset_request_MU* to request that DS_Receive reset its *MU_ID* registry.

Reset_DTM

Description: The *reset_date-time* contains the date and time at which the *reset_request_MU* was generated. Both sender and receiver store it as the ″time of last reset″ of their *MU_ID* registries.

Length Restriction: Originating FS2 DSUs never generate a local-only time. The minimum length for *reset_DTM* is 11 bytes (including its header).

Format: Byte string

Byte	Content
0-1	LT header
	DATE
2-3	Year, in binary (e.g., 1989 is encoded as X′07C5′)
4	Month of the year, in binary (values from 1 to 12 are valid)
5	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
6	Hour of the day, in binary (values from 0 to 23 are valid)
7	Minute of the hour, in binary (values from 0 to 59 are valid)
8	Second of the minute, in binary (values from 0 to 59 are valid)
9	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
10	Indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X′E9′ (Z) TIME is GMT and no offset required. X′4E′ (+) TIME is local, OFFSET VALUE is required, and encoder′s local time is ahead of GMT. X′60′ (-) TIME is local, OFFSET VALUE is required, and encoder′s local time trails GMT. local time Note: All other values are reserved.
	OFFSET VALUE
11	Hour offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 13 are valid)
12	Minute offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 59 are valid)

Note: Refer to “Representing Date and Time” on page 15-40 for a complete discussion of the encoding and interpretation of date and time.

Reset_Accepted_MU

Description: The *reset_accepted_message_unit* is sent from DS_Receive to DS_Send. DS_Receive issues the *reset_accepted_MU* in response to a *reset_request_MU* to inform DS_Send that DS_Receive has reset its MU_ID Registry.

Unrecognized_Reserve

Description:	The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure. Intermediate DSUs pass *unrecognized_reserve* structures through unchanged in outgoing DMUs.
Format:	Undefined byte string

Representing Date and Time

Following is a discussion of the date and time formats recognized and supported by SNA/DS. Definitions and examples are also provided that illustrate the encoding and interpretation of each format.

Generalized Time Building Blocks

Generalized time is a term that is used to refer to a very general representation of time. It is comprised of a calendar date, a time of day, and, optionally, an offset from that time of day to some common time of day (i.e., GMT/UTC). SNA/DS encodes and interprets generalized time in numerous message units.

Time Formats Supported by SNA/DS

SNA/DS can encode and interpret the following generalized time formats:

Local-Only Time Local-only time is encoded as a date and a base local time (e.g., date = May 31, 2001; time = 11:22:33.44 p.m.).

If the local-only time format is encoded to represent generalized time, the interpreter cannot relate that time to GMT and, hence, to its own local time unless it has awareness of the encoder's relationship to GMT/UTC.

GMT-Only Time GMT-only time is encoded as a date and base GMT time followed by a time flag (the character "Z"). A time flag with value "Z" signals the interpreter that the base time is GMT, not local time (e.g., May 31, 2001; time = 06:11:22.33; Flag = Z).

If the GMT-only time format is encoded to represent generalized time, the interpreter can successfully relate that time to its own local time, but cannot determine the encoder's local time unless it has awareness of the encoder's relationship to GMT/UTC.

Offset Time Offset time is encoded as a date and base local time followed by a *signed offset*. The signed offset indicates the time differential between the base time and GMT/UTC. The sign of the signed offset also acts as the time flag. Hence, the offset time format can be identified when the time flag is either a "+" sign or "-" sign.

The sign of the offset is based on the encoder's location relative to GMT/UTC. Those locations that are just east (e.g., ahead) of GMT use a "+" offset (time flag) whereas those locations just west of (e.g., behind) GMT use a "-" offset sign (e.g., offset for New York would be Flag="-"; Offset = 5 hours).

If the offset time format is encoded to represent generalized time, the interpreter can successfully relate that time to GMT and, hence, its own local time.

Encoding Generalized Times

The encodings for generalized time are usually included in an encoding structure that has an LT header and, possibly, some other substructure. (The *Seqno_DTM* structure on page 15-18, for example, contains an LT header and a *sequence_number* substructure in addition to the generalized time substructure). The following encodings define the generalized time substructure only.

Byte	Content
	DATE
1-2	Year, in binary (e.g., 1989 is encoded as X′07C5′)
3	Month of the year, in binary (values from 1 to 12 are valid)
4	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
5	Hour of the day, in binary (values from 0 to 23 are valid)
6	Minute of the hour, in binary (values from 0 to 59 are valid)
7	Second of the minute, in binary (values from 0 to 59 are valid)
8	Hundredth of the second, in binary (values from 0 to 99 are valid)
	TIME FLAG
9	When present, indicates whether TIME should be interpreted as local or GMT. It may also act as the sign of a signed offset value. Possible values are listed below (with their equivalent EBCDIC characters shown in parentheses). X′E9′ (Z) TIME is GMT and no offset required. X′4E′ (+) TIME is local, OFFSET VALUE is required, and encoder′s local time is ahead of GMT X′60′ (-) TIME is local, OFFSET VALUE is required, and encoder′s local time trails GMT Note: All other values are reserved. The absence of the time flag and offset value indicate that time is local.
	OFFSET VALUE
10	Hour offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 13 are valid)
11	Minute offset from GMT, in binary, occurs when GMT flag ≠ X′E9′ (values from 0 to 59 are valid)

Interpreting Time Formats

In order to properly interpret an encoded generalized time, the interpreter must understand whether the base encoded time is local or GMT and what formula to use to convert between local and GMT times.

SNA/DS interprets encoded generalized times based on the following rules:

- If the time flag exists and has value ″Z″, then the base time is GMT. Otherwise, the base time is local time.
- Conversion between local and GMT/UTC times is performed via the formula: *GMT_time = local_time - signed_offset*.

Examples

Following are two examples that illustrate how generalized time is encoded and interpreted by SNA/DS. For each of these examples, assume DSU A is located in New York City (5 hours behind GMT) and DSU B is in Tokyo, Japan (9 hours ahead of GMT).

1. DSU A sends an MU to DSU B at 11:22:33.44 p.m. NYC time on May 31, 2001. DSU A may encode its generalized time in either the local-only, GMT-only, or offset time formats. Following are the encodings for each generalized time format (with the character equivalent of each encoding provided to improve understanding of the applicable hex encoding):

   a. Local-only time:

      ```
      X'07D1051F1716212C'
        yyyyMMddHHmmsshh
      C'2001053123223344'
      ```

      Note: A SNA/DS FS2 DSU will not encode the local-only time format (see *Seqno_DTM* length restriction on page 15-18).

   b. GMT-only time:

      ```
      X'07D1060104162l2CE9'
        yyyyMMddHHmmsshhF  (June 1, 2001 at 4:22:33.44 a.m. GMT)
      C'20010601042233442'
      ```

   c. Offset time:

      ```
      X'07D1051F1716212C600500'
        yyyyMMddHHmmsshh- HHmm
      C'2001053123223344- 0500'
      ```

   If DSU A encoded the offset time format, DSU B can successfully relate the supplied generalized time to its own local time by:

   - converting the offset time to GMT via the formula:

     ```
     GMT_time = base_time - (signed_offset)
     ```

   - converting GMT to its relative local time via the formula:

     ```
     relative_local_time = GMT_time + (signed_offset_of_interpreter)
     ```

   Using these formulas (which are both based on the interpreter's formula discussed earlier), DSU B can interpret DSU A's local time to be equal to:

   ```
   X'07D106010D16212C'
     yyyyMMddHHmmsshh   (June 1, 2001 at 1:22:33.44 p.m. Tokyo time)
   C'2001060113223344'
   ```

2. DSU B sends an MU to DSU A on January 1, 2000 at 7:00 a.m. Tokyo time and encodes its generalized time in offset format as follows:

   ```
   X'07D00101070000004E0900'
     yyyyMMddHHmmsshhF HHmm
   C'2000010107000000+ 0900'
   ```

   DSU A can interpret this time as either:

X'07CF0C1F16000000E9'
  yyyyMMddHHmmsshhF (December 31, 1999 at 10:00 p.m. GMT), or as
C'19991231220000000Z'

X'07D00C1F05000000'
  yyyyMMddHHmmsshh (December 31, 1999 at 5:00 p.m. NYC time)
C'19991231170000000'

Transaction Program and Server Names

Following is a list of all transaction program and server names defined for SNA/DS in the FM header 5 (Attach), in the Distribution MU, or used internally in the distribution service unit (DSU).

Code	Meaning
X′20**F0F0F0′**	DIA process destination transaction program name
X′20**F0F0F1′**	DIA server name
X′20**F0F0F2′**	DIASTATUS transaction program name
X′21**F0F0F1′**	DS_SEND transaction program name (FS1)
X′21**F0F0F2′**	DS_RECEIVE transaction program name (FS1)
X′21**F0F0F3′**	DS_ROUTER_DIRECTOR transaction program name
X′21**F0F0F6′**	SNA/DS general server name
X′21**F0F0F7′**	DS_SEND transaction program name (FS2)
X′21**F0F0F8′**	DS_RECEIVE transaction program name (FS2)
X′23**F0F0F0′**	SNA/MS Change Management agent TP name
X′24**F0F0F0′**	SNA/File Services server name
X′30**F0F0F2′**	Object Distribution transaction program.
X′30**F0F0F3′**	Object Distribution server transaction program.

Code Points Used by SNA/DS FS2

The values of the ID component of the LLID structure as used for SNA/DS GDS variables are shown below:

ID	Structure Name	Applicable MUs
1532	SNA Condition Report	*DRMU, REMU*
1570	Transport Prefix	*DTMU*
1571	Transport Command	*DTMU*
1572	Destination List	*DTMU*
1573	Agent Object	*DTMU, DRMU, DCMU*
1574	Server Object	*DTMU, DCMU*
1575	Report Command	*DRMU*
1576	Report Information	*DRMU*
1577	Receiver Exception Command	*REMU*
1578	Sender Exception Message Unit (type FS2)	*SEMU*
1579	Completion Query Message Unit	*CQMU*
157A	Completion Report Message Unit	*CRMU*
157B	Continuation Prefix	*DCMU*
157C	Report Prefix	*DRMU*
157E	Purge Report Message Unit	*PRMU*
157F	Suffix	*DTMU, DRMU, DCMU*
1580	External Network Correlation	*DTMU, DCMU*
1581	External Network Object	*DTMU, DCMU*
1582	Reported-On External Network Correlation	*DRMU*
1583	Report-To DSU/User	*DRMU*
1585	Reset Request Message Unit	*RRMU*
1586	Reset Accepted Message Unit	*RAMU*

End of Chapter 15

Chapter 16. SNA/File Services (FS)

Introduction

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 16-1. 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	16-17	ID	1549	27-128	1	Y	Y	2-8	—
U_of_W_Requester_DSU	16-17	T	01	8-22	1	N	Y	2	—
U_of_W_Requester_RGN	16-17	T	01	3-10	1	—	—	—	—
U_of_W_Requester_REN	16-17	T	02	3-10	1	—	—	—	—
U_of_W_Requester_User	16-18	T	03	8-22	0-1	N	Y	2	—
U_of_W_Requester_DGN	16-18	T	01	3-10	1	—	—	—	—
U_of_W_Requester_DEN	16-18	T	02	3-10	1	—	—	—	—
U_of_W_Requester_Agent	16-18	T	04	3-10	0-1*	—	—	—	—
U_of_W_Seqno_DTM	16-19	T	02	15-17	1	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-53	—	—	—	—	—

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

SNA/FS **Agent Request**

Figure 16-2. 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	16-21	ID	1530	9-13321	1	N	Y	1-2	—
Command	16-21	ID	1546	5	1	—	—	—	—
Command_Parms	16-21	ID	1547	7-13312	0-1	Y	N	1-15	—
Source_Reporting_Action	16-22	T	02	3	0-1*	—	—	—	—
Target_Agent	16-22	T	03	3-10	0-1*	—	—	—	—
Target_Reporting_Action	16-22	T	04	3	0-1*	—	—	—	—
Report-To_Agent	16-23	T	05	3-10	0-1*	—	—	—	—
Report-To_DSU	16-23	T	07	8-22	0-1*	N	N	2	—
Report-To_RGN	16-24	T	08	3-10	1	—	—	—	—
Report-To_REN	16-24	T	09	3-10	1	—	—	—	—
Report-To_User	16-24	T	0A	8-22	0-1	N	N	2	—
Report-To_DGN	16-24	T	0B	3-10	1	—	—	—	—
Report-To_DEN	16-25	T	0C	3-10	1	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-13238	—	—	—	—	—

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

SNA/FS **Server Request**

Figure 16-3. 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	16-25	Del-ID	pfx	≥28	1	Y	Y	3-11	—
FS_Server_Request_Prefix	16-25	ID/pfx	1531	8-19	1	N	Y	1-3	—
Decoder_Instruction	16-25	T	01	4-5	0-1*	—	—	—	—
Source_Instruction	16-25	T	02	4-5	0-1*	—	—	—	—
Target_Instruction	16-26	T	03	4-5	0-1*	—	—	—	—
Data_Object_Group	16-27	Del-ID	pfx	≥16	1	N	Y	3-5	—
Group_Prefix	16-28	ID/pfx	1533	4	1	—	—	—	—
Supplemental_FS_Info1	16-28	ID	1534	4-1024	0-1	Y	Y	1-9	—
Unrecognized_Reserve	16-21	T	—	2-1020	—	—	—	—	—
Supplemental_FS_Info2	16-28	ID	1535	8-32767	0-1	Y	Y	1-15	—
Supplemental_FS_Info3	16-28	ID	153C	9-283	0-1	—	—	—	—
Supplemental_FS_Info4	16-28	ID	1550	12-2048	0-1	—	—	—	—
Unrecognized_Reserve	16-21	ID	—	4-30432	—	—	—	—	—
Data_Object	16-28	Del-ID	pfx	≥18	1	Y	Y	3-19	—
D_O_Prefix	16-28	ID/pfx	1536	4	1	—	—	—	—
D_O_Attributes	16-28	ID	1537	10-1024	1	Y	Y	1-9	—
D_O_Class	16-28	T	81	6	1	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-1014	—	—	—	—	—
D_O_Global_Name	16-28	ID	1538	9-283	1	N	Y	1-7	16-13
Supplemental_FS_Info5	16-28	ID	1539	12-2048	0-1	—	—	—	—
D_O_Description	16-28	ID	153B	10-512	0-1	Y	Y	1-8	16-14
D_O_Transforms	16-29	ID	153E	14-1024	0-1	Y	Y	1-7	16-14
D_O_Allocation_Info	16-29	ID	153F	14-1024	0-1*	Y	Y	1-7	16-15
D_O_Contents	16-29	ID/seg	1541	≥5	0-1*	—	—	—	—
Unrecognized_Reserve	16-21	ID	—	4-32767	—	—	—	—	—
D_O_Suffix	16-29	ID/sfx	1542	4	1	—	—	—	—
G_Suffix	16-29	ID/sfx	1543	4	1	—	—	—	—
Unrecognized_Reserve	16-21	ID	—	4-32767	—	—	—	—	—
FS_Suffix	16-29	ID/sfx	154C	4	1	—	—	—	—

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

SNA/FS **Agent Report**

Figure 16-4. 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	16-29	ID	154A	14-32763	1	N	Y	2-3	—
Command	16-21	ID	1546	5	1	—	—	—	—
SNA_Condition_Report	* *	ID	1532	10-32749	0-1*	Y	Y	1-10	* *
FS_Action_Summary	16-29	ID	1548	5	1	—	—	—	—

Notes:

1. * Refer to the structure description for presence rule(s).
2. * * Refer to Appendix B, “Common Structures.”

SNA/FS **Server Report**

Figure 16-5. 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	16-30	Del-ID	pfx	≥22	1	N	Y	3-4	—
FS_Server_Report_Prefix	16-30	ID/pfx	154B	8-9	1	N	Y	1	—
Decoder_Instruction	16-25	T	01	4-5	1	—	—	—	—
SNA_Condition_Report	* *	ID/seg	1532	10-32749	0-1*	Y	Y	1-10	* *
Data_Object_Group	16-27	Del-ID	pfx	≥16	0-1*	N	Y	3-4	—
Group_Prefix	16-28	ID/pfx	1533	4	1	—	—	—	—
Supplemental_FS_Info2	16-28	ID	1535	8-32767	0-1	Y	Y	1-7	—
Supplemental_FS_Info3	16-28	ID	153C	9-360	0-1	—	—	—	—
Supplemental_FS_Info4	16-28	ID	1550	9-2045	0-1	—	—	—	—
Unrecognized_Reserve	16-21	ID	—	4-30358	—	—	—	—	—
Data_Object	16-28	Del-ID	pfx	≥8	1	Y	Y	2-13	—
D_O_Prefix	16-28	ID/pfx	1536	4	1	—	—	—	—
D_O_Global_Name	16-28	ID	1538	9-360	1	N	Y	1-8	16-13
Supplemental_FS_Info5	16-28	ID	1539	9-2045	0-1	—	—	—	—
Unrecognized_Reserve	16-21	ID	—	4-30354	—	—	—	—	—
D_O_Suffix	16-29	ID/sfx	1542	4	1	—	—	—	—
G_Suffix	16-29	ID/sfx	1543	4	1	—	—	—	—
FS_Suffix	16-29	ID/sfx	154C	4	1	—	—	—	—

Notes:

1. * Refer to the structure description for presence rule(s).
2. * * Refer to Appendix B, “Common Structures.”

Subtables

Global Names

Figure 16-6. 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	16-30	T	01	3-12	0-1*	—	—	—	—
To_Be_Fetched_Name	16-31	T	02	5-77*	0-1*	N	Y	1-10	* *
Fetching_Match_Flags	16-32	T	03	3-12	0-1*	—	—	—	—
To_Be_Stored_Name	16-32	T	04	5-77*	0-1*	N	Y	1-10	* *
To_Be_Deleted_Name	16-33	T	05	5-77*	0-1*	N	Y	1-10	* *
Deleting_Match_Flags	16-33	T	06	3-12	0-1*	—	—	—	—
Supplemental_FS_Info6	16-28	T	07	3-12	0-1*	—	—	—	—
Fetched_Name	16-33	T	08	5-77*	0-1*	N	Y	1-10	* *
Stored_Name	16-34	T	09	5-77*	0-1*	N	Y	1-10	* *
Deleted_Name	16-34	T	0A	5-77*	0-1*	N	Y	1-10	* *
Reported-On_Name	16-34	T	0B	5-77*	0-1*	N	Y	1-10	* *

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.
6. * * Refer to Appendix B, “Common Structures.”

Object Description

Figure 16-7. Subtable Encoding of the Group/Object Description

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Object_Description									
Object_Text_Description	16-34	T	01	14-255	0-1	N	Y	3	—
CCSID	16-34	T	01	4	1	—	—	—	—
Language_ID	16-34	T	02	5	1	—	—	—	—
Description_Text	16-35	T	03	3-244	1	—	—	—	—
Uniform_Text_Content	16-35	T	02	6-11	0-1	N	Y	2	—
CCSID	16-34	T	01	4	1	—	—	—	—
Language_ID	16-34	T	02	5	0-1	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-242	—	—	—	—	—

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

Object Transforms

Figure 16-8. Subtable Encoding of the Group/Object Transforms

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Object_Transforms									
Compression_Transform	16-35	T	01	10-128	0-1*	Y	Y	1-8	—
Compressed_State	16-35	T	01	3	1	—	—	—	—
FS_Compression	16-35	T	02	5-8	0-1*	N	Y	1-2	—
Compression_Technique	16-35	T	01	3	1	—	—	—	—
Prime_Character	16-36	T	02	3	0-1*	—	—	—	—
User_Compression	16-36	T	03	12-49	0-1*	N	Y	3	—
Algorithm_Name	16-36	T	01	3-14	1	—	—	—	—
Algorithm_Parms	16-36	T	02	3-29	1	—	—	—	—
CCSID	16-34	T	03	4	1	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-74	—	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-892	—	—	—	—	—

Notes:

1. This subtable is referenced by the *FS_server_request*.
2. * Refer to the structure description for presence rule(s).

Allocation Information

Figure 16-9. Subtable Encoding of the Allocation Information

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Allocation_Info									
Transfer_Size	16-36	T	08	10	1	—	—	—	—
Record_Information	16-36	T	09	15-64	0-1	Y	Y	2	—
Record_Format	16-36	T	0A	3	1	—	—	—	—
Max_Record_Size	16-37	T	0B	10	1	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-49	—	—	—	—	—
Unrecognized_Reserve	16-21	T	—	2-995	—	—	—	—	—

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

SNA Condition Report

See Appendix B, “Common Structures” on page B-1 for the SNA Condition Report. Note that the six Structure Names for the children of the Reported-On_Dest parent have different names in this chapter than appears in the referenced Appendix.

In this Chapter the names are:

- Reported-On_Dest
  - Reported-On_Dest_DSU
    - Reported-On_Dest_RGN
    - Reported-On_Dest_REN
  - Reported-On_Dest_User
    - Reported-On_Dest_DGN
    - Reported-On_Dest_DEN

In the Appendix the names are:

- Reported-On_Dest
  - Reported-On_Location_Name
    - Reported-On_NETID
    - Reported-On_Node_ID
  - Reported-On_User
    - Reported-On_Naming_Auth
    - Reported-On_Individual_ID

Structure Descriptions

Agent_Unit_of_Work

Description: The *agent_unit_of_work*, assigned by the requesting agent, provides the basis to track the progress of a particular defined task. The unit-of-work request is uniquely identified by the combination of *u_of_w_requester_DSU*, *u_of_w_requester_user*, *u_of_w_requester_agent*, and *u_of_w_sequence number/date-time*.

In SNA/FS, the unit of work identifies one or more generated SNA/DS distributions as belonging to the same SNA/FS defined task.

U_of_W_Requester_DSU

Description: The *unit_of_work_requester_DSU* is the name of the DSU at which the unit-of-work was requested.

U_of_W_Requester_RGN

Description: The *unit_of_work_requester_RGN* is the first part of the name of the DSU at which the unit-of-work was requested. This is typically, but not necessarily, the network ID.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

U_of_W_Requester_REN

Description: The *unit_of_work_requester_REN* is the second part of the name of the DSU at which the unit-of-work was requested. This is typically, but not necessarily, the LU name.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Note: In existing networks where LU names are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the REN may contain any of the three special characters; however, these characters

may not be available on keyboards in every country and should not be used in new LU names.

U_of_W_Requester_User

Description: The *unit_of_work_requester_user* is the user name of the originator of the unit-of-work request.

U_of_W_Requester_DGN

Description: The *unit_of_work_requester_DGN* is the first part of the user name of the unit-of-work originator.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

U_of_W_Requester_DEN

Description: The *unit_of_work_requester_DEN* is the second part of the user name of the unit-of-work originator.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

U_of_W_Requester_Agent

Description: The *unit_of_work_requester_agent* identifies the transaction program that originated the unit-of-work request.

Presence Rule: When the *unit_of_work_requester_agent* is absent, the *origin_agent* specified in the SNA/DS distribution is the default.

Format: Character string, except for first byte

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

The first byte of an SNA-registered transaction program name ranges in value from X′00 to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA-registered. X′40′ is not a valid first-byte value.

U_of_W_Seqno_DTM

Description:	The sequence number is the number assigned to the unit-of-work request by the SNA/FS originating agent. The value ranges from 1 to (2^{31})-1. The date of the unit-of-work request is assigned by the *u_of_w_requester_agent*; the time of the unit-of-work request is assigned by the *u_of_w_requester_DSU*. The offset from GMT for local time is included.
Format:	Byte string (See below)
Note:	Specification of local time without GMT is not supported, however, the ISO conformant method for specifying local time is to not include the GMT Flag and Offset bytes (14 byte structure length).

Byte	Contents
0-1	LT header
2-5	Sequence number Signed binary integer limited to $(2^{31})-1$.
	DATE
6-7	Year, in binary (e.g., year 1989 is encoded as X′07C5′)
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 23 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	GMT FLAG
14	Indicates that specified TIME is GMT and identifies whether offsets from GMT are required to calculate local time. (Equivalent EBCDIC characters are shown in parentheses.) X′E9′ (Z) no offset required X′4E′ (+) add required offset to GMT to get local time X′60′ (−) subtract required offset from GMT to get local time
	OFFSET
15	Hour offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 23 are valid)
16	Minute offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 59 are valid)

Examples

A 9-byte date/time encoding is a date/time followed immediately by an EBCDIC “Z” and is considered to be GMT. Thus, 12:00 GMT on 2 January 1988 would be

```
X′07C401020C000000E9′
  yyyymmddhhmmsshhZ
```

An 11-byte date/time encoding is a date/time followed immediately by an EBCDIC “+” or “-” and two one-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00 a.m. on 2 January 1988 in New York would be 12:00 GMT - 5 hours, or

```
X′07C401020C000000600500′
  yyyymmddhhmmsshh- hhmm
```

Unrecognized_Reserve

Description: The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure.

SNA/FS servers pass *unrecognized_reserve* structures through unchanged in the outgoing *server_object*.

Format: Undefined byte string

FS_Agent_Request

Description: The *FS_agent_request* contains the control information that describes the SNA/FS agent action to be performed.

Command

Description: The *command* specifies the type of SNA/FS request or SNA/FS reporting action.

Format: Byte string

Byte	Content	
0-3	LLID header	
4	X'10'	REPORT_FS_ACTION
	X'11'	REPORTING_FS_ACTION
	X'12'	TRANSFER_TO_REQUESTER

Note: REPORTING_FS_ACTION is valid only in reporting flows, while the other values are valid only in requesting flows.

Command_Parms

Description: The *command_parameters* contain and qualify the control information for the *command*.

Source_Reporting_Action

Description:	The *source_reporting_action* describes the type of reporting the source agent performs.
Presence Rule:	Occurs when the requesting agent requires reports from the source, and the *command* is TRANSFER_TO_REQUESTER; otherwise, precluded.
Format:	Byte string
Note:	The reporting action requested of the agent cannot be more demanding than that requested of the server.

Byte	Contents	
0-1	LT header	
2	X′01′	DETAILED
	X′10′	SUMMARY_OR_EXCEPTIONS
	X′11′	ONLY_IF_EXCEPTIONS

Target_Agent

Description:	The *target_agent* is the transaction program at the target location.
Presence Rule:	Occurs when the target_agent is different from the source agent, and the *command* is TRANSFER_TO_REQUESTER; otherwise, precluded. When the *target_agent* is absent, the *dest_agent* specified in the SNA/DS distribution is the default.
Format:	Character string, except for the first byte
CGCSGID:	01134-00500
String Conventions:	Leading, imbedded, and trailing space (X′40′) characters are not allowed. The first byte of an SNA-registered transaction program name ranges in value from X′00 to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA-registered. X′40′ is not a valid first-byte value.

Target_Reporting_Action

Description:	The *target_reporting_action* describes the type of reporting the target agent performs.
Presence Rule:	Occurs when the requester requires target reporting, and the *command* is REPORT_FS_ACTION or TRANSFER_TO_REQUESTER; otherwise, precluded.
Format:	Byte string

Byte	Contents	
0-1	LT header	
2	X′01′	DETAILED
	X′10′	SUMMARY_OR_EXCEPTIONS
	X′11′	ONLY_IF_EXCEPTIONS

Report-To_Agent

Description: The *report-to_agent* is the name of the transaction program to which reports are to be delivered after the SNA/FS activity has concluded.

Presence Rule: Occurs when the requester requires reporting to a third-party agent that is different from the requesting agent, and the *command* is REPORT_FS_ACTION or TRANSFER_TO_REQUESTER; otherwise, precluded. When the *report-to_agent* is absent and reporting is required, the *dest_agent* specified in the SNA/DS distribution is the default.

Format: Character string, except for the first byte

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

The first byte of an SNA-registered transaction program name ranges in value from X′00 to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA-registered. X′40′ is not a valid first-byte value.

Report-To_DSU

Description: The *report-to_DSU* is the name of the DSU to which the SNA/FS reports are to be delivered.

Presence Rule: Occurs when the requester requires reporting and requests the reports be delivered to a DSU other than the default DSU. When the *report-to_DSU* is absent, the *report-to_DSU* specified in the SNA/DS distribution is the default. If the *report-to_DSU* is also absent, the *origin_DSU* is the default. Typically the SNA/DS distributions between the source and target locations normally carry the requesting DSU as the SNA/DS *report-to_DSU*.

Report-To_RGN

Description:	The *report-to_RGN* is the first part of the DSU name to which the SNA/FS reports are to be delivered. This is typically, but not necessarily, the network ID.
Format	Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

Report-To_REN

Description:	The *report-to_REN* is the second part of the DSU name to which the SNA/FS reports are to be delivered. This is typically, but not necessarily, the LU name.
Format	Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Note: In existing networks where LU names are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the REN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new LU names.

Report-To_User

Description:	The *report-to_user* is the name of the user to which the SNA/FS reports are to be delivered.

Report-To_DGN

Description:	The *report-to_DGN* is the first part of the user name to which the SNA/FS reports are to be delivered.
Format:	Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Report-To_DEN

Description: The *report-to_DEN* is the second part of the user name to which the SNA/FS reports are to be delivered.

Format: Character string

CGCSGID: 01134-00500

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

FS_Server_Request

Description: The *FS_server_request* describes the action to be performed by the server, and may also contain object identifiers and object contents.

FS_Server_Request_Prefix

Description: The *FS_server_request_prefix* identifies the beginning of the *FS_server_request*.

Decoder_Instruction

Description: The *decoder_instruction* describes the server action to be performed by the decoder-role server at either the source location or report-to location.

Presence Rules: Occurs when:

- The TRANSFER_TO_REQUESTER agent command and its accompanying server request flow from the requesting location to the source location.
- The REPORTING_FS_ACTION agent command and its accompanying server report flow from the target location to the report-to location.

Format: Bit string

Note: The values for the *decoder_instruction* are described on page 16-27.

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 16-27.

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 16-27.

Server Instructions, Decoder, Source, and Target

Byte	Bit	Contents	Server Role
0-1		LT header	
2	0-3	Server instruction:	
		0001 FETCH	source
		0010 DECODE	decoder
		0011 CREATE_LOAD_OR_REPLACE	target
		0100 DELETE	target
		0101 REPLACE	target
		0110 CREATE_LOAD	target
	4-7	Exception action:	
		0001 ABEND	decoder, source, or target
		0010 BACKOUT	target
3	0-3	Reporting action:	
		0001 DETAILED	decoder, source, or target
		0010 SUMMARY_OR_EXCEPTIONS	decoder, source, or target
		0011 ONLY_IF_EXCEPTIONS	decoder, source, or target
	4-7	Reserved	
4	0-3	Intention (see Note):	
		0001 EXECUTING	target
		0011 STORING	target
		0100 NOT APPLICABLE	decoder, source, or target
	4-7	Transform Action (see Note):	
		0000 NOT APPLICABLE	source or target
		0001 COMPRESS	source or target
		0010 DECOMPRESS	source or target

Note: Byte 4 is optional and may be omitted.

Data_Object_Group

Description: The *data_object_group* defines the overall characteristics about the data object.

Presence Rules: Required in:

- The *FS_server_request*.
- The *FS_server_report* whenever the *SNA_condition_report* is absent; otherwise, optional.

Group_Prefix

Description: The *data_object_group_prefix* identifies the beginning of the *data_object_group*.

Supplemental_FS_Info1-Supplemental_FS_Info6

Description: The *supplemental_FS_info1* - *supplemental_FS_info6* structures are reserved for future use.

Data_Object

Description: The *data_object* is the basic entity managed by SNA/FS.

D_O_Prefix

Description: The *data_object_prefix* identifies the beginning of the *data_object*.

D_O_Attributes

Description: The *data_object_attributes* contain information about the contents of the data object that the SNA/FS server uses to determine whether the server can honor the request.

D_O_Class

Description: The *data_object_class* identifies the class of the data object by means of a hierarchical structure of codes. The classification and intention information are used by the target server to determine whether or not the request can be honored.

Format: Byte string

Notes: 1. Refer to “SNA/FS Data Object Classification Codes” on page 16-38 for the value descriptions.

2. When unknown by requester, all 0 bits are used. Source will supply.

D_O_Global_Name

Description: The *data_object_global_name* is the unique, system-independent identifier for the data object. The name is assigned according to naming conventions established by the using architecture. The canonical identifier consists of a string of tokens, where the leftmost tokens are more significant. A higher-order token identifies the naming authority that issues or manages the values of the lower-order tokens.

D_O_Description

Description: The *data_object_description* provides brief descriptive information about the object.

D_O_Transforms

Description: The *data_object_transform* defines the object transform that has been applied to the contents of the data object.

D_O_Allocation_Info

Description: The *data_object_allocation_info* provides the target location with space requirements needed to store the data object.

Presence Rule: Occurs when *data_object_contents* is present.

D_O_Contents

Description: The *data_object_contents* is the byte contents of the *data_object*.

Presence Rule: Precluded when the *decoder_instruction* is present or the *target_instruction* is DELETE.

Format: Undefined byte string

D_O_Suffix

Description: The *data_object_suffix* contains no information and marks the end of the *data_object*.

G_Suffix

Description: The *data_object_group_suffix* contains no information and marks the end of the *data_object_group*.

FS_Suffix

Description: The *FS_suffix* contains no information and marks the end of the *FS_request* or the *FS_report*.

FS_Agent_Report

Description: The *FS_agent_report* provides a summary on the relative success of a previous SNA/FS request.

FS_Action_Summary

Description: The *FS_action_summary* indicates whether the actions requested of the server were successfully performed.

Format Bit string

Note: The values for the *FS_action_summary* bit string are described on page 16-29.

Byte	Bit	Contents
0-3		LLID header
4	0-1	01 ALL_SUCCESSFUL (see Note) 11 NONE_SUCCESSFUL
	2-3	00 NO_BACKOUT_ATTEMPTED 01 ALL_BACKED_OUT
	4-5	00 ABEND_NOT_APPLICABLE 01 SERVER_ABEND
	6-7	Reserved

Note: If this value (ALL SUCCESSFUL) is present, all subsequent bits are 0.

FS_Server_Report

Description: The *FS_server_report* provides information on the relative success of one or more server operations.

FS_Server_Report_Prefix

Description: The *FS_server_report_prefix* identifies the beginning of the *FS_server_report*.

Token_Attributes

Description: The *token_attributes* define for each token in the global name how that token can be used in partial matching or token value generation. These attributes are stored in the SNA/FS catalog.

Presence Rule: Occurs when the server instruction is a create operation (e.g., CREATE_LOAD; CREATE_LOAD_OR_REPLACE).

Format: Bit string (See below)

Byte	Contents
0-1	LT header
2-11	Up to 10 different token attributes can be specified.

Token Attribute Values

For each token in the token string, there will be a single byte of attribute information, as follows:

Bit	Contents	
0	0	MUST_MATCH
	1	NEED_NOT_MATCH
1	0	NOT_GENERABLE
	1	GENERABLE
2	Reserved	
3-7	00000	UNSPECIFIED TYPE, ≤16 CHARACTERS
	00001	NETID
	00010	LU-NAME
	00011	SYSTEM_TYPE
	00100	SUBTREE_INDICATOR (STI)
	10000	ORDERED, ≤16 CHARACTERS
	10001	ORDERED, ≤16 DECIMAL NUMERICS
	10010	ORDERED, DATE - Y1991M12D31
	10011	ORDERED, TIME - H23M59S59
	10100	ORDERED, G00V00

Notes:

1. The target SNA/FS server is obligated to preserve the attribute characteristic in the catalog at the target node and to honor subsequent deletion requests based on this characteristic. If all bits in the catalog entry attribute byte are 0, i.e., MUST_MATCH, the corresponding identifier must be exactly matched for deleting and replacing operations.
2. If all of the flag bits in all of the token attribute bytes are set at their default values, e.g., 0, the token attributes may be omitted and the target server assumes the default.

To_Be_Fetched_Name

Description:	The *to_be_fetched_name* is the name of the object, at the source location, that is to be fetched by the SNA/FS server.
Presence Rule:	Occurs in: • The *FS_server_request* when an object is to be fetched from the source location. The source server instruction must be FETCH. • The *FS_server_report* when the FETCH server operation was unsuccessful or not attempted, and reporting was requested.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Fetching_Match_Flags

Description:	The *fetching_match_flags* govern the partial matching operation at fetch time.
Presence Rule:	Occur when partial matching is required at fetch time.
Format:	Byte string

Byte	Contents
0-1	LT header
2-11	For each token in the token string, up to a maximum of 10 tokens, a single byte describes that token's use in a fetch operation.

Fetching Match Flag Values

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

To_Be_Stored_Name

Description:	The *to_be_stored_name* is the name of the object that is to be stored at the target location. Typically, the source-role server will obtain the name at fetch time.
Presence Rule:	Occurs in: • The *FS_server_request* flow between the source and target locations when an object is to be stored at the target location. The requester can also specify parts of a *to_be_stored_name*; therefore, in this case, the structure is present between the requesting and source locations. • The *FS_server_report* when the storing operation was unsuccessful or not attempted, and reporting was requested.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

To_Be_Deleted_Name

Description:	The *to_be_deleted_name* is the name of the object, at the target location, that is to be deleted by the SNA/FS server.
Presence Rule:	Occurs in: • The *FS_server_request* when an object is to be deleted from the target location. • The *FS_server_report* when the delete operation was unsuccessful or not attempted, and reporting was requested.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.
Note	For a replace operation, the *to_be_deleted* name needs to contain only the NEED_NOT_MATCH tokens that differ from the values in the identifier of the *to_be_stored* data objects.

Deleting_Match_Flags

Description:	The *deleting_match_flags* govern the matching operation, at the target location, of the object to be deleted.
Presence Rule:	Occurs when partial matching is required to identify the *to_be_deleted* object.
Format:	Byte string

Byte	Contents
0-1	LT header
2-11	For each token in the token string, up to a maximum of 10 tokens, a single byte describes that token's use in a delete operation.

Deleting Match Flag Values

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

Fetched_Name

Description:	The *fetched_name* is the name of the object fetched by the SNA/FS server.
Presence Rule:	Occurs only in the *FS_server_report* when the source agent reports that an object has been fetched.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Stored_Name

Description:	The *stored_name* is the name of the object stored by the SNA/FS server.
Presence Rules:	Occurs: • In the *FS_server_report* when the target agent reports that an object has been stored. • When the request is being used to convey a data object name.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Deleted_Name

Description:	The *deleted_name* is the name of the object deleted by the SNA/FS server.
Presence Rule:	Occurs only in the *FS_server_report* when the target agent reports that an object has been deleted.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Reported-On_Name

Description:	The *reported-on_name* is the name of the object being reported by the SNA/FS server. The *reported-on_name* is used in cases when the state of the object being reported on cannot be determined.
Presence Rule:	Occurs only in the *FS_server_report*.
Length Restriction:	The maximum length for the global name is 65-n, where n is the number of tokens in the name.

Object_Text_Description

Description:	The *object_text_description* identifies the descriptive text and how the text is to be interpreted.

CCSID

Description:	The *coded_character_set_id* identifies the codepage and character set in which the text message is encoded. The structure of the CCSID is documented in the *Character Data Representation Architecture Reference.*
Format:	Bit string

Language_ID

Description:	The *language_id* identifies the coded national language in which the text message is written. The language IDs are defined in Volume 2 of the *National Language Information and Design Guide.*
Format:	Character string

Descriptive_Text

Description: The *descriptive_text* contains a brief description about the data object.

Format: Character string

Uniform_Text_Content

Description: The *uniform_text_content* identifies for any text data object its associated codepage and character set.

Compression_Transform

Description: The *compression_transform* indicates that the data object has been compressed.

Presence Rule: Required when the *data_object_contents* was stored compressed at the source server.

Compressed_State

Description: The *compressed_state* indicates if the *data_object_contents* is compressed.

Format: Byte string

Byte	Contents
0-1	LT header
2	X'01' OBJECT_IS_COMPRESSED
	X'02' OBJECT_IS_NOT_COMPRESSED

FS_Compression

Description: The *FS_Compression* identifies that the *data_object_contents* has been compressed using the compression algorithm defined by the SNA/FS server.

Presence Rule: Precluded when *user_compression* is present.

Compression_Technique

Description: The *compression_technique* identifies the SNA/FS-defined compression algorithm.

Format: Byte string

Byte	Contents
0-1	LT header
2	X'01' SCB_COMPRESSION (STRING CONTROL BYTE)

Prime_Character

Description:	The *prime_character* identifies the character to be used to replace repetitive sequences of that character.
Presence Rule:	Required when the *prime_character* is not the default value.
Format:	Byte string, single byte; the default is the space character (X′40′).

User_Compression

Description:	The *user_compression* identifies that the *data_object_contents* has been compressed using a user-defined compression algorithm.
Presence Rule:	Precluded when *FS_compression* is present.

Algorithm_Name

Description:	The *algorithm_name* identifies the user-defined compression algorithm.
Format:	Character string

Algorithm_Parms

Description:	The *algorithm_parms* identifies the parameters needed for the user-defined algorithm.
Format:	Character string

Transfer_Size

Description:	The *transfer_size* is an estimate of the number of bytes in the *data_contents*. It can be larger or smaller than the actual size; however, it should be accurate enough for the target location to use for space decisions.
Format:	Unsigned binary integer (1-origin)

Record_Information

Description:	The *record_information* describes the record layout of the data object.

Record_Format

Description:	This specifies the *record_format* of the data object.
Format	Byte string

Byte	Contents
0-1	LT header
2	X′01′ FIXED
	X′02′ VARIABLE

Max_Record_Size

Description:	This is the *maximum_record_size* of any record that can occur in the data object.
Format	Unsigned binary integer (1-origin)

SNA/FS Data Object Classification Codes

SNA/FS Data Object Classes				Hex Codes			
Level 1	**Level 2**	**Level 3**	**Level 4**	**1**	**2**	**3**	**4**
Executable Data Object	System Microcode *	Unspecified	Unspecified	10	10	00	00
		Patch	Unspecified Product Specific	10 10	10 10	10 10	00 Ex
		Fix	Unspecified Product Specific	10 10	10 10	20 20	00 Ex
		Suffix_EC	Unspecified Product Specific	10 10	10 10	30 30	00 Ex
		Maint_EC	Unspecified Product Specific	10 10	10 10	40 40	00 Ex
		Funct_EC	Unspecified Product Specific	10 10	10 10	50 50	00 Ex
		Feature	Unspecified NLS_EC Product Specific I/O_EC Customer Specific	10 10 10 10 10	10 10 10 10 10	60 60 60 60 60	00 51 Ex E0 Fx
	Microcode Customization	Unspecified	Unspecified	10	20	00	00
		Product Specific	Unspecified	10	20	Ex	00
		History_Log	Unspecified	10	20	E0	00
		Activate_Log	Unspecified	10	20	E1	00
		Canonical_Directory	Unspecified	10	20	E2	00
		MCF_Directory	Unspecified	10	20	E3	00
	Application Procedure	Unspecified	Unspecified	10	50	00	00
		CLIST	Unspecified	10	50	20	00
		EXEC	Unspecified	10	50	30	00
		SAA REXX	Unspecified	10	50	50	00
		Product Specific	Unspecified	10	50	Ex	00
		AS/400	Program	10	50	E2	01
Maintenance	Dump	Unspecified	Unspecified	40	10	00	00
	Configuration File	Unspecified	Unspecified	40	20	00	00
	Trace Information	Unspecified	Unspecified	40	30	00	00
	Error Log	Unspecified	Unspecified	40	40	00	00

Note: * Microcode may be classified as IBM Licensed Internal Code. See “Notices” near the beginning of this document for more information.

Code Points Used by SNA/FS

The values of the ID component of the LLID structures as used for SNA/FS GDS variables are shown below:

ID	Structure Name
1530	FS Agent Request
1531	FS Server Request Prefix
1532	SNA Condition Report
1533	Data Object Group Prefix
1534	Supplemental FS Information1
1535	Supplemental FS Information2
1536	Data Object Prefix
1537	Data Object Attributes
1538	Data Object Global Name
1539	Supplemental FS Information5
153B	Data Object Description
153C	Supplemental FS Information3
153E	Data Object Transforms
153F	Data Object Allocation Information
1541	Data Object Contents
1542	Data Object Suffix
1543	Data Object Group Suffix
1546	Command
1547	Command Parms
1548	FS Action Summary
1549	Agent Unit of Work Correlator
154A	FS Agent Report
154B	FS Server Report Prefix
154C	FS Suffix
1550	Supplemental FS Information4

Transaction Program and Server Names

The following is a list of the SNA/FS-defined server name, the SNA/FS-defined transaction program name, and the names of other SNA/FS-capable transaction programs.

Code	Meaning
X′24F0F0F0′	SNA/FS server name
X′24F0F0F1′	SNA/FS agent TP name
X′23F0F0F0′	SNA/MS change management agent TP name

Global Name Registration

The following is a list of the identifier tokens that have been registered in SNA/FS on behalf of SNA/FS-capable agents.

First Identifier	Agent
C′MCODE′	SNA/MS change management
C′MCUST′	SNA/MS change management
Registered Enterprise ID	SNA/MS change management

The following is a list of the subtree indicator tokens that have been registered in SNA/FS on behalf of SNA/FS-capable agents.

Subtree Indicator	Agent
C′GRP′ (Group)	SNA/MS change management
C′REF′ (Refresh)	SNA/MS change management
C′UPD′ (Update)	SNA/MS change management
C′FIX′ (Fix)	SNA/MS change management
C′LIB′ (Library)	Requester specified (SNA/FS agent TP name is default)
C′MEM′ (Member)	Requester specified (SNA/FS agent TP name is default)
C′OBJ′ (Object)	Requester specified (SNA/FS agent TP name is default)

SNA/FS Subtrees and Enterprise Structured Names

Enterprise Structured Names

Figure 16-10. Identification Tokens for All Objects Using Enterprise Structured Names

Token number	Token Attributes (assigned when file is created)	Contents
1	Must match, not generable, not subtree ID, unordered - type unspecified	Enterprise ID (from the structured net ID registered with IBM)
2 - 10	User specifiable	User specifiable

The **nationally structured** enterprise ID token of the enterprise structured name consists of a 2-byte country code followed by a 4-byte enterprise code. (See your IBM representative to register structured net ID's using the SNAREGISTRY application on HONE).

The structured network ID is defined as follows:

countrycode(2*bytes*)||*enterprisecode*(4*bytes*)||*networksuffix*(2*bytes*)

The nationally structured enterprise ID is defined as follows (notice that the 2-byte network suffix is not part of it):

countrycode(2*bytes*)||*enterprisecode*(4*bytes*)

A newer, preferred name for the SNA net ID enterprise code is the SNA net ID national organization code, resulting in:

countrycode(2*bytes*)||*nationalorganizationcode*(4*bytes*)

The user-specifiable tokens of the enterprise structured name contain one or more component name tokens and may contain, as a subset, one of the subtrees defined in the tables on the following pages or in the *&msbook.* change management chapters. An architecturally defined subtree may be used in global names that represent a library, a generic object, a member, or a change management data object.

SNA/FS **Defined Subtree structure**

Figure 16-11. Identification Tokens of the Architecturally Defined Subtree for Library Objects

Token number	Token Attributes (assigned when file is created)	Contents
n	Must match, not generable, subtree ID, unordered - type unspecified	C′LIB′
n+1	Must match, not generable, not subtree ID, ordered character	Library name
n+2	Need not match, not generable, not subtree ID, ordered character	Prepared-for qualifier (optional — value may be null or empty)

For enterprise structured global names representing library objects (e.g., AS/400), this is the definition of the LIB subtree. The LIB subtree identifier token value is registered by SNA/FS. Library object global names include a LIB subtree at the end of their common root token string.

The LIB subtree is used to extend the global name of a data object for the purposes of identifying the object as a complete library with a system-specific internal structure. The data object class code identifies the specific system.

The library name token value is the system-specific name of the library the data object contains.

The prepared-for qualifier token value specifies the functional level of the software for which the data object has been prepared. This token is used only when this information is required to specify the internal structure of the data object for the target node.

Library objects must follow the rules of elementary objects with respect to supporting the function of the SNA/FS global catalog. Specifically, library object names must be unique and object immutability must be enforced. It is the responsibility of the name tree designer and the creator of the objects named with LIB subtrees to ensure that subtree tokens provide for unique object names. The subtree tokens as defined above may not fully satisfy this requirement.

Additional tokens may be required following the LIB subtree to completely fulfill this requirement (e.g., date and time).

Figure 16-12. Identification Tokens of the Architecturally Defined Subtree for Save File Objects

Token number	Token Attributes (assigned when file is created)	Contents
n	Must match, not generable, subtree ID, unordered - type unspecified	C′OBJ′
n + 1	Must match, not generable, not subtree ID, ordered character	Library name
n + 2	Need not match, not generable, not subtree ID, ordered character	Object name1
n + 3	Need not match, not generable, not subtree ID, ordered character	Object name2 (optional — value may be null or empty)
n + 4	Need not match, not generable, not subtree ID, ordered character	Prepared-for qualifier (optional — value may be null or empty)

For enterprise structured global names representing generic (e.g., AS/400 save file or save/restorable) objects, this is the definition of the OBJ subtree. The OBJ subtree identifier token value is registered by SNA/FS. Generic object global names include an OBJ subtree at the end of their common root token string.

The OBJ subtree is used to extend the global name of a data object for the purposes of identifying the object as a generic object with a system-specific internal structure. The data object class code identifies the specific system.

The library name, object name1 and object name2 tokens contain the value of the system-specific name of the generic object the data object contains.

The prepared-for qualifier token value specifies the functional level of the software for which the data object has been prepared. This token is used only when this information is required to specify the internal structure of the data object for the target node.

Generic objects must follow the rules of elementary objects with respect to supporting the function of the SNA/FS global catalog. Specifically, generic object names must be unique and object immutability must be enforced. It is the responsibility of the name tree designer and the creator of the objects named with OBJ subtrees to ensure that subtree tokens provide for unique object names. The subtree tokens as defined above may not fully satisfy this requirement.

Additional tokens may be required following the LIB subtree to completely fulfill this requirement (e.g., date and time).

Figure 16-13. Identification Tokens of the Architecturally Defined Subtree for Member Objects

Token number	Token Attributes (assigned when file is created)	Contents
n	Must match, not generable, subtree ID, unordered - type unspecified	C′MEM′
n+1	Must match, not generable, not subtree ID, ordered character	Library name
n+2	Need not match, not generable, not subtree ID, ordered character or ordered date	File name
n+3	Need not match, not generable, not subtree ID, ordered character	Member name
n+4	Need not match, not generable, not subtree ID, ordered character	Prepared-for qualifier (optional — value may be null or empty)

For enterprise structured global names representing member (e.g., AS/400 file member or source member) objects, this is the definition of the MEM subtree. The MEM subtree identifier token value is registered by SNA/FS. Member object global names include a MEM subtree at the end of their common root token string.

The MEM subtree is used to extend the global name of a data object for the purposes of identifying the object as a library member for a specific system. The data object class code identifies the specific system.

The library name, file name and member name tokens contain the value of the system-specific name of the member the data object contains.

The prepared-for qualifier token value specifies the functional level of the software for which the data object has been prepared. This token is used only when this information is required to specify the internal structure of the data object for the target node.

Member objects must follow the rules of elementary objects with respect to supporting the function of the SNA/FS global catalog. Specifically, member object names must be unique and object immutability must be enforced. It is the responsibility of the name tree designer and the creator of the objects named with MEM subtrees to ensure that subtree tokens provide for unique object names. The subtree tokens as defined above may not fully satisfy this requirement.

Additional tokens may be required following the LIB subtree to completely fulfill this requirement (e.g., date and time).

End of Chapter 16

Appendix A. SNA Character Sets and Symbol-String Types

Introduction

This appendix describes the character sets and symbol-string types used, for example, for the following fields:

- LU name
- Network-qualified LU name
- Mode name
- COS name
- Transaction program name
- Access security information subfields
- Program initialization parameters (PIP) subfields
- Map name
- SNADS server, user (DGN, DEN), and service unit (RGN, REN) names

The detailed syntax of these strings is described in other chapters where their usage within individual message units is defined.

Symbol-String Type

The symbol-string type specifies the set of code points and corresponding characters from which the strings listed above are composed, as follows:

- Type A (Assembler oriented): a character string consisting of one or more characters from character set A. The first character of a type-A symbol string is not a numeric; i.e., it is different from X′F0′, X′F1′, ..., or X′F9′.
- Type 1134 (Type A subset): a character string consisting of one or more EBCDIC uppercase letters A through Z and numerics 0 through 9. For certain names, IBM implementation usage constrains the leading character to be alphabetic; these names include the following:
  - network ID
  - network name (e.g., LU name, link name)
  - mode name — SNA-defined user-session mode names are prefixed by X′7B′ (represented by the “#” graphic character in U.S. EBCDIC fonts) to distinguish them from user-defined names
  - class-of-service name (COS name) — SNA-defined user-session COS names are prefixed by X′7B′ (represented by the “#” graphic character in U.S. EBCDIC fonts) to distinguish them from user-defined names

  Earlier versions of the architecture permitted type-A symbol strings in network IDs and network names; this usage is now retired, but implementations must continue to support receipt of such strings from back-level partners.
- Type AE (A extended): a character string consisting of one or more characters from character set AE, with no restriction on the first character.
- Type 930 (distribution services oriented): a character string consisting of one or more characters from character set 930, with the following rules:

  - No leading space (X′40′) characters are used, but no other restrictions exist on the first character.
  - Imbedded space (X′40′) characters are significant.
  - Trailing space (X′40′) characters are not significant.
- Type USS (unformatted system services oriented, used for character-coded requests): a character string consisting of one or more characters from character set USS, with no restriction on the first character.
- Type GR (EBCDIC graphics): a byte string consisting of one or more bytes within the range X′41′ through X′FE′, with no restriction on the first byte.
- Type G (general): a byte string consisting of one or more bytes within the range X′00′ through X′FF′, with no restriction on the first byte.
- Type DB (double byte): a byte-string consisting of an even number of four or more bytes beginning with a byte set to X′0E′, followed by bytes having values in the range X′41′ through X′FE′, and ending with a byte set to X′0F′.

SNA Character Sets and Encodings

A character set is a set of graphic characters, such as letters, numbers, and special symbols. SNA formats make use of a variety of character sets. Character sets A, AE, 930, USS, 1134, and 640 define the characters that are allowed in the corresponding symbol-strings.

Each character set is encoded using a code page. A code page is the specification of code points, or hexadecimal values, for one or more character sets. All character sets used by SNA are encoded using IBM code page 00500, the relative encodings of which are shown in Figure A-1.

Character sets encoded using a specific code page are officially denoted by the concatenation of their character set and code page numbers, such as 00640-00500 and 01134-00500. The concatenation of these two numbers specifies a *coded graphic character set*. The older character sets—A, AE, 930, and USS—and their encodings continue to be supported but not for new formats, which now use 00640-00500 and 01134-00500.

Figure A-1 on page A-5 defines the character sets and encodings for A, AE, 930, USS, 01134-00500, and 00640-00500. The code points that do not belong to any of these sets are not shown.

Figure A-1 (Page 1 of 3). Character Sets A, AE, 930, USS, 1134, and 640

Hex Code	Graphic	Description	Set					
			A	AE	930	USS	1134	640
15		Line Feed				X		
40		Space			X	X		X
4B	.	Period		X	X	X		X
4C	<	Less Than Sign						X
4D	(	Left Parenthesis				X		X
4E	+	Plus Sign				X		X
50	&	Ampersand			X	X		X
59	ß	Sharp s			X			
5B	$	Dollar Sign	X	X	X	X		
5C	*	Asterisk				X		X
5D	)	Right Parenthesis				X		X
5E	;	Semicolon						X
60	-	Minus Sign			X	X		X
61	/	Slash			X	X		X
62	Â	A Circumflex, Capital			X			
63	Ä	A Diaeresis, Capital			X			
64	À	A Grave, Capital			X			
65	Á	A Acute, Capital			X			
66	Ã	A Tilde, Capital			X			
67	Å	A Overcircle, Capital			X			
68	Ç	C Cedilla, Capital			X			
69	Ñ	N Tilde, Capital			X			
6B	,	Comma			X	X		X
6C	%	Percent Sign						X
6D	_	Underline						X
6E	>	Greater Than Sign						X
6F	?	Question Mark						X
71	É	E Acute, Capital			X			
72	Ê	E Circumflex, Capital			X			
73	Ë	E Diaeresis, Capital			X			
74	È	E Grave, Capital			X			
75	Í	I Acute, Capital			X			
76	Î	I Circumflex, Capital			X			
77	Ï	I Diaeresis, Capital			X			
78	Ì	I Grave, Capital			X			
7A	:	Colon						X
7B	#	Number Sign	X	X	X	X		
7C	@	At Sign	X	X	X	X		
7D	′	Apostrophe			X	X		X
7E	=	Equal Sign				X		X
7F	″	Quotation Marks						X

Figure A-1 (Page 2 of 3). Character Sets A, AE, 930, USS, 1134, and 640

Hex Code	Graphic	Description	Set					
			A	AE	930	USS	1134	640
80	Ø	O Slash, Capital			X			
81	a	a, Small		X				X
82	b	b, Small		X				X
83	c	c, Small		X				X
84	d	d, Small		X				X
85	e	e, Small		X				X
86	f	f, Small		X				X
87	g	g, Small		X				X
88	h	h, Small		X				X
89	i	i, Small		X				X
91	j	j, Small		X				X
92	k	k, Small		X				X
93	l	l, Small		X				X
94	m	m, Small		X				X
95	n	n, Small		X				X
96	o	o, Small		X				X
97	p	p, Small		X				X
98	q	q, Small		X				X
99	r	r, Small		X				X
9A	a̲	a Underscore, Small			X			
9B	o̲	o Underscore, Small			X			
9E	Æ	AE Dipthong, Capital			X			
A0	µ	Micro, Mu			X			
A2	s	s, Small		X				X
A3	t	t, Small		X				X
A4	u	u, Small		X				X
A5	v	v, Small		X				X
A6	w	w, Small		X				X
A7	x	x, Small		X				X
A8	y	y, Small		X				X
A9	z	z, Small		X				X
AC		D Stroke, Capital			X			
AD	·	Y Acute, Capital			X			
AE		Thorn, Capital			X			
C1	A	A, Capital	X	X	X	X	X	X
C2	B	B, Capital	X	X	X	X	X	X
C3	C	C, Capital	X	X	X	X	X	X
C4	D	D, Capital	X	X	X	X	X	X
C5	E	E, Capital	X	X	X	X	X	X
C6	F	F, Capital	X	X	X	X	X	X
C7	G	G, Capital	X	X	X	X	X	X

Figure A-1 (Page 3 of 3). Character Sets A, AE, 930, USS, 1134, and 640

Hex Code	Graphic	Description	Set					
			A	AE	930	USS	1134	640
C8	H	H, Capital	X	X	X	X	X	X
C9	I	I, Capital	X	X	X	X	X	X
D1	J	J, Capital	X	X	X	X	X	X
D2	K	K, Capital	X	X	X	X	X	X
D3	L	L, Capital	X	X	X	X	X	X
D4	M	M, Capital	X	X	X	X	X	X
D5	N	N, Capital	X	X	X	X	X	X
D6	O	O, Capital	X	X	X	X	X	X
D7	P	P, Capital	X	X	X	X	X	X
D8	Q	Q, Capital	X	X	X	X	X	X
D9	R	R, Capital	X	X	X	X	X	X
DF	ÿ	y Diaeresis, Small			X			
E2	S	S, Capital	X	X	X	X	X	X
E3	T	T, Capital	X	X	X	X	X	X
E4	U	U, Capital	X	X	X	X	X	X
E5	V	V, Capital	X	X	X	X	X	X
E6	W	W, Capital	X	X	X	X	X	X
E7	X	X, Capital	X	X	X	X	X	X
E8	Y	Y, Capital	X	X	X	X	X	X
E9	Z	Z, Capital	X	X	X	X	X	X
EB	Ô	O Circumflex, Capital			X			
EC	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			

End of Appendix A

Appendix B. Common Structures

Introduction

This appendix contains the information about the SNA/DS, SNA/FS, and SNA/MS SNA Condition Report (SNACR). For more information on the SNACR, refer to the following books:

- *SNA/Distribution Services Reference*
- *SNA/File Services Reference*
- *SNA/Management Services Reference*

The format descriptions comprise two parts: header description tables and structure descriptions. A header description table contains the header information for each structure. A structure description contains a prose description of the structure, bit-level representations, and any presence rules or length restrictions associated with a particular structure.

Encoding Rules and Representations

The SNA Condition Report (SNACR) format is described in terms of encoded fields referred to as ″structures″ and the hierarchical relationship between these structures. In this document, the header description tables show each structure and its header. Elsewhere in this book, the header length is assumed not to be part of the overall structure length (e.g., *SNA_report_code*).

Structure Classifications

Fields and groupings of fields are known as structures. They are categorized in terms of their hierarchical position (″atomic,″ ″child,″ or ″parent″), the method by which their beginning and endings are determined, (length-bounded, delimited, or implied) and which kind of header is used to identify them (LT or LLID). Only certain combinations of characteristics are possible.

Length-bounded Structures

Length-bounded structures consist of a header and usually some following information. A header may be either two bytes in length, referred to as an ″LT″ (length and type), or four bytes in length, referred to as an ″LLID″ (length and GDS code point). In either case, the length byte(s) include the length of the header itself and the following information, if any.

Atomic Structures

In many cases, a structure consists only of its own header followed by data. These structures cannot be decomposed, and therefore they are called ″atomic.″ Atomic structures are always length-bounded and may have either LT or LLID headers.

Parent and Child Structures

Structures can contain other structures within them. The containing structure is known as a parent structure and the contained structures are known as children. These terms are relative, since a non-atomic child structure itself contains other structures and is a parent to them. Children of the same parent are siblings of each other. Parent structures may be length-bounded, delimited, or implied; and may be identified by LTs or LLIDs.

Length-Bounded Parent Structures

In this case, the parent structure has its own header, either an LT or an LLID. Its length includes the lengths of all its children plus the length of its own header. A length-bounded parent exists both as a logical grouping of its children and as an explicit encoded structure at its own encoding level.

Delimited Parent Structures

Sometimes it is convenient to define a group of related structures as existing within a parent structure without having that parent structure appear as a length-bounded structure in the message. The beginning and end of the parent are defined by its first and last children. These children are known as delimiters, the first child is the prefix delimiter and the last is the suffix delimiter. Delimiter children are length-bounded and must be present. They may be null, that is, with an LT of length=2 or an LLID of length=4. When the children′s headers are LTs, the parent is classified as a delimited LT structure. When they are LLIDs, the parent is a delimited LLID structure.

Implied Parent Structures

It is possible to define a set of related structures as children of a parent structure where the existence and boundaries of the parent are implied by the existence and order of certain child structures. This set of children may occur within the parent structure, either ordered or unordered, until a structure occurs that is not an element of this set. This break in sequence implies the boundary between parent structures. Depending on its children′s headers, an implied parent is classified as either implied LT or implied LLID.

Segmented Structures

Length-bounded LLID structures may be either segmentable or non-segmentable. For segmentable structures, the most significant bit of the LL bytes indicates whether any particular segment is the last (bit is equal to 0) or not last (bit is equal to 1) segment of the structure. The ID bytes of the segmentable structure are present on the first segment only.

Properties of Parent Structures

Order

A parent structure may have either ordered or unordered children. Ordered children occur in the parent structure in the same order as they are described in the format description table. Unordered children may occur in the parent structure in any order.

Unrecognized Children

Future enhancements to the formats might add structures that will not be recognized by implementations of the current format definitions. The current format must specify for each parent whether or not unrecognized child structures are allowed. If they are allowed, the definition must specify how long they might be. When unrecognized structures are found where they are allowed, they must be passed through without change at intermediate locations and gracefully ignored at final destinations. Unrecognized structures are identified by either LT or LLID headers, being of the same type as their siblings.

Number of Children

The number of children within a parent may range from a required minimum to an allowed maximum. For example, a parent might have several children, each defined with an occurrence of 0-1, and a number of children defined as 1. This means that any one, but only one, child is allowed.

Header Description Table

The header information and primary syntax associated with each structure are formally described in tabular form. These header description tables represent the formatting information required to either parse or build the SNACR.

Structure Name

The first column of the header description table identifies the SNACR structures, by name, and illustrates their hierarchical relationship by indentation of the column entries. The order of the structure entries in the table represents, unless specified otherwise, the order in which the structures appear in the SNACR datastream.

Structure Reference (Struct Ref)

As header information and primary syntax are described in the header description of a particular table, the semantics, bit representations, presence rules, and other characteristics are described formally in the structure description. This column contains a reference page number to where this structure information is found.

Structure Class (Struct Class)

Structures are classified as either length-bounded LLIDs (ID), length-bounded LTs (T), delimited LLIDs (Del-ID), delimited LTs (Del-T), implied LLIDs (Imp-ID), or implied LTs (Imp-T).

A structure classified as delimited must contain at least two required, length-bounded children that act as the prefix (pfx) and suffix (sfx) of the delimited structure. The ″/pfx″ notation indicates the length-bounded child structure that serves as the prefix for its parent delimited structure. The ″/sfx″ notation indicates the length-bounded structure that serves as the suffix for its parent delimited structure.

A structure classified as implied uses an identified child to identify the beginning of a sequence of children. The ″/idc″ notation indicates the length-bounded structure that serves as an identified child of its parent implied structure.

The ″/seg″ notation indicates that segmentation is allowed.

ID/T

This column contains the ID or T value within the header, in hexadecimal. To indicate that a delimited structure is identified by its prefix, the notation ″pfx″ is used. To indicate that an implied structure is identified by one of its children, the notation ″idc,″ for identified child, is used.

Length

This column describes the length verification that would be appropriate at presentation services time. The range of length values specifies the minimum and maximum lengths of structures which an implementation is required to receive. For structures that allow unrecognized children, the maximum length value accommodates the possibility of these yet-to-be-defined structures. On the sending side, the maximum length value for a particular structure may be determined by subtracting the unrecognized reserve, if unrecognized children are allowed, from the maximum length.

Note: An asterisk denotes length restrictions for a particular structure. Length restrictions are detailed in the corresponding structure description.

Occurrences

Multiple occurrences of the SNACR structures may or may not be permitted. A value of ″1 - <some number>″ in this column indicates the allowed range of occurrences of the corresponding structure. A value of ″≥1″ indicates that there is no architecturally defined maximum. A value of ″1″ in this column indicates that only a single instance of the corresponding structure is appropriate. A value of ″0 - 1″ indicates that an instance of the corresponding structure is optional.

Note: An asterisk denotes presence rules for a particular structure. Presence rules are detailed in the corresponding structure description.

Children

Unrecognized Children Allowed (Unrec): An entry of ″Y″ in the ″Unrec″ column indicates that the corresponding structure tolerates unrecognized child structures. An entry of ″N″ indicates that the particular structure tolerates only the architecturally-defined child structures. An entry of ″—″ indicates that unrecognized children are not applicable to the particular structure. By definition, atomic structures do not contain children, recognized or not.

Order: A value of ″Y″ in this column indicates that children are ordered, a value of ″N″ indicates that children are unordered, and a value of ″—″ indicates that no children are present.

Note: If a structure is atomic, this column is not applicable.

Number (Num): Each parent structure contains a certain number of different children. This column specifies the minimum and maximum number of different children for a particular parent structure. The maximum number also accounts for unrecognized children, if they are allowed within the parent structure. This column does not account for multiple occurrences of a particular child structure within the parent structure. The number of occurrences of each child is indicated in the ″Occurrences″ column.

Subtable: Sometimes the need to divide large tables into subtables becomes apparent, particularly when common children appear frequently within different header description tables. This column contains a reference page number to where these common children are described.

Structure Description

The structure description is referenced by a page number appearing in the ″Structure Reference″ column corresponding to each structure in the header description table. This description contains information pertaining to the data portion of a particular structure. Prose descriptions, presence rules, and semantics associated with the corresponding entry in the header description table may appear in the structure description.

Overview

The SNA Condition Report is a means of encoding exception information for any type of SNA exception. Information about the exception and the location of the exception can be encoded in the SNA Condition Report for the purpose of reporting. The SNA Report Code contains the code and subcode that describe the exception. The location information is encoded in standard structure identification of the Structure Report. When needed, other structures can also be included in the SNA Condition Report to add any other necessary information about the exception.

The SNA Condition Report consists of the following major parts:

SNA_Report_Code
: The SNA_Report_Code is a required SNA registered code identifying the condition that is being reported. The primary report code is placed in bytes 2-3 and the subcode is placed in bytes 4-5.

Structure_Report
: The structure_report contains information about the structure involved in a format related exception.

Reported-On_Dest_List
: Contains the portion of the distribution destinations that are being reported on.

Reported-On_Agent
: Contains the name of the transaction program that is being reported on.

Reported-On_Token_String
: Contains the canonical identifier of a data object related to the detected condition.

Supplemental_Report
: Contains other information about the exception that could not be encoded in any of the structures described above.

The combination of the SNA registered report code and a standard structure identifier creates a powerful tool for reporting format exceptions. Both the reason for the exception and its location in the data stream can be described in a general, independent fashion. This has a secondary benefit of allowing similar exceptions to be reported on with the same report code, even if the exceptions occurred in entirely different data streams.

Header Description Tables for SNA Condition Report

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table
SNA_Condition_Report	B-11	ID	1532	10-32749	0-1*	Y	Y	1-10	—
SNA_Report_Code	B-11	T	7D	6	1	—	—	—	—
Structure_Report	B-11	T	01	14-255	0-10*	Y	Y	2-10	—
Structure_State	B-12	T	01	3	1	—	—	—	—
Structure_Contents	B-12	T	02	3-100	0-1*	—	—	—	—
Parent_Spec	B-12	T	03	5-17	0-7	N	Y	1-4	—
Parent_ID_Or_T	B-12	T	01	3-4	1	—	—	—	—
Parent_Class	B-12	T	02	3	0-1*	—	—	—	—
Parent_Position	B-13	T	03	4	0-1	—	—	—	—
Parent_Instance	B-13	T	04	4	0-1	—	—	—	—
Structure_Spec	B-13	T	04	5-17	0-1*	N	Y	1-4	—
Structure_ID_Or_T	B-13	T	01	3-4	0-1*	—	—	—	—
Structure_Class	B-14	T	02	3	0-1*	—	—	—	—
Structure_Position	B-14	T	03	4	0-1	—	—	—	—
Structure_Instance	B-14	T	04	4	0-1	—	—	—	—
Structure_Segment_Number	B-14	T	05	4	0-1*	—	—	—	—
Structure_Byte_Offset	B-15	T	06	4	0-1	—	—	—	—
Sibling_List	B-15	T	07	3-100	0-1*	—	—	—	—
Unrecognized_Reserve	B-18	T	—	2-241	—	—	—	—	—
Reported-On_Dest_List	B-15	Del-T	pfx	12-11268	0-1*	N	Y	3	—
Reported-On_Dest_Prefix	B-15	T/pfx	08	2	1	—	—	—	—
Reported-On_Dest	B-15	Imp/T	idc	8-5654	≥1	N	Y	1-2	—
Reported-On_Location_Name	B-15	T/idc	09	2-22	1	N	Y	0-2	—
Reported-On_NETID	B-16	T	01	3-10	0-1*	—	—	—	—
Reported-On_Node_ID	B-16	T	02	3-10	0-1*	—	—	—	—
Reported-On_User	B-16	T	0A	8-22	≥0*	N	Y	2	—
Reported-On_Naming_Auth	B-17	T	01	3-10	1	—	—	—	—
Reported-On_Individual_ID	B-17	T	02	3-10	1	—	—	—	—
Reported-On_Dest_Suffix	B-17	T/sfx	0B	2	1	—	—	—	—
Reported-On_Agent	B-17	T	04	3-10	0-1*	—	—	—	—
Reported-On_Token_String	B-18	T	02	5-182	0-10*	N	Y	1-10	B-10
Supplemental_Report	B-18	T	03	3-255	0-5*	—	—	—	—
Unrecognized_Reserve	B-18	T	—	2-15826	—	—	—	—	—

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

Figure B-1. SNA_Condition_Report as Defined by SNA/FS and CM

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Tokens									
First_Token	B-18	T	01	3-18	1	—	—	—	—
Second_Token	B-18	T	02	3-18	0-1	—	—	—	—
Third_Token	B-19	T	03	3-18	0-1	—	—	—	—
Fourth_Token	B-19	T	04	3-18	0-1	—	—	—	—
Fifth_Token	B-19	T	05	3-18	0-1	—	—	—	—
Sixth_Token	B-19	T	06	3-18	0-1	—	—	—	—
Seventh_Token	B-19	T	07	3-18	0-1	—	—	—	—
Eighth_Token	B-19	T	08	3-18	0-1	—	—	—	—
Ninth_Token	B-19	T	09	3-18	0-1	—	—	—	—
Tenth_Token	B-19	T	0A	3-18	0-1	—	—	—	—

Figure B-2. Subtable Encoding of the Global Name Tokens.

SNA_Condition_Report

Description: The *SNA_condition_report* describes the condition being reported. The condition is always identified by an *SNA_report_code*.

Certain conditions can be more fully described by supplementary information. Conditions pertaining to one or more structures in a format can have the location and contents of each of those structures specified by a *structure_report*. Certain conditions arise from inconsistencies among multiple portions of the MU. Each portion is described by a separate *structure_report*.

Data objects related to the reported-on condition can be specified in a *reported-on_token_string*. Other information related to the condition can be specified in a *supplemental_report*.

Presence Rule: Occurs when a reportable condition was detected by the agent/server and the agent has determined that reporting is appropriate.

SNA_Report_Code

Description: The *SNA_report_code* is an SNA registered code identifying the condition that is being reported. Refer to the ″Sense Data″ chapter in the *SNA/Format* book for allowable values and descriptions.

Format: Byte string

Byte	Content
0-1	LT header
2-3	Primary report code
4-5	Subcode

Structure_Report

Description: The *structure_report* reports on a structure involved in a format-related condition. Depending on the condition, the *structure_report* may describe a structure that was present in, or absent from, the reported-on MU.

A format condition has its location in the MU pinpointed by a *structure_spec* and a list of *parent_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: The presence or absence of this structure is governed by the using architecture.

Structure_State

Description:	The *structure_state* indicates whether the reported-on structure was present or absent.
Format:	Hexadecimal code

Byte	Content
0-1	LT header
2	X′01′ STRUCTURE_PRESENT
	X′02′ STRUCTURE_ABSENT

Note: All other values are reserved.

Structure_Contents

Description:	The *structure_contents* is the portion of the MU that is relevant to the detected condition. Typically, the *structure_contents* contains the header of the structure and at least the beginning of its contents. When the condition can be isolated to a portion of the structure, the *structure_contents* contains only that portion of the structure relevant to the condition. In this case, the *structure_segment_number* and *structure_byte_offset* locate the portion of the structure relevant to the condition.
Presence Rule:	Allowed only when *structure_state* = STRUCTURE_PRESENT.
Format:	Undefined byte string

Parent_Spec

Description:	The *parent_specification* contains the identifier (ID or T) and the class of a parent structure. For a parent structure that occurs multiple times, the instance may also be included. The value of the *parent_instance* identifies the particular instance. The position of this parent structure within its parent (if one exists) may also be included. This would typically be done when this parent structure is an unordered child of its parent.

Parent_ID_Or_T

Description:	The *parent_ID_or_T* is the ID or T value of a parent structure. ID values are the registered GDS code points. T values are architecture-specific values relative to the encompassing ID.
Format:	Undefined byte string

Parent_Class

Description:	The *parent_class* is the class of a parent structure.
Presence Rule:	If absent, defaults to LENGTH-BOUNDED_LT_STRUCTURE.
Format:	Hexadecimal code

Byte	Content	
0-1	LT header	
2	X′01′	LENGTH-BOUNDED_LLID_STRUCTURE (ID)
	X′02′	LENGTH-BOUNDED_LT_STRUCTURE (T) (default)
	X′03′	DELIMITED_LLID_STRUCTURE (DEL-ID)
	X′04′	DELIMITED_LT_STRUCTURE (DEL-T)
	X′05′	IMPLIED_LLID_STRUCTURE (IMP-ID)
	X′06′	IMPLIED_LT_STRUCTURE (IMP-T)

Note: All other values are reserved.

Parent_Position

Description: The *parent_position* is the position of this parent structure within its parent (if one exists) in this particular MU. Multiple consecutive instances of a repeatable parent structure share a single position, and can be distinguished by *parent_instance*.

Format: Signed binary integer

Parent_Instance

Description: The *parent_instance* is used when a parent structure occurs multiple times. The value of *parent_instance* identifies the particular instance within a position.

Format: Signed binary integer

Structure_Spec

Description: The *structure_specification* contains the identifier (ID or T) and the class of a structure. For a structure that occurs multiple times, the instance may also be included. The value of the *structure_instance* identifies the particular instance. The position of this structure within its parent structure may also be included. This would typically be done when the parent structure contains unordered children.

Presence Rule: Absent only when the *structure_class* is the default and the *sibling_list* contains all pertinent ID or T values.

Structure_ID_Or_T

Description: The *structure_ID_or_T* is the ID or T value of the structure. ID values are the registered GDS code points. T values are architecture-specific values relative to the encompassing ID.

Presence Rule: Required except when *sibling_list* contains all pertinent ID or T values. In this case, the structures specified by *sibling_list* are the structures being reported on.

Format: Undefined byte string

Structure_Class

Description:	The *structure_class* is the class of the reported-on structure and any siblings identified in *sibling_list*.
Presence Rule:	If absent, defaults to LENGTH-BOUNDED_LT_STRUCTURE.
Format:	Hexadecimal code

Byte	Content	
0-1	LT header	
2	X′01′	LENGTH-BOUNDED_LLID_STRUCTURE (ID)
	X′02′	LENGTH-BOUNDED_LT_STRUCTURE (T) (default)
	X′03′	DELIMITED_LLID_STRUCTURE (DEL-ID)
	X′04′	DELIMITED_LT_STRUCTURE (DEL-T)
	X′05′	IMPLIED_LLID_STRUCTURE (IMP-ID)
	X′06′	IMPLIED_LT_STRUCTURE (IMP-T)

Note: All other values are reserved.

Structure_Position

Description:	The *structure_position* is either the actual or expected position of this structure within its parent in this particular MU. Multiple consecutive instances of a repeatable structure share a single position, and can be distinguished by *structure_instance*.
Format:	Signed binary integer (1-origin)

Structure_Instance

Description:	The *structure_instance* is used when the structure is one of multiple occurrences of a repeatable structure. The value of *structure_instance* identifies the particular instance within a position.
Format:	Signed binary integer (1-origin)

Structure_Segment_Number

Description:	The *structure_segment_number* is the segment of the structure in which the condition was detected.
Presence Rule:	Occurs when the beginning of *structure_contents* was not contained in the first segment of the reported-on structure.
Format:	Signed binary integer (1-origin)

Structure_Byte_Offset

Description: The *structure_byte_offset* marks the start of *structure_contents* within the reported-on structure. If *structure_segment_number* is present, this value is the offset from the start of the indicated segment; otherwise, it is the offset from the beginning of the structure.

Format: Signed binary integer (0-origin)

Sibling_List

Description: The *sibling_list* contains a string of ID or T values necessary to describe the detected condition. The structures identified in *sibling_list* are children of the parent identified in *parent_spec* and/or siblings of the structure identified in *structure_spec*. The class of the sibling structures is the same as *structure_class*. The expected position, when applicable, is given by *structure_position*.

Presence Rule: The presence or absence of this structure is governed by the using architecture.

Format: Byte string

Reported-On_Dest_List

Description: The *reported-on_destination_list* contains the portion of the distribution destinations that are being reported on.

Presence Rule: The presence or absence of this structure is governed by the using architecture.

Reported-On_Dest_Prefix

Description: The *reported-on_destination_prefix* is the prefix of the *reported-on_destination_list*.

Reported-On_Dest

Description: The *reported-on_destination* associates *reported-on_user*s with a *reported-on_location_name* for those destinations specified in the original distribution request being reported on. For flat destination lists (i.e., lists containing only location names and/or location-user pairs), there are zero or one user names per location list. For factored destination lists, there can be multiple user names per location list.

Reported-On_Location_Name

Description: The *reported-on_location_name* is one of the original destination locations being reported on.

Reported-On_NETID

Description:	The *reported-on_NETID* is the first part of the name of one of the original destination locations being reported on.
Presence Rule:	Always present, unless the *reported-on_location_name* has passed through a SNA/DS FS1 node (or nodes), in which case the *reported-on_user* is present.
Format:	Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

Reported-On_Node_ID

Description:	The *reported-on_Node_ID* is the second part of the name of one of the original destination locations being reported on.
Presence Rule:	Always present, unless the *reported-on_location_name* has passed through a SNA/DS FS1 node (or nodes), in which case the *reported-on_user* is present.
Format:	Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

Reported-On_User

Description:	The *reported-on_user* is the name of one of the original destination users being reported on.
Presence Rule:	Required when the *reported-on_NETID reported-on_Node_ID* are not present.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

Reported-On_Naming_Auth

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

Format: Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Reported-On_Individual_ID

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

Format: Character string

CGCSGID: 01134-00500 (character set AR)

String Conventions: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

Reported-On_Dest_Suffix

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

Reported-On_Agent

Description. The *reported-on_agent* is the name of the transaction program that is being reported on.

Presence Rules: Present if needed to identify the reported-on event and when not implied by the context.

Format Character string, except for the first byte.

CGCSGID: 01134-00500 (Character Set AR)

String Convention: Leading and imbedded blanks are not allowed; however trailing blanks are allowed.

The first byte of an SNA-registered transaction program name ranges in value from X′00 to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA-registered. X′40′ is not a valid first-byte value.

Reported-On_Token_String

Description: The *reported-on_token_string* contains the SNA/FS canonical identifier of a data object related to the detected condition.

Presence Rule: The presence or absence of this structure is governed by the using architecture.

Supplemental_Report

Description: The *supplemental_report* contains other information pertaining to a condition. The contents of the *supplemental_report* are governed by the using architecture.

Presence Rule: The presence or absence of this structure is governed by the using architecture.

Unrecognized_Reserve

Description: The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure.

Format: Undefined byte string

First_Token

Description: The *first_token* is the highest level part of the data object name. Its values are assigned and registered by SNA.

Format: Character string

CGCSGID: 01134-00500 (Character Set AR)

String Conventions: Leading, imbedded, and trailing space (X′40′) characters are not allowed.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

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.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

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.

Note: In existing networks where network IDs are defined using SNA character set A (includes character set AR, plus the special characters @, #, and $), the RGN may contain any of the three special characters; however, these characters may not be available on keyboards in every country and should not be used in new network IDs.

Unit of Work Correlator—Overview

The Unit of Work Correlator carries enough information to be a network-wide correlator. It is used in situations where multiple requests may be sent over the network before replies are received. In such cases the responding agent uses the Unit of Work Correlator to identify the request that each reply corresponds to, thus allowing the requesting agent to match the responses to the requests.

The Unit of Work Correlator consists of the following parts:

Requester_Location_Name
The name of the location where the request originated.

Requester_User
The user name of the requester.

Requester_Agent
The transaction program that originated the request.

Seqno_DTM
The sequence number assigned to the request by the requesting agent.

Header Description Tables

Unit of Work Correlator

Figure B-3. Unit of Work Correlator as defined by SNA/FS and SNA/MS

Structure Name	Struct Ref Pg	Struct Class	ID/T	Length	Occur-rences	Children			
						Unrec	Order	Num	Sub Table Page
Agent_Unit_of_Work	B-22	ID	1549	27-128	1	Y	Y	2-8	—
Requester_Location_Name	B-22	T	01	8-22	1	N	Y	2	—
Requester_Netid	B-22	T	01	3-10	1	—	—	—	—
Requester_Node_ID	B-22	T	02	3-10	1	—	—	—	—
Requester_User	B-22	T	03	8-22	0-1	N	Y	2	—
Requester_Naming_Auth_ID	B-23	T	01	3-10	1	—	—	—	—
Requester_Individual_ID	B-23	T	02	3-10	1	—	—	—	—
Requester_Agent	B-23	T	04	3-10	0-1*	—	—	—	—
Seqno_DTM	B-23	T	02	15-17	1	—	—	—	—
Unrecognized_Reserve	B-25	T	—	2-53	—	—	—	—	—

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

Structure Descriptions

Agent_Unit_of_Work

Description: The *agent_unit_of_work*, assigned by the requesting agent, provides the basis to track the progress of a particular defined task. The unit-of-work request is uniquely identified by the combination of *requester_location_name*, *requester_user*, *requester_agent*, and *sequence_number/date-time*.

Requester_Location_Name

Description: The *requester_location_name* is the name of the location at which the unit-of-work was requested.

Requester_Netid

Description: The *requester_netid* is the first part of the name of the location at which the unit-of-work was requested.

Format: Character string

CGCSGID: 01134-00500 (Character Set AR)

String Conventions: Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_Node_ID

Description: The *requester_node_ID* is the second part of the name of the location at which the unit-of-work was requested.

Format: Character string

CGCSGID: 01134-00500 (Character Set AR)

String Conventions: Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_User

Description: The *requester_user* is the user name of the originator of the unit-of-work request.

Requester_Naming_Auth_ID

Description: The *requester_naming_authority_ID* is the first part of the user name of the unit-of-work originator.

Format: Character string

CGCSGID: 01134-00500 (Character Set AR)

String Conventions: Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_Individual_ID

Description: The *requester_individual_ID* is the second part of the user name of the unit-of-work originator.

Format: Character string

CGCSGID: 01134-00500 (Character Set AR)

String Conventions: Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

Requester_Agent

Description: The *requester_agent* identifies the transaction program that originated the unit-of-work request.

Presence Rule: When the *requester_agent* is absent, the originating agent specified in the distribution is the default.

Format: Character string, except for first byte

CGCSGID: 01134-00500 (Character Set AR)

String Convention: Leading and imbedded blanks are not allowed; however, trailing blanks are allowed.

The first byte of an SNA-registered transaction program name ranges in value from X′00 to X′3F′. When the first byte ranges in value from X′41′ to X′FF′, the transaction program is not SNA-registered. X′40′ is not a valid first-byte value.

Seqno_DTM

Description: The sequence number is the number assigned to the unit-of-work request by the originating agent. The value ranges from 1 to (2^{31})-1. The date of the unit-of-work request is assigned by the *requester_agent*; the time of the unit-of-work request is assigned by the *requester_location_name*. The offset from GMT for local time is included.

Format: Byte string

Byte	Contents
0-1	LT header
2-5	Sequence number Signed binary integer limited to (2^{31})-1.
	DATE
6-7	Year, in binary (e.g., year 1989 is encoded as X′07C5′)
8	Month of the year, in binary (values from 1 to 12 are valid)
9	Day of the month, in binary (values from 1 to 31 are valid)
	TIME
10	Hour of the day, in binary (values from 0 to 23 are valid)
11	Minute of the hour, in binary (values from 0 to 59 are valid)
12	Second of the minute, in binary (values from 0 to 59 are valid)
13	Hundredth of the second, in binary (values from 0 to 99 are valid)
	GMT FLAG
14	Indicates that specified TIME is GMT and identifies whether offsets from GMT are required to calculate local time. (Equivalent EBCDIC characters are shown in parentheses.) X′E9′ (Z) no offset required X′4E′ (+) add required offset to GMT to get local time X′60′ (−) subtract required offset from GMT to get local time
	OFFSET
15	Hour offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 23 are valid)
16	Minute offset from GMT in binary, occurs when *GMT_flag* ≠ Z (values from 0 to 59 are valid)

Examples

A 9-byte date/time encoding is a date/time followed immediately by an EBCDIC ′Z′, and is considered to be GMT. Thus, 12:00 GMT on 2 January 1988 would be

```
X′07C401020C000000E9′
  yyyymmddhhmmsshhZ
```

An 11-byte date/time encoding is a date/time followed immediately by an EBCDIC ′+′ or ′-′ and two one-byte binary numbers, and is considered to be GMT and the offset from GMT to local time. Thus, 7:00 a.m. on 2 January 1988 in New York would be 12:00 GMT - 5 hours, or

```
X′07C401020C000000600500′
  yyyymmddhhmmsshh- hhmm
```

Unrecognized_Reserve

Description:	The *unrecognized_reserve* is the number of bytes reserved for unrecognized structures. An unrecognized structure occurs within its parent structure. The number of unrecognized structures allowable for a particular parent structure is limited by the number of children allowable for that parent structure.
Format:	Undefined byte string

End of Appendix B

Communicating Your Comments to IBM

Systems Network Architecture
Formats

Publication No. GA27-3136-18

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers′ comment form (RCF) from a country other than the United States, you can give the RCF to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.
- If you prefer to send comments electronically, use this network ID:

  Internet: **appn@us.ibm.com**

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

Help us help you!

Systems Network Architecture Formats

Publication No. GA27-3136-18

We hope you find this publication useful, readable and technically accurate, but only you can tell us! Your comments and suggestions will help us improve our technical publications. Please take a few minutes to let us know what you think by completing this form.

Overall, how satisfied are you with the information in this book?	Satisfied	Dissatisfied
	•	•

How satisfied are you that the information in this book is:	Satisfied	Dissatisfied
Accurate	•	•
Complete	•	•
Easy to find	•	•
Easy to understand	•	•
Well organized	•	•
Applicable to your task	•	•

Specific Comments or Problems:

Please tell us how we can improve this book:

Thank you for your response. When you send information to IBM, you grant IBM the right to use or distribute the information without incurring any obligation to you. You of course retain the right to use the information in any way you choose.

Name

Address

Company or Organization

Phone No.

Help us help you!
GA27-3136-18

IBM®

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Networking Software
Department BRQA/Building 502
PO Box 12195
Research Triangle Park, North Carolina 27709-9990

Fold and Tape **Please do not staple** Fold and Tape

GA27-3136-18

Cut or Fold
Along Line

=Common=Structures=

IBM®

Printed in U.S.A.

GA27-3136-18