GC20-1868-2

Systems

Systems Network Architecture - Sessions Between Logical Units

This book describes Systems Network Architecture (SNA) as it relates to sessions between logical units (LUs). LUs are the ports through which application programs, terminals, and terminal operators communicate across an SNA network. LU to LU sessions support communication between these end-users.

This book describes the structure of an LU and defines the characteristics of five types of LU to LU sessions (types 1 through 4, and type 6). It also describes the data streams that are valid for each type of LU to LU session.

This book does not describe any specific equipment or programs that may implement SNA, nor does it describe any implementation subsets or deviations from the architecture. These matters, as well as information on product installation and system definition, are described in implementation literature for the applicable products.

This book is intended for systems programmers and others who need detailed information about SNA in order to develop or adapt a product or program to function within an SNA network.

An understanding of SNA concepts is assumed throughout this book. The basic concepts can be obtained by reading *SNA Concepts and Products* (GC30-3072).

Third Edition (April 1981)

This is a major revision of, and obsoletes, GC20-1868-1. Changes in this third edition are extensive; a summary of these changes is included in the Preface. In addition, the second-edition title, *Systems Network Architecture, Logical Unit Types,* has been changed.

Information in this manual is subject to further changes; any such changes will be published in subsequent revisions or Technical Newsletters. Before using this manual in connection with the operation of IBM systems or equipment, refer to the latest *IBM System/370 Bibliography* (GC20-0001), and *IBM System/370 Bibliography of Industry Systems and Application Programs* (GC30-0370) to find out which edition is applicable.

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.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for readers' comments has been provided at the back of this publication. If this form has been removed, address comments concerning the contents of this publication to IBM Corporation, Technical Publications, Dept. 824, 1133 Westchester Avenue, White Plains, New York 10604. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

PREFACE

This book describes Systems Network Architecture (SNA) as it relates to sessions between logical units (LUs). LUs are the ports through which application programs, terminals, and terminal operators communicate across an SNA network. LU to LU sessions support communication between these end users.

This book shows the structure of an LU and defines the characteristics of five types of LU to LU sessions. It also discusses the data streams that are valid for each type of LU to LU session.

This book does not describe any specific equipment or programs that may implement SNA, nor does it describe any implementation subsets or deviations from the architecture. These matters, as well as information on product installation and system definition, are described in implementation literature for the applicable products.

Intended Audience

This book is intended for systems programmers and others who need detailed information about SNA in order to develop or adapt a product or program to function within an SNA network.

An understanding of SNA concepts is assumed throughout this book. The basic concepts can be obtained by reading SNA Concepts and Products. (See "Related Publications" below for an order number.)

Intended System Level

Systems Network Architecture is independent of product schedules and levels. Extensions and modifications to SNA will be described in future editions of this book, however, as SNA-based products are released.

HOW THIS BOOK IS ORGANIZED

This book is organized into two parts and their supporting appendixes. Part 1 describes the types of LU to LU sessions. Part 2 describes the elements of the data streams used by LU to LU sessions.

Part 1 Chapter 1, "Introduction," describes the types of LU-LU sessions that are allowed and the structure of the LU.

Part 1 Chapters 2 through 6 describe LU-LU sessions types 1, 2, 3, 4, and 6. The characteristics of these LU-LU sessions are discussed, and activation parameters associated with their BIND SESSION requests are given.

Part 2 Chapter 1, "SNA Character String (SCS)," describes the control functions for SCS controls used by LU-LU session types 1 and 4.

Part 2 Chapter 2, "SNA 3270 Data Stream," describes the data stream used by LU-LU sessions types 2 and 3.

Part 2 Chapter 3, "Structured Fields," describes the structured fields used by LU-LU session type 1. Structured fields used by LU-LU session types 2 and 3 are described in 3270 Data Stream Programmer's Reference.

Part 2 Chapter 4, "Function Management Headers," describes the FM headers used by LU-LU session types and how they are used to select destinations and manage data.

Part 2 Chapter 5, "Compression and Compaction," describes how data can be compressed or compacted using string control bytes (SCBs) on LU-LU session types 1 and 4.

RELATED PUBLICATIONS

The following publications are related to this book.

Systems Network Architecture: Concepts and Products, GC30-3072, introduces you to SNA. This book helps you develop a basic understanding of an SNA network.

Systems Network Architecture: Technical Overview, GC30-3073 (when available), introduces you to the functions performed by SNA products. Information is organized on a function-by-function basis rather than on a component-by-component basis. It does not describe SNA in the detail required for design or development of products.

3270 Data Stream Programmer's Reference, GA23-0059. This book describes the SNA 3270 data stream, which is used by LU-LU session types 2 and 3.

Systems Network Architecture: Reference Summary, GA27-3136, is a handbook that provides selected reference information. The handbook contains SNA formats and protocols that you need most frequently when using manuals on specific IBM SNA products.

Systems Network Architecture, Format and Protocol Reference Manual: Architecture Logic, SC30-3112, is a comprehensive reference book containing the formats and protocols of SNA from a design viewpoint.

IBM Cryptographic Subsystem Concepts and Facilities, GC22-9063. This book describes the cryptographic facility used by SNA.

SUMMARY OF AMENDMENTS

Part 1 Chapter 1 - Introduction

This chapter was rewritten to provide added detail on LU structure.

Part 1 Chapter 2 - LU-LU Session Type 1

LU-LU session type 1 now allows data streams that contain structured fields and the Set Attribute (X'28') SNA character string control. These enhancements enable data streams to carry color, highlighting, and character set information on a type 1 session.

Part 1 Chapter 3 - LU-LU Session Type 2

LU-LU session type 2 now allows options with the SNA 3270 data stream that carry color, highlighting, and character set information. The data stream also allows for partitioning of the display screen, scrolling of information within those partitions, and validating of fields.

Part 1 Chapter 4 - LU-LU Session Type 3

LU-LU session type 3 now allows options with the SNA 3270 data stream that carry color, highlighting, and character set information.

Part 1 Chapter 5 - LU-LU Session Type 4

LU-LU session type 4 remains unchanged except for a few changes made for technical clarity.

Part 1 Chapter 6 - LU-LU Session Type 6

This is a new chapter with this edition.

Part 2 Chapter 1 - SNA Character String (SCS)

Additional SCS controls have been added.

Part 2 Chapter 2 - SNA 3270 Data Stream

The SNA 3270 data stream now can carry color, highlighting, and character set information. The data stream also allows for partitioning of the display screen, scrolling of information within those partitions, and validating of fields.

This chapter no longer contains the formats and protocols of the data stream. For information on formats and protocols, see 3270 Data Stream Programmer's Reference.

Part 2 Chapter 3 - Structured Fields

This is a new chapter with this edition. It contains structured fields used by LU-LU session type 1. For structured fields used by LU-LU session types 2 and 3, see 3270 Data Stream Programmer's Reference.

Part 2 Chapter 4 - Function Management Headers (FMHs)

FM headers used on LU-LU session type 6 have been added.

Part 2 Chapter 5 - String Control Bytes (SCBs)

This chapter remains unchanged except for a few changes made for technical clarity.

This chapter briefly describes Systems Network Architecture (SNA) as it relates to the structure of logical units (LUs). It summarizes some of the information found in SNA Concepts and Products and introduces the structure of an LU.

SNA networks provide two broad categories of services, as shown in Figure 1-1. One category is the services of the path control network, which consists of the path control and data link control components shown in Figure 1-1. These services fulfill the fundamental purpose of any network: to transmit data quickly and accurately between network locations, regardless of how distant they are from one another. Services of the path control network are described in SNA Concepts and Products.

The other category is network addressable unit (NAU) services, which handle the information passed between end users across the path control network. This category also includes services that allow the network to coordinate its activities, such as resource allocation, through the network nodes.

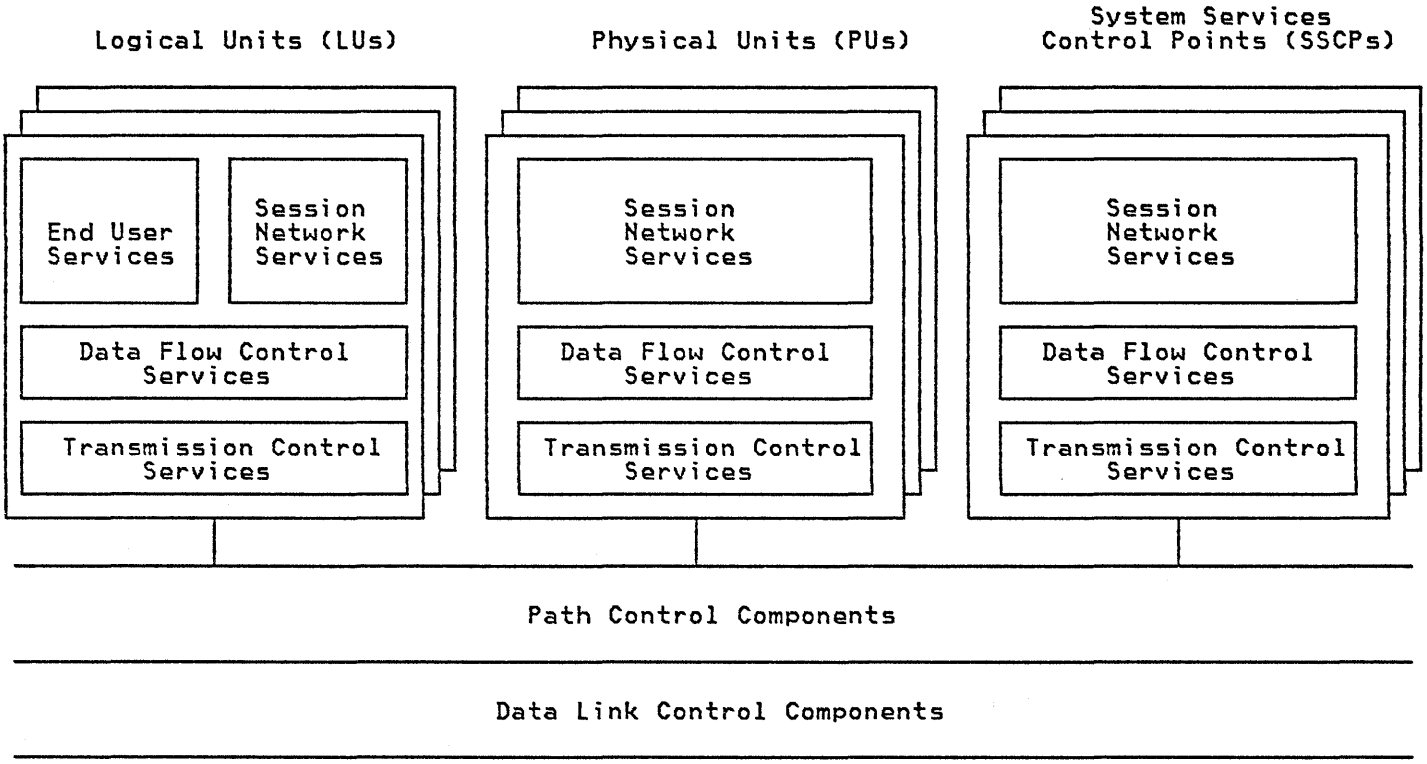


Figure 1-1. Distribution of services with an SNA network.

NAU services used in exchanging data between end users are referred to as end-user services. These are the principal services provided by logical units. NAU services that allow the network to coordinate its activities are called session network services. Each network addressable unit also contains data flow control services and transmission control services, which lie between the other NAU services and the path control network. Figure 1-1 shows the services mentioned, and SNA Concepts and Products has introductory material on each service.

Systems Network Architecture is structured so that system designers can build systems with interchangeable components and network users can share common function. One common function is the path control network. Each user has

access to the path control network through a logical unit tailored to the user. One user can talk to another user when one logical unit is "connected" to the other logical unit through the path control network.

The session between two end users is called an LU-LU session. In order to discuss network functions conveniently, an LU-LU session has three major categories (Figure 1-2):

- Application. A terminal operator, user application program, or device medium. This category is called the end user throughout this book.
- Logical unit (LU). The LU handles the way data is presented to the end user and the flow of data into the path control network.
- Path control network. The path control network coordinates the transmission of data over the telecommunications links.

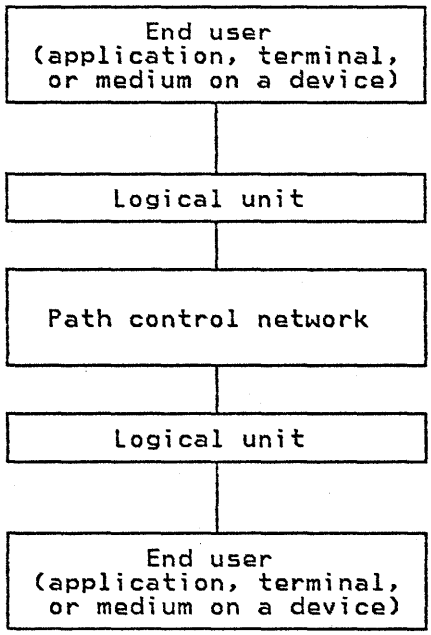


Figure 1-2. LU-LU session categories. The LU-LU session is divided into three categories: end user, logical unit (LU), and path control network. In some LU-LU sessions, the second end user is replaced by a service within the LU.

Logical units are the ports (or addresses) through which users gain access to the path control network. This chapter describes the structure of logical units and shows how they relate to the other components in the network.

Note: Logical units are one form of network addressable unit (NAU). The other NAUs, the physical unit and the system services control point, are introduced in SNA Concepts and Products and described in detail in SNA Format and Protocol Reference Manual: Architecture Logic.

SESSIONS

One end user talks to another by establishing a session. A session is a temporary logical connection between NAUs or, in the only case covered in detail by this book, between logical units. An LU-to-LU (LU-LU) session is started when one LU, called the primary LU or BIND sender, sends a BIND SESSION request to another LU, called the secondary LU or BIND receiver. The BIND SESSION request (plus the ensuing responses) establishes this formally bound pairing of LUs. The BIND SESSION request also defines the capabilities of the session partners and the protocols they can use during the session.

Each LU represents its session partner with a group of functions, called a half-session. Two formally bound half-sessions make a session.

The LU must establish a session with the system services control point (SSCP) if it wants to operate within the domain of that SSCP. This session is called the SSCP-LU session. The SSCP-LU session is not apparent to end users except when they require network services. (For detailed information on the SSCP-LU session, see SNA Format and Protocol Reference Manual: Architecture Logic.)

The LU also establishes sessions with other LUs on behalf of its end users. Each implementation defines the maximum number of concurrent sessions that are allowed. For example, Figure 1-3 shows an LU in session with an SSCP and three LUs.

The LU services manager controls the end user activities for each half-session activated by the LU. It is the LU component that is responsible for sending data to and receiving data from the end user. It is also responsible for activating sessions, managing the resources of the LU, and handling recovery and restart management.

TYPES OF LU-LU SESSIONS

SNA defines more functions than are needed or can be used by some end users, whether the end user is an application program, a terminal alone, or a terminal and operator. As a result, some LU functions are mandatory and others are optional.

The options are selected by specifying a profile number and by establishing usage criteria in the BIND SESSION request that activates the session. By grouping these profiles to match the three components of the half-session, one obtains the LU-LU session types that are discussed in this book. Figure 1-3 shows the components of the half-session and their related profiles, and Figure 1-4 shows how parameters of the BIND SESSION request are used to select options of those components. Figure 1-5 shows the valid profile numbers for each type of LU-LU session. The SNA Reference Summary lists each profile by profile number, identifies what SNA requests are valid in each profile, and highlights the protocols that each profile allows.

This manual describes LU-LU session types 1, 2, 3, 4, and 6 at a level of detail that allows users of IBM products to understand the architecture underlying those products. LU-LU session type 0 is not described outside of this chapter. Type 0 implementations may use any format or protocol defined by SNA, and may supplement SNA formats and protocols with implementation-defined formats and protocols. They may not replace SNA formats and protocols with implementation-defined formats and protocols.

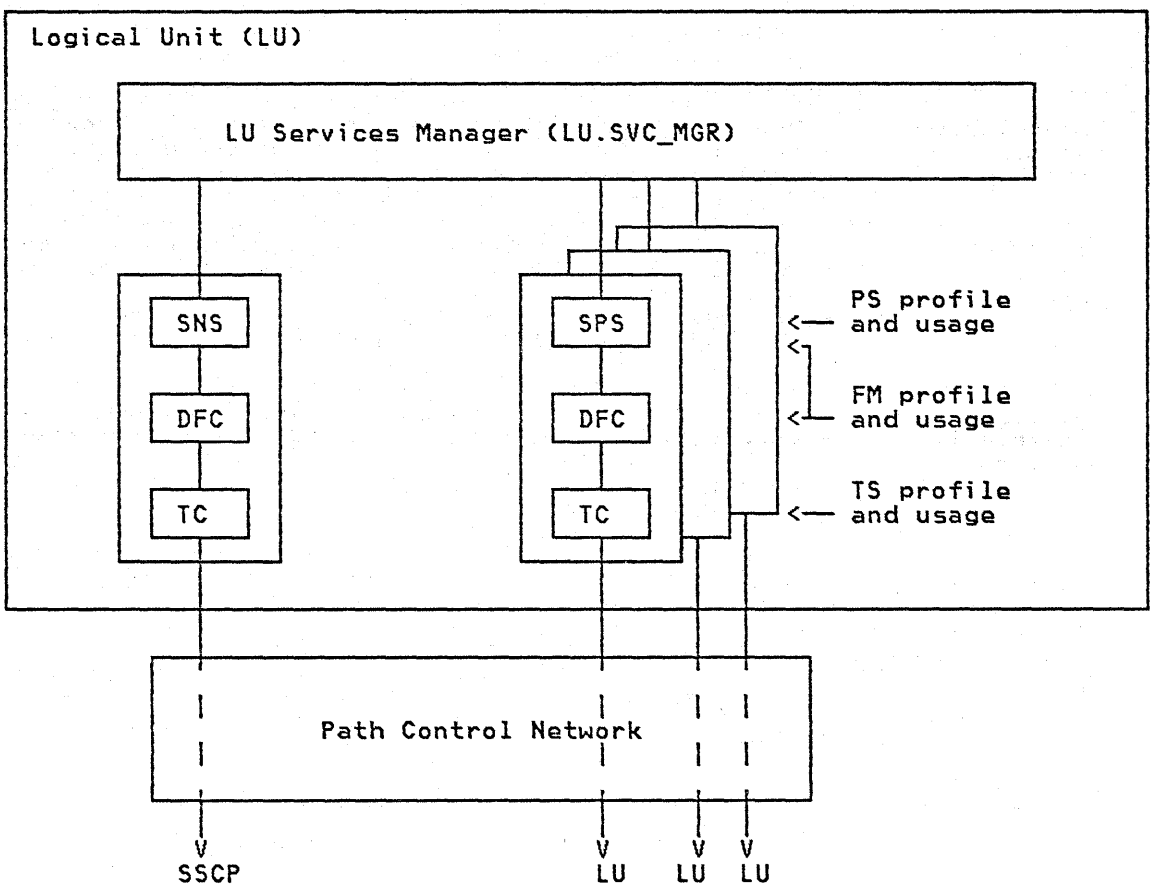


Figure 1-3. Example of LU_LU and LU-SSCP sessions. An LU comprises an LU services manager and one or more half-sessions. It handles the way data is presented to the end user. A half-session comprises either session network services (SNS) or session presentation services (SPS), plus data flow control (DFC) and transmission control (TC). Each half-session is connected through the path control network to another half-session, either in the system services control point (SSCP) or in another logical unit.

Half-session component	Selectable functions (not a complete list)	BIND SESSION fields used
Transmission control (TC)	• Request unit (RU) size • Pacing • Session control requests • Cryptography options	Transmission subsystem (TS) profile and usage
Data flow control (DFC)	• Request and response modes • Half-duplex or full-duplex send-receive mode • Brackets and chaining rules • Data flow control techniques and requests	Function management (FM) profile and usage
Session presentation services (SPS)	• SNA character string usage • Code repertoire (EBCDIC, ASCII, other) • Attended or unattended mode • FM header usage	Presentation services (PS) profile and usage

Figure 1-4. Use of BIND SESSION request parameters. Each LU has three half-session components that support the end user. The profile and usage parameters of the BIND SESSION request are used to define the half-session functions desired.

PS profile (session type)	TS profile	FM profile	PS characteristics
0	3,4,7	2,3,4,7,18	Any option desired
1	3,4	3,4	Data streams based on SNA character string or structured fields FM headers (none, or one or more of FMH-1, FMH-2, FMH-3) Data processing media support
2	3	3	SNA 3270 data stream No FM headers Display support
3	3	3	SNA 3270 data stream No FM headers Printer support
4	7	7	Data streams based on SNA character string FM headers (none, or FMH-1, FMH-2, FMH-3) Data processing and word processing media support
6	4,7	18	Data streams are user defined FM Headers (FMH-4, FMH-5, FMH-6, FMH-7, and FMH-10 optional) Data processing media support

Figure 1-5. LU-LU session types. The allowable subsets that can be specified with the TS (transmission services), FM (function management), and PS (presentation services) profiles are shown.

STRUCTURE OF AN LU

An LU is composed of components that provide services to the end user as well as components that interact with the path control network. The components that provide end-user services are the LU services manager (LU.SVC_MGR) and session presentation services (SPS) component of the half-session (see Figure 1-3). Although there is only one LU services manager, there can be as many half-sessions as the resources of the LU permit.

For LU-LU sessions, the half-session provides session presentation services (SPS). For the SSCP-LU session, the half-session provides session network services (SNS).

There is a common protocol at the boundary between DFC (data flow control) and SPS and SNS. That is, DFC doesn't know whether it has the SPS or SNS component above it. DFC controls the flow of function management data (FMD) requests and responses between FMD pairs (pairs of SNS components on an SSCP-LU session or pairs of SPS components on an LU-LU session).

Note: FMD is a term used to group the two components that handle function management data, SPS and SNS. This book does not use FMD when use of SNS or SPS is more precise, but you will see FMD used in the books shown as related publications in the Preface.

The TC (transmission control) component sends and receives data across the path control network.

The remainder of this chapter describes the functions performed by each component within the LU. You should refer to Figure 1-3 to see how each component relates to the overall structure.

STRUCTURE OF THE LU SERVICES MANAGER (LU.SVC_MGR)

The LU services manager activates and deactivates sessions, provides network services to the end user, and controls end user interactions with the half-sessions so that messages from different sessions are not arbitrarily interleaved. The LU services manager has overall recovery and restart responsibilities for the LU, but these responsibilities are shared by all components.

A half-session control block (HSCB) exists for each half-session that is not reset (in reset state). The HSCB provides storage for all the finite state machines (FSMs), queues, lists, and variables used by the half-session. A finite state machine is logic that compares an item of input (a search argument) against the current state of the system (in our case, the session state) and produces an item of output (a result), a state change, or both.

Included in each HSCB is a set of variables called the half-session activation parameters (HSAP). The LU services manager fills the first portion of the HSAP with that portion of the BIND SESSION request containing the TS, FM, and PS profiles and usage fields so that the LU has access to the protocols and capabilities of the half-session.

The LU services manager is also responsible for initializing the states of the half-session being activated. The three half-session components (SPS, DFC, and TC) are initialized to the reset state.

COMPONENTS OF THE LU SERVICES MANAGER

The LU services manager provides two categories of service:

End-user services
Session network services

End-user services are those services used when exchanging data between session partners. These services are discussed in this book. Session network services are those services that the LU uses when coordinating its network activity. These services are summarized in the next few paragraphs and are defined in detail in SNA Format and Protocol Reference Manual: Architecture Logic.

SESSION NETWORK SERVICES

The LU services manager interacts with the SSCP services manager (SSCP.SVC_MGR) to provide session, configuration, network operator, and maintenance and management services.

Session services are distributed between SSCPs and their LUs. Session services enable the SSCP and the LU to activate and deactivate LU-LU sessions.

Configuration services control the network's physical configuration. These functions include activating and deactivating links between nodes.

Network operator services handle communication between network operators and SSCPs, such as activating a physical or logical unit.

Maintenance and management services allow an LU and its SSCP to conduct various tests to detect failures within the network.

The interaction between an LU and its SSCP is based on the division of the network into domains. Each domain consists of an SSCP and the resources (physical units, logical units, and links) that the SSCP controls by having the capability to activate them (via the activation requests ACTPU, ACTLU, and ACTLINK).

(For more information on these services, see SNA Format and Protocol Reference Manual: Architecture Logic.)

END USER SERVICES

The LU services manager provides a protocol boundary to the end user. It coordinates end user interactions so that messages from different sessions are not arbitrarily interleaved across a destination.

The two components that provide end-user services are presentation services (LU.SVC_MGR.PS) and synchronization services (LU.SVC_MGR.SYNC). Synchronization services are use to restart print jobs in LU_T1 and to synchronize distributed data bases in LU_T6.

LU Presentation Services (LU.SVC_MGR.PS)

LU.SVC_MGR.PS correlates a half-session's request for a destination with the resources available. Once a resource is selected, the LU services manager keeps information on the resource for its own use and the use of the half-session. This function is done by UPM.ALLOCATE. (See Note below.)

Note: Some of the protocols defined in this book interact directly with undefined components; these undefined components, or undefined protocol machines (UPMs), generally are not named, shown in block diagrams, or discussed in the text. They represent implementation options that are not architecturally prescribed or options that are not, as yet, formally specified.

LU Synchronization Services (LU.SVC MGR.SYNC)

In LU-LU session type 1, an LU can restart printer jobs at its session partner by being aware of where the printer head is on the page when an error occurs. The LU with the printer takes checkpoints (records its current status) at an interval specified by the originating LU. If an error occurs, the printer LU notifies the originating LU, and the recovery sequence is started.

In LU-LU session type 6, an LU can present to its end user(s) an environment in which various system resources, such as terminals, data bases, and queuing facilities, are allocated and deallocated within the period that a session is active. Typically, allocation and deallocation are synchronous with the attaching of a transaction processing application program to the LU and its subsequent termination, or detaching from the LU.

Frequently, transaction processing involves multiple-step actions, wherein all steps must be completed as a unit because only the combined action has meaning. For example, correlated records in a distributed data base must all be updated in synchronism with each other. Sync point protocols enable the two ends of a session to complete each unit of work, either committing to the update if no errors occurred, or undoing the unit of work if the multiple-step process fails.

SESSION PRESENTATION SERVICES (HSID_N.SPS)

Each half-session provides protocols for presenting information to the end user in a usable form. These protocols include handling of data streams, encoding and compressing data, and error recovery procedures. The end user could be an application program or implementation-specific coding that interacts with a device or medium (for example, a printer, display, or diskette). Since devices typically need an operator, the operator can be considered as part of the end user.

The structure of SPS is shown in Figure 1-6. The SPS components vary from half-session to half-session depending on the LU-LU session type and the PS profile specified in BIND SESSION. In addition, the SPS components are influenced by the origin or destination that ultimately processes the data. For the full names of SPS components, see "Receive Side" and "Send Side" below.

RU Routing

An RU router routes the MU or DMU to the proper side of SPS, either send or receive, depending on the direction of the flow through the session. The RU router is not shown in Figure 1-6 to simplify the figure.

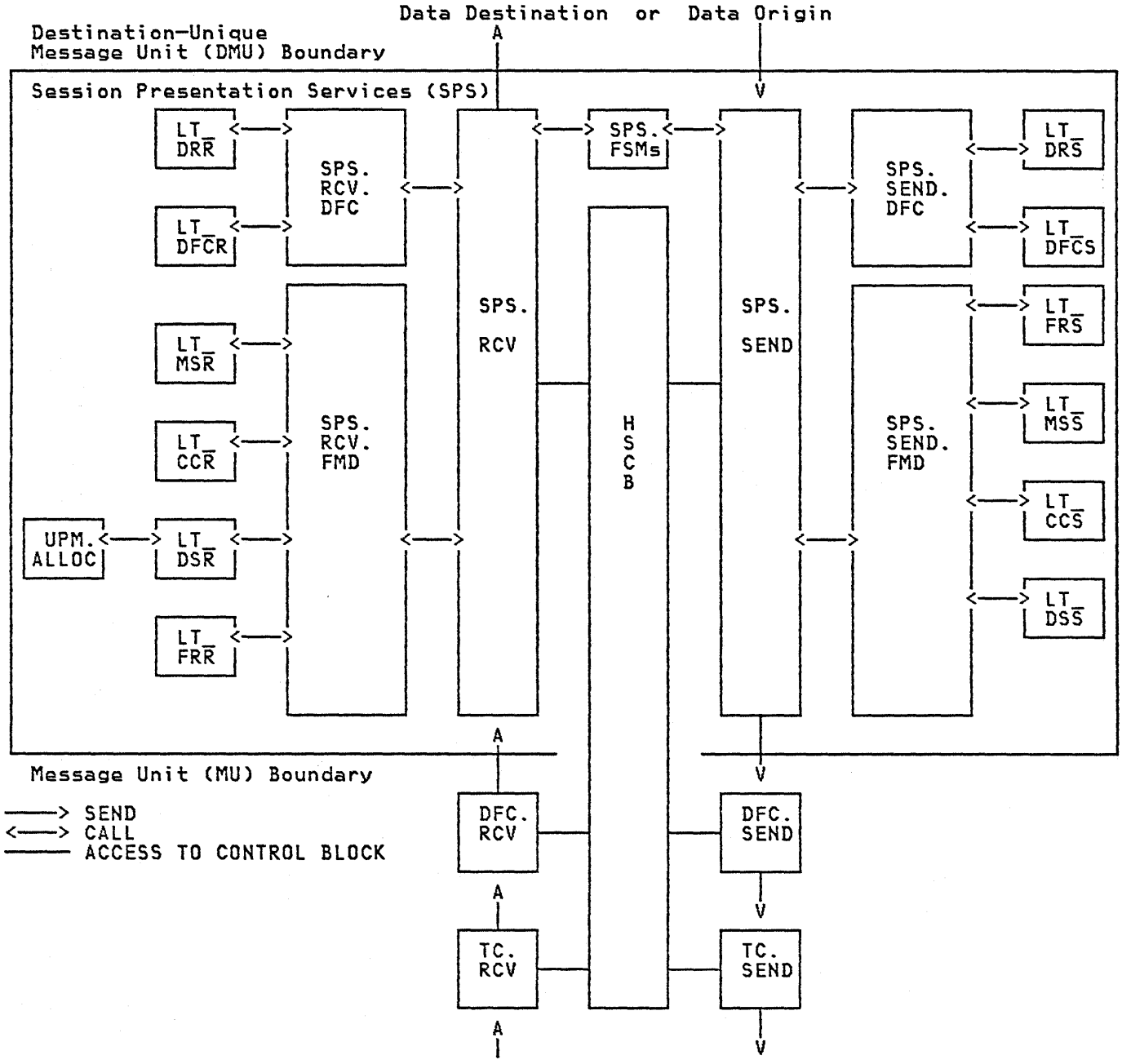


Figure 1-6. Session presentation services (SPS) components. SPS processes message units (MUs) received from DFC and creates destination-unique message units (DMU) which are passed to the selected destination. SPS also receives DMUs from the data originator, creates MUs and sends them to DFC.

Note: FMD is used in the RU category field of the RH as an indication that the RU is associated with the data origin or data destination functions of SNA. When the RU category field (RU_CTGY) is FMD, the RU router uses SPS.RCV.FMD and SPS.SEND.FMD. When RU_CTGY is DFC, the RU router uses SPS.RCV.DFC and SPS.SEND.DFC.

Receive Side

Session presentation services receives requests and responses with an RU_CTGY of FMD or DFC across the data flow control boundary from DFC Receive (DFC.RCV) as message units (MUs), which consist of a request/response header (RH) and request/response unit (RU). It sends destination-unique message units (DMUs) across the session protocol boundary to the data origin or data destination.

Session presentation services receive (SPS.RCV) enforces the appropriate SPS protocols for the MU received. It routes the MU to SPS.RCV.FMD or SPS.RCV.DFC depending on RU_CTGY. It is responsible for transforming MUs to DMUs across the session protocol boundary to the data origin or data destination.

SPS.RCV.DFC validates DFC requests and responses by calling DFC request receive (LT_DFCR) and DFC response receive (LT_DRR), respectively. (Components named LT_xxxx are different for each LU-LU session type.) If LT_DFCR indicates a request contains an error, or if LT_DRR indicates a response contains an error, then SPS.RCV.DFC changes the request or response to an exception request (EXR).

SPS.RCV.FMD validates FMD requests and responses by calling the following procedures, as appropriate:

- Destination services receive (LT_DSR) validates the formats and protocols of function management headers (FMHs) received on the half-session. It detects certain FMH protocol and format errors and notifies SPS.RCV of those errors. The action LT_DSR takes is based on the FMH it receives:

  FMH-1 - Manages a destination stack, which indicates which destination is the active destination and which destinations, if any, have been suspended.

  FMH-2 - Validates the data management activity requested by the FMH-2 against the active destination.

  FMH-3 - Validates the data management activity requested by the FMH-3 against all destinations of the half-session.

  FMH-4 - Validates control information against information received in previous FMHs.

  FMH-5 - Validates the format of the FMH-5, then uses UPM.ALLOC to activate (attach) a process that handles subsequent FMHs and their associated data.

  FMH-6 - Validates the command in the FMH-6 against the process attached by the FMH-5.

  FMH-7 - Validates that the FMH-7 follows the negative response X'0846'.

  FMH-10 - No action.

- Message unit services receive (LT_MSR) processes requests and responses as required by the half-session. LT_MSR deblocks the data stream and validates its format. If the data stream format is faulty, LT_MSR notifies SPS.RCV.FMD of the error. LT_MSR creates destination unique message units (DMUs) to be passed to the selected destination.

- Compression/Compaction Receive (LT_CCR) decompresses or decompacts the data stream as required by the selected destination using information in the string control byte (SCB). It detects invalid string control byte (SCB) code points or count fields and notifies SPS.RCV.FMD of the error.

- FMD Response Receive (LT_FRR) processes responses to FMD data units previously sent to the other half-session.

SPS.RCV passes requests or responses in the form of DMUs to the data origin or data destination. All errors detected by LT_DSR, LT_MSR, LT_CCR, LT_FRR or SPS.RCV are changed to exception requests (EXR) by SPS.RCV.

Send Side

Session presentation services receives requests and responses with an RU_CTGY of FMD or DFC across the session protocol boundary as DMUs, and sends them across the data flow control boundary as MUs.

Session presentation services send (SPS.SEND) enforces the appropriate SPS protocol for the DMU received. It routes the DMU to the other SPS components as required.

Since DFC.SEND is responsible for sending positive and negative responses to the session partner, SPS.SEND provides DFC.SEND with:

- No notification if the chain is a no-response chain
- A negative notification to the RU in error when the chain is a definite- or exception-response chain
- A positive notification to the last RU in the chain not found to have an error when the chain is a definite- or exception-response chain

SPS.SEND.DFC validates DFC requests and responses (those identified by RU_CTGY of DFC) by calling DFC request send (LT_DFCS) and DFC response receive (LT_DRS), respectively. (Components named LT_xxxx are different for each LU-LU session type.)

SPS.SEND.FMD validates FMD requests and responses by calling the following procedures, as appropriate:

- Message unit services send (LT_MSS) validates the syntax and semantics of the DMU.
- Destination services send (LT_DSS) validates the formats and protocols of Function Management Headers (FMH) to be sent on the half-session. LT_DSS checks for FMH protocol and format violations in the data stream being sent to the session partner by SPS.SEND.FMD. LT_DSS also maintains private FSMs, such as the destination stack for FMH-1.
- Compression/compaction send (LT_CCS) compresses or compacts the data stream as required for efficient transmitting of the data. LT_CCS builds SCBs, inserts them into the data stream, and returns the compressed or compacted data stream to SPS.SEND.FMD.
- FMD Response Send (LT_FRS) processes responses to be sent to the other half-session in reply to FMD data units received by this half-session.

If any SPS.SEND component detects an error, SPS.SEND.DFC or SPS.SEND.FMD changes the request or response to an exception request (EXR), and returns the EXR to the data origin or data destination.

SPS finite state machines (SPS.FSM) provides the common FSMs and tables required by SPS. These FSMs and tables are available for query and update by SPS components.

Component Matrix

Each LU-LU session type has some variation of the components depicted in Figure 1-6. Figure 1-7 shows that relationship.

Component		LU-LU Session Type					
		0	1	2	3	4	6
Session presentation services send	SPS.SEND	R	R	R	R	R	R
Session presentation services send	SPS.SEND_FMD	R	R	R	R	R	R
Message unit services send	LT_MSS	R	R	R	R	R	R
Compression/compaction send	LT_CCS	O	O	-	-	O	-
FMD response send	LT_FRS	R	R	R	R	R	R
Destination services send	LT_DSS	O	O	-	-	O	R
Session presentation services send	SPS.SEND_DFC	R	R	R	R	R	R
DFC response send	LT_DRS	R	R	R	R	R	R
DFC request send	LT_DFCS	R	R	R	R	R	R
Session presentation services receive	SPS.RCV	R	R	R	R	R	R
Session presentation services receive	SPS.RCV.FMD	R	R	R	R	R	R
FMD response receive	LT_FRR	R	R	R	R	R	R
Compression/compaction receive	LT_CCR	O	O	-	-	O	-
Destination services receive	LT_DSR	O	O	-	-	O	R
Message unit services receive	LT_MSR	R	R	R	R	R	R
Session presentation services receive	SPS.RCV.DFC	R	R	R	R	R	R
DFC response receive	LT_DRR	R	R	R	R	R	R
DFC request receive	LT_DFCR	R	R	R	R	R	R
Undefined protocol machine allocate	UPM_ALLOC	O	O	O	O	O	R

R = Required
O = Optional
- = Not applicable

Figure 1-7. Relationship between session presentation services components and LU-LU session types.

Type 1 LU-LU sessions operate in a network controlled by one or more SSCPs. One half of the session (called a half-session or session partner) is the primary LU (PLU), and the other is the secondary LU (SLU).

LU-LU session type 1 (LU_T1) is used to manage multiple input and output devices associated with a logical unit, such as printers, card readers, card punches, and storage devices. An operator console that can be used interactively or for diagnostic messages is often present.

Some LU_T1 implementations perform such data management functions as storing data for later retrieval and distribution to multiple destinations. Application programs and subsystems may also use LU-LU session type 1 to communicate within their network.

LU_T1 half-sessions operate in an asymmetric environment; that is, the primary LU controls the session, and the secondary LU conforms to the primary LU's requirements. The primary LU also assumes error recovery responsibility for the session.

The session begins when the primary LU sends a BIND SESSION request to the desired session partner. The BIND SESSION request identifies those functions the primary LU wishes to use. If the secondary LU's send and receive capability is comparable, the secondary LU returns a positive response; otherwise, it must return a negative response.

LU_T1 implementations need not contain all functions of the LU_T1 protocol. By using BIND SESSION, implementations can determine whether there is a subset of functions that allows meaningful communication between session partners.

HALF-SESSION CHARACTERISTICS

FM, TS, AND PS PROFILES

LU_T1 implementations use FM profile 3 or 4. Profile 4 allows the LUSTAT (LU Status) request to be sent in both directions; whereas profile 3 restricts it to secondary-to-primary only. Profile 4 also has a more comprehensive quiesce-processing protocol than profile 3.

LU_T1 implementations use TS profile 3 or 4. Profile 4 allows the REQUEST RECOVERY (RQR) and SET AND TEST SEQUENCE NUMBER (STSN) requests and profile 3 does not.

The SNA requests that are allowed with each profile are found in SNA Reference Summary.

For TS profile 3, the current operation is aborted if a session failure occurs. For TS profile 4, a checkpoint record is created when the error is detected for later STSN-based restart.

LU_T1 implementations use a PS profile that allows:

- Function management headers. LU_T1 implementations can run with or without FM headers. All FMH-1, FMH-2, and FMH-3 functions discussed in "Part 2 Chapter 4: Function Management Headers" are allowed with the exception of the DSP select field in FMH-1. Data stream profile (DSPs) values can be X'0' (default) or X'B' (structured fields).

  The BIND SESSION request contains a presentation services usage field which enables its sender to specify an FMH set. In general, the sets are:

Set 0 FMH-1 is supported, but only for the default destination. Medium selection is not allowed. Set 0 is used by sessions that use structured fields in the data stream but otherwise do not use FMH-1. The FMH-1 and FM data must be in the same chain, and the FMH-1 must both begin and end the destination (see the BEGIN/END FMH-1 information under "Using FMH-1 for Destination Selection" in Part 2 Chapter 4). No FMH-2s or FMH-3s are allowed.

Set 1 FMH-1 is supported for any destination available to the session. The FMH-1 and FM data must be in separate chains. Medium selection is allowed; data set selection using DSNAME is not. COMPACTION TABLE and PDIR FMH-2s are allowed (see BIND SESSION bytes 16 and 21). COMPACTION TABLE and QUERY FOR COMPACTION TABLE FMH-3s are allowed (see BIND SESSION bytes 16 and 21).

Set 2 FMH-1 is supported for any destination available to the session. The FMH-1 and FM data may be in the same chain. Medium selection is allowed; data set selection using DSNAME is not. COMPACTION TABLE and PDIR FMH-2s are allowed if specified in BIND SESSION (bytes 16 and 21). COMPACTION TABLE and QUERY FOR COMPACTION TABLE FMH-3s are allowed if specified in BIND SESSION (bytes 16 and 21).

Set 3 All FMH-1, FMH-2, and FMH-3 functions are supported except as noted above for DSP Select field of FMH-1.

- Data streams. All LU_T1 data streams are based on the SNA character string (SCS). The components of the data stream are described in Part 2 Chapters 1, 3, 4, and 5. See "Data Stream Profiles" on page 17 and "Set Structure" on page 18 for information on allowed functions.
- Attended and unattended terminal operations. An operator does not have to be present at the terminal if the LU is able to hold messages to the operator for delivery when the operator is present.

TC (TRANSMISSION CONTROL) CONSIDERATIONS

Transmission Control services, such as pacing and cryptography, are summarized in SNA Concepts and Products and defined in SNA Format and Protocol Reference Manual: Architecture Logic.

DFC (DATA FLOW CONTROL) CONSIDERATIONS

Data flow control services, such as chaining, brackets, and send/receive modes, are summarized in SNA Concepts and Products and defined in SNA Format and Protocol Reference Manual: Architecture Logic. The following paragraphs give additional considerations.

SEND-RECEIVE MODES

The following exceptions apply:

- Half-duplex contention. The secondary half-session is always the contention winner (specified in BIND SESSION). If the secondary half-session encounters a contention condition, it sends a negative response (X'081B' - receiver in transmit mode). It then sends its S->P transmission with EC on, or LUSTAT (X'00010000' - component available) if it cannot start the S->P transmission because of some intervening activity at the SLU. (It is possible that this intervening activity may cause the P->S retry to fail also; the operator, for example, may seize the secondary's buffer resource after the LUSTAT is sent.)
- Half-duplex flip-flop. No exceptions.
- Full duplex. Not allowed.

BRACKET PROTOCOL

The following exceptions apply:

- The secondary half-session must be the bracket first speaker as defined in BIND SESSION.
- The primary half-session may send the BID request before transmitting to avoid a bracket contention condition (both session partners attempting to begin a bracket at the same time).
- If the primary half-session begins transmitting without first sending BID, the RU that begins the bracket (the one that carries BB) must be sent RQD (request definite response).
- Destination selection cannot span brackets. Multiple destinations may be selected within a bracket, but only one destination can be active at a time.

HALF-DUPLEX CONTENTION PROTOCOL

When an LU_T1 session uses half-duplex contention protocol, the primary and secondary half-sessions are in contention-to-send state when they are between chained transmissions. Either primary half-session or secondary half-session can initiate a transmission. Contention is the result of simultaneous transmissions from the primary and secondary half-sessions. Byte 7, bit 3 of the BIND SESSION request defines that the secondary half-session wins contention. The contention loser (the primary) receives a negative response with sense data indicating that the secondary is in send state. When this response is returned, the contention loser must enter receive state. The send/receive relationship is in effect for the duration of one chained transmission. When the current chain is completed, the primary and secondary half-sessions are again in contention for send state.

HALF-DUPLEX FLIP-FLOP CHANGE-DIRECTION PROTOCOL

Use of CD Indicator (CDI): The change direction indicator (CDI) in the request header (RH) is used to change the direction of data flow when session partners are using half-duplex flip-flop mode and are within brackets (after BB but before EB). To change the direction of flow, the half-session that has the authority to send at a given moment turns on the CDI in the last RU it sends to its session partner.

The sending of the CDI should occur under the following circumstances:

- An LU has reached an end-of-send data-transition condition (including any keyboarded data on the LU flow).
- An LU has received SIG X'0001' (request change direction). The procedures that must be followed when returning the CDI are discussed below under "Signal Codes."
- A PLU has been session-activated in the send state, but has no data traffic to send.
- At session activation, the PLU receives a CINIT that indicates that the SLU must send data first.
- When the LU receives CD and the LU does not have the capability to send data (for example, a printer). In this case, the CDI is returned as part of LUSTAT X'0002'.

An LU should send available FM data, if any, when the CDI is received. If there is no FM data pending, the LU must wait for additional FM data to send or for one of the conditions described above.

Use of SIG (Signal): An LU that is capable of sending FM data must be capable of sending SIG X'0001' (request change direction), under operator control or automatically, when in receive state with FM data to send.

The sender of SIG X'0001' must return the CDI when it has no more data to send.

The receiver of SIG X'0001' must not send SIG X'0001' after sending the CDI until the end-chain indicator (ECI) has been returned. This prevents a signal ping/pong condition from occurring with no data flowing.

Data Direction Resolution: At session activation, the session partners are between brackets, and neither is in send state. The PLU normally sends a BID request if it wants to enter send state, but it may enter send state by sending data to the SLU if that data is sent requiring a definite response (BB,RQD). The SLU enters send state by sending data to the PLU. The secondary half-session is the contention winner in the event that both PLU and SLU attempt to begin a bracket concurrently.

PS (PRESENTATION SERVICES) CONSIDERATIONS

FM HEADER PROCESSING

Primary LUs must be able to access data in data sets located at a secondary LU (SLU). There are a number of data set organizations for user data that may appear at the secondary LU. Not all of these organizations are supported by all SLU implementations. The FMH sets and FMH flags fields of BIND SESSION indicate which organizations are supported by specifying which operations are allowed on the data sets. The minimum functions for data management, which are specified by FMH set 3, consists of all the functions allowed in FMH set 2 plus an addressed direct data set located at the secondary and accessible by the primary. With FMH set 3, the PLU is allowed to issue FMH-1 and the REPLACE FMH-2 (using the RECORD ID and PASSWORD FMH-2s) against the SLU data sets. The VOLUME ID FMH-2 is supported for the BEGIN and BEGIN/END FMH-1s.

In addition to this basic level of data management, the FMH flags fields may specify that FM headers support other data set organizations and other operations on the addressed direct organization. The additional organizations supported are sequential (using ADD) and keyed direct (using ADD and ERASE FMH-2s) organizations. The additional functions for addressed direct organizations are sequential access, using ADD, NOTE and NOTE REPLY FMH-2s.

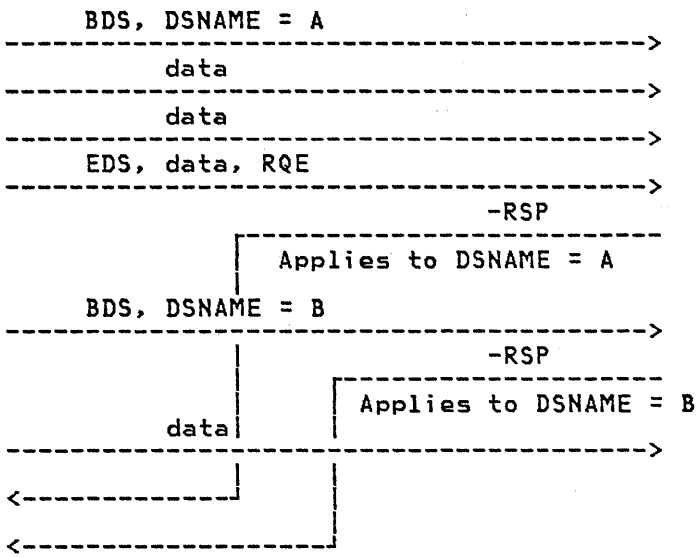
The SERIES ID and STATUS FMH-3s are optionally allowed, as are the ADD REPLICATE and REPLACE REPLICATE FMH-2s.

The QUERY FOR DATA SET FMH-2 is optionally allowed, but when the BEGIN FMH-1 is issued that identifies the requested data set, the BEGIN FMH-1 must be the only entry in the issuer's send stack.

The EXECUTE PROGRAM OFFLINE FMH-2 is optionally allowed.

The file maintenance FMH-2s (CREATE DATA SET, SCRATCH DATA SET, and SCRATCH ALL DATA SETS) are optionally allowed.

If an error occurs in a chain carrying an FM header, the data following the header can be applied to a different data set than was intended. Consider the following sequence:



The rules that avoid this situation are:

1. If BIND SESSION allows an LU to send RQD and specifies FMH set 3, then any FMH-1 or FMH-2 that the LU sends that modifies a resident data set (as opposed, for example, to transmitting data for output processing on printer, console, or card) must be sent RQD and the LU must wait until the response is returned.

2. A BIND SESSION may be rejected by an SLU if the BIND SESSION specifies both (1) FMH set 3, and (2) PLU will only send RQE.

DATA STREAM PROFILES

LU_T1 implementations do not use the DSP select field of the FMH-1 to select data stream profiles. They either specify X'0' so that the profile (data stream) is assumed when the medium is selected, or specify X'B' to indicate that the data stream contains structured fields. (See "Function Management Header Type 1 (FMH-1)" in Part 2 Chapter 4.)

DATA STREAMS CONTAINING STRUCTURED FIELDS

Structured fields in LU_T1 are introduced by FMH-1s with the DSP select field set to X'B' (structured fields). Reply type structured fields are introduced by FMH-1s with the DSP select field set to X'B' and the stack reference indicator set to B'1'.

A BEDS FMH-1 (begin/end destination selection) may be used to carry structured fields to SLUs that do not support FMHs for destination selection. (See BIND bytes 6 and 15.)

Structured fields can be implemented in sets based on the structure in Figure 2-1.

In order to implement a function in any given set in this structure, a product must implement all of the operations defined by that set and all of the operations in previous sets. A product may implement one or more of the sets.

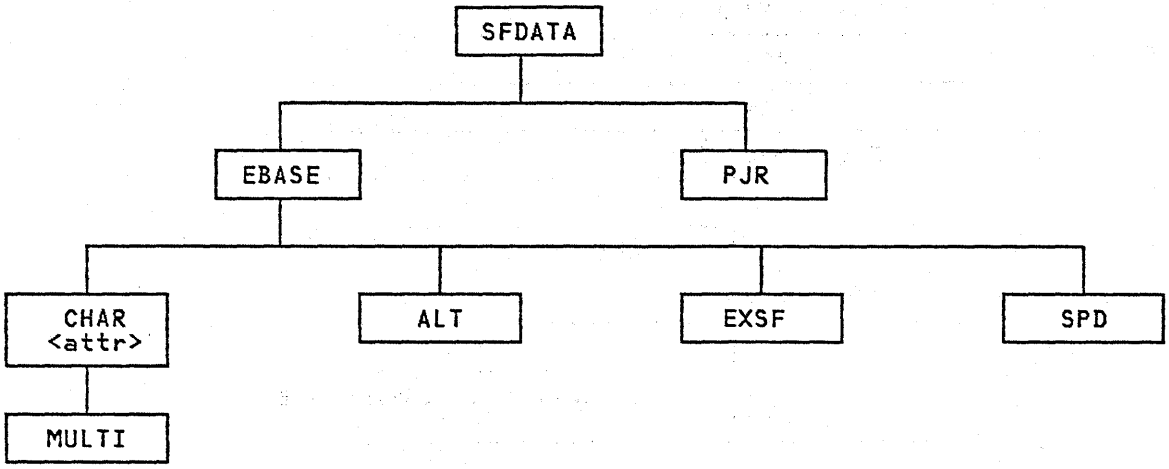


Figure 2-1. Structured field sets. Function sets are used when implementing structured fields. If the MULTI set is chosen, the values of <attr> must include character sets. Character sets are selected using the Set Attribute <attr> control described in Part 2 Chapter 1.

Set Structure

The operations required in each set are summarized in the following table and defined in full in the following text.

SET	QUERY REPLIES	FUNCTION
SFDATA		BEDS FMH-1 (DSP = X'B' structured fields)
PJR		Request Recovery Data structured field Recovery Data structured field Set Checkpoint Interval structured field Restart structured field
EBASE	Usable Area	Read Partition (Query) structured field
CHAR	<attr>	SA <attr> SCS control (X'28')
ALT	Character Sets (ALT flag = B'1')	Graphic Escape SCS control (X'08')
MULTI	Character Sets (MULTI flag on)	Load Programmed Symbols structured field
EXSF	Usable Area (EXSF flag = B'0')	SCS Data structured field
SPD	Device Character- istics	Set Print Density SCS control

STRUCTURED FIELD SUPPORT (SFDATA): This set supports the structured field data stream. If this set is implemented, the half-session must support:

- BEDS FMH-1 (DSP = Structured Fields)

Other destination selection options, such as BEGIN and CONTINUE, are optional for FMH-1s with DSP = Structured Fields.

PRINT JOB RESTART (PJR): If this set is implemented, the half-session must support:

- Request Recovery Data structured field
- Recovery Data structured field
- Set Checkpoint Interval structured field
- Restart structured field

EXTENDED PRESENTATION BASE (EBASE): This is the base set of functions for a half-session which supports any of the data presentation functions supported by structured fields. If this set is implemented, the half-session must support:

- Read Partition structured field
  (Query operation only)
- Query Reply (Usable Area) structured field
- SCS Data structured field

CHARACTER ATTRIBUTES (CHAR): If this set is implemented, the half-session must support:

- Set Attribute <attr> SCS control

ALTERNATE CHARACTER SET (ALT): If this set is implemented, the half-session must support:

- Query Reply (Character Sets) structured field
  (with ALT flag set on)
- Graphic Escape SCS control

MULTIPLE LANGUAGE CAPABILITY (MULTI): If this set is implemented, the half-session must support:

- Query Reply (Character Sets) structured field
  (with MULTI flag set on)
- Load Programmed Symbols structured field

EXTENDED STRUCTURED FIELD SUPPORT (EXSF): If this set is implemented, the half-session must support:

- Query Reply (Usable Area) structured field
  (with EXSF flag set to B'0')
- SCS Data structured field

SET PRINT DENSITY (SPD): If this set is implemented, the half-session must support:

- Query Reply (Device Characteristics) structured field
  (with SPD descriptor)
- Set Print Density SCS control

STRUCTURED FIELDS USED BY LU_T1

The following structured fields are common between the structured field data stream used by LU_T1 and the SNA 3270 data stream used by LU_T2 and LU_T3:

- Load Programmed Symbols
- Read Partition (Query operation)
- Query Reply
- SCS Data

The following structured fields are unique to LU_T1:

- Set Checkpoint Interval
- Request Recovery Data
- Recovery Data
- Restart

LOAD PROGRAMMED SYMBOLS: This structured field lets an application program load character definition data into a specified PS (programmed symbol) storage in the device. For details see Part 2 Chapter 3.

READ PARTITION (QUERY): This structured field lets an application program enquire about the definition of the device -- what features are present and how they are defined. For details see Part 2 Chapter 3.

QUERY REPLY: This structured field lets the LU to respond to a Read Partition (Query). Query replies for character sets, color, and highlighting are supported for LU_T1. For details see Part 2 Chapter 3.

SCS DATA: The SCS Data structured field allows sending of SCS (SNA character string) data intermixed with other structured fields. The SCS data portion of this structured field is treated the same as SCS data not delimited by an SCS structured field. For details on the structured field, see Part 2 Chapter 3. For details on SCS controls, see Part 2 Chapter 1.

SET CHECKPOINT INTERVAL: Four structured fields are used for restarting printer jobs:

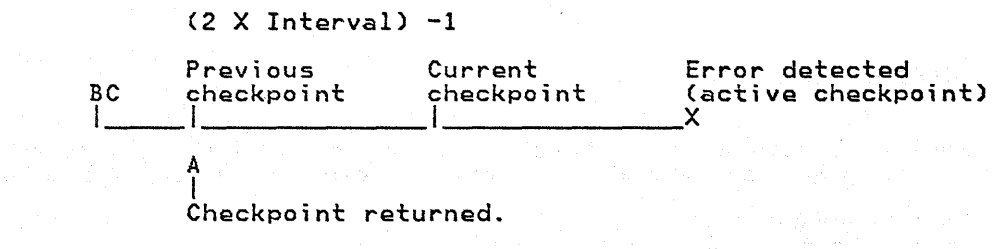
- Set Checkpoint Interval
- Request Recovery Data
- Recovery Data
- Restart

The primary LU (PLU) uses Set Checkpoint Interval to send to the secondary LU (SLU) the number of pages that are to be in the interval between SLU checkpoints (the SLU is the LU for the printer). When the SLU is capable of restarting printer jobs, it takes a checkpoint (that is, records its status) when it receives Set Checkpoint Interval and each time it completes the interval set by the structured field. The structured field should not be sent if the SLU does not restart printer jobs. (See BIND SESSION byte 17 bit 4.)

REQUEST RECOVERY DATA: The PLU uses Request Recovery Data to request that the SLU send the recovery data for restarting the print job. The structured field cannot be followed by other data, and the last RU of the RU chain must have the change direction indicator set on.

RECOVERY DATA: The SLU sends Recovery Data to the PLU in response to Request Recovery Data. The Recovery Data structured field contains all the checkpoint information needed to restart the printer job, such as the number of pages printed since the last (current) checkpoint, the number of lines printed on the page where the error occurred, and horizontal and vertical formatting controls.

The checkpoint returned by the SLU is always the one previous to the last, except that the SLU never goes back beyond the beginning of the chain. That is, it can be as far from the error as:



Note: When checkpoint is used in LU_T1, it refers to the checkpoint returned as illustrated above.

The Recovery Data structured field contains zeros when there is no recovery data or when the SLU is unable to send the proper data.

The SLU may also send Recovery Data when restarting on an alternate printer, but only after receiving Set Checkpoint Interval and Request Recovery Data.

RESTART: The PLU sends Restart to the SLU after the PLU has been told that the error condition has been corrected. The Restart structured field contains the number of pages and lines to be bypassed before printing starts again using the data that follows the structured field. The PLU ensures that the first byte of data, which is in the next RU chain, resumes at the checkpointed spot, or at the start of the SCB (string control byte) string or structured field containing the checkpointed spot.

SIGNAL CODES

SIG is an expedited request that may be sent between half-sessions regardless of the normal flows. It carries a 4-byte signal code set by the sending LU.

The valid signal codes for LU_T1 are:

X'0001' + 'rrrr'	Request to send (0001) and reserved field (rrrr).
X'0003' + 'uuuu'	Intervention required, no data lost (0003) and user defined field.

For LU_T1 the user defined field is:

Byte	Bits	Name	Content
3	0-3	Select medium	See FMH-1 Byte 2 Bits 0-3 in Part 2 Chapter 4.
	4-7	Logical subaddress	See FMH-1 Byte 2 Bits 4-7 in Part 2 Chapter 4.
4	0-7	Intervention required	X'00' Undefined intervention X'01' Out of forms

If a signal code is received that is not recognized or can not be acted on, it may be rejected with a negative response (sense code X'1003' function not supported).

LU STATUS (LUSTAT) CODES

Secondary LUs may send the following status codes in the LUSTAT request.

0001	Component now available
0002	No FM data to transmit
0003	Entering attended mode of operation
0004	Entering unattended mode of operation
081C	Component failure
082B	Component available but presentation space integrity lost
0831	Component disconnected (power off or other disconnecting condition)

The format of LUSTAT is:

Byte	Value	Definition
0	X'04'	Request code
1-4		Status value and status extension field (two bytes each)
	X'0001'+'cc00'	Component now available (0001) and component identification (cc00)
	X'0002'+'rrrr'	No FMD requests to send and reserved field.
	X'0003'+'cc00'	Entering attended mode of operation (0003) and component identification (cc00)
	X'0004'+'cc00'	Entering unattended mode of operation (0004) and component identification (cc00)
	X'081C'+'cc00'	Component failure - permanent error (081C) and component identification (cc00)
	X'082B'+'cc00'	Component available, but presentation space integrity lost (082B), and component identification (cc00)
	X'0831'+'cc00'	Component disconnected (0831) and component identification (cc00)

Values for 'cc' byte are:

X'00' = LU itself rather than a specific LU component

Otherwise,
Bit 0, set to 1 = media other than LU
Bits 1-3, LU component class:

'000' Console
'001' Exchange
'010' Card
'011' Document
'100' Nonexchange disk
'101' Extended document
'110' Extended card

Bits 4-7, LU component device address

LU_T1 does not allow BB (begin bracket) on LUSTAT.

ERRORS

All errors are turned into exception requests (EXRs) by the element of the network that detects the error. The exception request is sent forward (except for loss of path type errors) to the LU named in the DAF (destination address field of the transmission header). The LU then processes the EXR and returns a negative response if the request was marked definite or exception response requested.

Errors can be categorized in LU_T1 by this algorithm:

```
IF negative response received by SLU THEN
   DO;
   .  IF path error THEN
   .     DO;
   .     .  Notify operator of loss of session;
   .     .  IF running TS profile 4   THEN
   .     .        create a checkpoint;
   .     .  ELSE;
   .     END;
   .  ELSE
   .     DO;
   .        DFC processing required to go to receive state;
   .        await primary LU error recovery action;
   .     END;
   END;
ELSE;


IF negative response sent by SLU  THEN
   DO;
   .  IF RH error or state error or request error  THEN
   .     DO;
   .     .  Purge chains;
   .     .  Go to receive state;
   .     .  Await primary ERP;
   .     END;
   .  ELSE;
   .  IF contention error (X'081B'), BETB or INB under
   .        a contention BIND  THEN
   .     DO;
   .     .  IF state is/becomes INB and S->P transmission is
   .     .     sent  THEN
   .     .     Continue until EC is sent;
   .     .  ELSE;
   .     .  IF EC on the normal flow has not been sent
   .     .        after the negative response (X'081B')  THEN
   .     .     Send LUSTAT (X'00010000');
   .     .  ELSE;
   .     END;
   .  ELSE;
   .  IF brackets reject (0813) or 0814)  THEN
   .     continue;
   .  ELSE;
   .  IF intervention required (0802)  THEN
   .     DO;
   .     .  Supply the intervention locally;
   .     .  When it is complete THEN
   .     .     Send LUSTAT (X'0001cc00'); /* to report that
   .     .                                ** intervention has
   .     .                                ** been supplied */
   .     .  IF another device has been selected
   .     .        meanwhile  THEN
   .     .     The LUSTAT must contain the component ID
   .     .           of the device now available;
   .     .  ELSE;
   .     END;
   .  ELSE;
   .  IF request reject (08xx) (including 081B) while HDX-FF
   .        INB Send State THEN
   .     DO;
   .     .  Purge the chain;
   .     .  Go to receive state;
   .     .  Await primary ERP;
   .     END;
   .  ELSE;
   END;
ELSE;
```

The following algorithm is used at the SLU in order to determine whether to send intervention required (0802) or request not executable (081C) when a device error is detected during an output operation (for example, an out of paper condition).

```
IF data not lost in attended mode  THEN
   DO;
   .  Hold pacing;
   .  Start a timer of between 0 and 10 minutes duration;
   .  IF timer expires  THEN
   .     Recoverable data loss = 1; /* this protects the
   .                                ** network and PLU from
   .                                ** resource depletion */
   .  ELSE
   .     Resume pacing when the intervention has been supplied;
   END;
ELSE;
IF recoverable data loss = 1 in attended mode  THEN
   DO;
   .  Send intervention required (0802);
   .  Go to contention state;
   .  When intervention has been supplied  THEN
   .     Send LUSTAT (0001cc00); /* cc is the component ID*/
   .                             /* X'00' may be used if no
   .                             ** other component has been
   .                             ** selected after the 0802
   .                             ** response. */
   END;
ELSE;
If unrecoverable data loss or unattended mode  THEN
   DO;
   .  Purge chain;
   .  Send request not executable (081C);
   .  Stay in receive state;
   .  Optionally send RSHUTD;
   END;
ELSE;
```

SENSE CODES AND SENSE DATA

Function management (FM) errors detected by the PLU are not to be sent to the SLU by way of sense codes. The PLU must correct these errors through application program action.

The following errors may be sent by the SLU: (For a complete description of each sense code, see SNA Format and Protocol Reference Manual: Architecture Logic.)

State Errors: 20xx

2002	Chaining error
2003	Bracket error

Request Errors: 10xx

1001	RU data error
1002	RU length error
1003	Function not supported
1005	Parameter error
1008	Invalid FM header

When 1008, Invalid FM header, is returned, bytes 2 and 3 of the sense may contain X'0000', or may optionally contain a detailed explanation of the FM header error, as follows:

1008 0801	Invalid function code parameters
1008 0803	Forms function cannot be performed
1008 0805	Unable to perform copy function

1008 0806	Compaction table outside supported set
1008 0807	Invalid PDIR identifier
1008 0808	Printer train function cannot be performed
1008 0809	FCB load function cannot be performed
1008 080A	FCB load function not supported
1008 080B	Invalid compaction table name
1008 080C	Invalid ACCESS
1008 080D	Invalid RECLEN
1008 080E	Invalid NUMRECS
1008 080F	Data set in use
1008 0810	Data set not found
1008 0811	Invalid password
1008 0812	Function not allowed for destination
1008 0813	Record too long
1008 0814	Data set full
1008 0815	Invalid RECID
1008 0817	Invalid VOLID format
1008 0818	Number of logical records per chain exceeded
1008 0819	Data set exists
1008 081A	No space available
1008 081B	Invalid VOLID
1008 081C	Invalid DSACCESS
1008 081D	Invalid RECTYPE
1008 081E	Insufficient resolution space
1008 081F	Invalid key technique
1008 0820	Invalid key displacement
1008 0821	Invalid key
1008 0822	Invalid N (number)
1008 0823	Invalid KEYIND
1008 0824	Invalid SERID
1008 0826	Invalid RECID format
1008 0827	Password not supplied
1008 0828	Record ID not supplied
1008 0829	Volume ID not supplied
1008 082A	Invalid PGMNAME
1008 2001	Invalid destination -- active
1008 2002	Invalid destination -- inactive
1008 2003	Invalid destination -- suspended
1008 2004	Invalid Suspend-Resume sequence
1008 2005	Interruption level violation
1008 2006	Invalid resume properties
1008 2007	Destination not available
1008 2008	Invalid end sequence
1008 2009	Invalid FM header length
1008 200A	Invalid field setting
1008 200B	Invalid destination
1008 200C	Invalid ERCL
1008 200D	Invalid DST
1008 200E	Invalid concatenation
1008 200F	FM data not allowed for header
1008 2010	BIND FM header set violation
1008 2014	FM header not sent concatenated
1008 2019	Invalid stack reference indicator (SRI)
1008 201A	Unable to accept CMI modification
1008 201B	Unable to accept CPI modification
1008 201C	Unable to accept ERCL modification
1008 4001	Invalid FMH type
1008 4002	Invalid FMH code
1008 4003	Compression not supported
1008 4004	Compaction not supported
1008 4005	Basic exchange not supported
1008 4006	Only basic exchange supported
1008 4007	Medium not supported
1008 4008	Code selection compression violation
1008 4009	FMHC not supported
1008 400A	Demand select not supported
1008 400B	DSNAME not supported
1008 400C	Invalid media subaddress field
1008 400D	Insufficient resources to perform function requested
1008 400E	DSP select not supported

Request Reject: 08xx

0802	Intervention required
0805	Session limit exceeded (in response to BIND)
080A	Permission rejected - SSCP not notified (contrast with 0845)
080B	Bracket race error
0811	Break
0812	Insufficient resource
0813	Bracket bid reject - No RTR forthcoming
0814	Bracket bid reject - RTR forthcoming
081B	Receiver in transmit mode
081C	Request not executable
0821	Invalid session parameter (in response to BIND)
0825	Component not available
0829	Change direction required
082B	Presentation space integrity lost
0831	LU component disconnected
0845	Permission rejected - SSCP will be notified (contrast with 080A)
084C	Permanent insufficient resource
0860	Function not supported - continue session
0862	Medium presentation space recovery
0863	Referenced local character set identifier (LCID) not found
0871	Read partition state error

BIND SESSION FORMAT FOR LU-LU SESSION TYPE 1

This section defines valid bit settings for the BIND request. The BIND sender selects the values to be used. Any value may be specified except those identified as not valid. All values listed must be supported by the BIND receiver unless the value is optional (identified by X in the Opt column) or not valid.

Byte	Bits	Value	Opt	Discussion
0	0-7	X'31'		Identifies this RU as a BIND request
1	0-3	X'0'		Specifies the format of the BIND RU. Only one format has been defined: format 0 (zero).
(1)	4-7			Type - denotes the type of BIND to be performed
		X'0'		Not valid
		X'1'		Nonnegotiable BIND

FM (Function Management) Profile

2	0-7	X'03'		FM profile 3
		X'04'	X	FM profile 4
				The FM profile defines the data flow control (DFC) protocols to be used for this session. (See SNA Reference Summary for highlights of the FM profile.)

TS (Transmission Subsystem) Profile

3	0-7	X'03'		TS profile 3
		X'04'	X	TS profile 4
				The TS profile defines the transmission control (TC) protocols to be used for this session. (See SNA Reference Summary for highlights of the FM profile.)

FM Usage

(4-7)				See bytes 4 through 7 below.

Primary Half-Session Protocols for FM Data

				Chaining Use:
4	0	B'0'		Primary half-session can send only single element chains.
		B'1'		Primary half-session can send single or multiple element chains.
				Request Mode Selection:
(4)	1	B'0'		Immediate request mode is used. Only one definite response can be outstanding at a time. That response must be received before the primary half-session can send another RU.
		B'1'	X	Delayed request mode is used. Multiple definite responses can be outstanding at one time. The sender of data in delayed request mode must be responsible for recovery.

Byte	Bits	Value	Opt	Discussion
				Chaining Responses:
(4)	2-3	B'00'		Not valid
		B'01'		Primary half-session can request only exception responses.
		B'10'		Primary half-session can request only definite responses.
		B'11'		Primary half-session can request definite or exception responses.
				Note: Secondary half-session is not required to accept "only exception response" in BIND when FM header set 3 is specified in byte 15.
(4)	4-5			Reserved
				Compression Indicator:
(4)	6	B'0'		Primary half-session cannot send compressed data.
		B'1'	X	Primary half-session can send compressed data.
				Note: B'1' is valid only if the code is EBCDIC (see byte 6 bit 4).
				Send End Bracket Indicator:
(4)	7	B'0'	X	Primary half-session will not send EB.
		B'1'		Primary half-session can send EB.

Secondary Half-Session Protocols for FM Data

				Chaining Use:
5	0	B'0'	X	Secondary half-session can only send single element chains.
		B'1'		Secondary half-session can send single or multiple element chains.
				Request Mode Selection:
(5)	1	B'0'		Immediate Request Mode is used. Secondary half-session can issue a request for a single definite response. No further transmission is sent until the secondary half-session receives the requested response.
		B'1'		Delayed request mode is used. Secondary half-session allows several definite responses to be outstanding at one time.
				Chaining Responses:
(5)	2-3	B'00'		Not valid
		B'01'		Secondary half-session can request only exception responses.
		B'10'		Secondary half-session can request only definite responses.
		B'11'		Secondary half-session can request either definite or exception responses.
(5)	4-5			Reserved
				Compression Indicator:
(5)	6	B'0'		Secondary half-session cannot send compressed data.
		B'1'	X	Secondary half-session can send compressed data.
				Note: B'1' is valid only if the code is EBCDIC (see byte 6 bit 4).
				Send End Bracket Indicator:
(5)	7	B'0'		Secondary half-session cannot send EB.
		B'1'		Secondary half-session can send EB.

Byte	Bits	Value	Opt	Discussion
Common Protocols for FM Data				
6	0			Reserved
(6)	1	B'0'		Primary half-session and secondary half-session cannot exchange FM headers.
		B'1'	X	Primary half-session and secondary half-session can exchange FM headers.
				Brackets Usage and Reset State:
(6)	2	B'0'		Not valid
		B'1'		Brackets are used and bracket state manager's reset state is BETB (between brackets).
				Bracket Termination Rule Selection:
(6)	3	B'0'	X	Bracket termination rule 2 is used. A bracket is terminated unconditionally when the last request of the chain that has EB in its first request is processed, regardless of the form of response requested.
		B'1'		Bracket termination rule 1 is used. Bracket termination is controlled by the form of response requested (definite or exception) for the chain containing (¬BB,EB).
				Alternate Code Selection:
(6)	4	B'0'		Alternate code set will not be used.
		B'1'	X	Alternate code set may be used.
				If byte 6, bit 4, is B'1', the code selection indicator (CSI field of the RH (request header)) controls whether the alternate code is to be used: CSI of 0 = EBCDIC CSI of 1 = ASCII
(6)	5-7			Reserved
				Normal-Flow Send and Receive Mode Selection:
7	0-1	B'00'		Not valid
		B'01'	X	Half-Duplex Contention
				Primary half-session and secondary half-session are in contention-to-send state when they are between chained transmissions.
		B'10'		Half-Duplex Flip-Flop
		B'11'		Reserved
				Recovery Responsibility:
(7)	2	B'0'		The contention loser is responsible for any error recovery attempts for the session.
		B'1'	X	Symmetric responsibility for recovery. The sender of data is responsible for recovery. The receiver of data purges to the synchronizing event, then awaits sender action.
				Contention Winner/Loser:
(7)	3	B'0'		Secondary half-session is contention winner (and therefore the brackets first speaker) and primary half-session is contention loser.
		B'1'		Not valid
(7)	4-6			Reserved

Byte	Bits	Value	Opt	Discussion
				Half-Duplex Flip-Flop (HDX-FF) Reset States:
(7)	7			Reserved
TS Usage				
(8-13)				See bytes 8 through 13 below.
				Staging Indicator and Pacing Count for Secondary CPMGR (connection point manager) to Primary CPMGR normal flow:
8	0	B'0' B'1'		Pacing in this direction occurs in one stage. Pacing in this direction occurs in two stages.
				Note: The meanings of 0 and 1 are reversed from the staging indicator for primary CPMGR to secondary CPMGR (see byte 12).
(8)	1			Reserved
(8)	2-7	B'nnnnnn'		Secondary CPMGR send pacing count. A value of 000000 means no pacing of requests flowing from the secondary.
9	0-1			Reserved
(9)	2-7	B'nnnnnn'		Secondary CPMGR receive pacing count. A value of 000000 causes the boundary function to substitute the value set by a system definition pacing parameter (if the system definition includes such a parameter) before it sends the BIND RU on to the secondary LU. A value of 000000 received at the secondary LU is interpreted to mean no pacing of requests flowing to the secondary LU.
10		X'ab'	X	Maximum RU size sent on the normal flow by the secondary half-session. This value represents the largest RU that can be sent. It is expressed as a mantissa and an exponent value of 2 by which the mantissa is multiplied.
		X'85'		if the mantissa is 8 and the exponent is 5 (X'85'), the RU size is 256 bytes (8×2^5).
		B'0n...'		When bit 0 is set to zero, no maximum is specified and the remaining bits 1-7 are ignored. When bit 0 is set to one, the byte is interpreted as X'ab'. See Appendix B for these values.
				Note: All LU_T1 implementations must support receiving an RU of 256 bytes and accept BINDs that say the smallest maximum RU send size is 256 bytes.
11		X'ab'	X	Maximum RU size sent on the normal flow by the primary half-session. This value represents the largest RU that can be sent by the primary half-session and is specified in the same format as the secondary half-session (byte 10).
		X'85'		if the mantissa is 8 and the exponent is 5 (X'85'), the RU size is 256 bytes (8×2^5).
		B'0n...'		When bit 0 is set to zero, no maximum is specified and the remaining bits 1-7 are ignored. When bit 0 is set to one, the byte is interpreted as X'ab'. Staging Indicator and Pacing Count for Primary CPMGR to Secondary CPMGR normal flow:
12	0	B'0' B'1'		Pacing in this direction occurs in two stages. Pacing in this direction occurs in one stage.
				Note: The meanings of 0 and 1 are reversed from the staging indicator for secondary CPMGR to primary CPMGR (see byte 8).
(12)	1			Reserved

Byte	Bits	Value	Opt	Discussion
(12)	2-7	B'nnnnnn'		Primary CPMGR send pacing count. A value of 000000 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 000000, it means no pacing of requests flowing from the primary. For single-stage pacing in the primary-to-secondary direction, this field is redundant with, and will indicate the same value as, the secondary CPMGR receive pacing count (see byte 9, bits 2-7 above).
13	0-1			Reserved
(13)	2-7	B'nnnnnn'		Primary CPMGR receive pacing count. A value of 000000 means no pacing of requests that flow to the primary. For single-stage pacing in the secondary-to-primary direction, this field is redundant with, and will indicate the same value as, the secondary CPMGR send pacing count (see byte 8, bits 2-7 above).

PS Profile

(14-25)				See bytes 14 through 25 below.
14	0	B'0' B'1'		PS usage field format is the basic format. Reserved
(14)	1-7	B'0000001'		LU-LU session type 1

PS Usage

15	0-3			FM Header Sets:
		X'0'	X	FM header set 0. BEDS FMH-1 allowed for default destination only.
		X'1'	X	FM header set 1. FMH-1s have no data set name field; restricted header and data combinations.
		X'2'	X	FM header set 2. FMH-1s have no data set name field; less restricted header and data combinations. Note: The difference between sets 1 and 2 is that 2 allows headers and data in the same chain. Set 1 allows the use of headers and data together only on the BEGIN/END (BEDS) FMH-1.
		X'3'	X	FM header set 3. Data management headers. The base functions of data management are specified by set 3. An Addressed Direct data set located at the secondary half-session is accessible by the primary half-session. With set 3, the primary half-session is allowed to issue FMH-1s and the REPLACE FMH-2 (using the RECORD ID and PASSWORD FMH-2s) against the secondary half-session data sets. The VOLUME ID FMH-2 is valid with FMH-1s that begin destinations.
(15)	4-7	X'0'		Data stream profile 0 (SCS parameters and card data streams must not span RUs).
		X'1'	X	Data stream profile 1 (SCS parameters and card data streams may span RUs).

Byte	Bits	Value	Opt	Discussion
Primary Half-Session Usage				
				(Secondary Half-Session Receive Capability)
(16-20)				Bytes 16-17 contain FM header flags. Bytes 18-19 contain data stream flags. Byte 20 contains media flags.
				FM Header Flags:
••• Flags for FM header set 0				
16	0-7			Reserved
17	0-3			Reserved
(17)	4-7			Reserved
••• Flags for FM header set 1 or 2				
16	0	B'0'		Two destinations may be outstanding. The primary half-session may interrupt itself once.
		B'1'	X	Three destinations may be outstanding. The primary half-session may interrupt itself twice.
(16)	1	B'0'		Compacted data must not be sent.
		B'1'	X	Compacted data may be sent.
(16)	2	B'0'		PDIR not allowed.
		B'1'	X	PDIR may be sent by primary half-session.
(16)	3-7			Reserved
17	0-3			Reserved
(17)	4-7			Reserved
••• Flags for FM header set 3				
16	0	B'0'		Two destinations may be outstanding. The primary half-session may interrupt itself once.
		B'1'	X	Three destinations may be outstanding. The primary half-session may interrupt itself twice.
(16)	1	B'0'		Compacted data must not be sent.
		B'1'	X	Compacted data may be sent.
(16)	2	B'0'		PDIR not allowed.
		B'1'	X	PDIR may be sent by primary half-session.
(16)	3	B'0'		Keyed Direct data sets not allowed.
		B'1'	X	Keyed Direct data sets allowed. (Permits use of ADD, ERASE, and RECORD ID FMH-2s on Keyed Direct data sets.)
(16)	4	B'0'		Sequential data sets not allowed.
		B'1'	X	Sequential data sets allowed. (Permits use of ADD and ERASE FMH-2s.)
(16)	5	B'0'		Sequential access to addressed direct data sets not allowed.
		B'1'	X	Sequential access to addressed direct data sets allowed. (Permits use of ADD, NOTE, and NOTE REPLY FMH-2s.)
(16)	6	B'0'		Use of SERIES ID and STATUS FMH-3s not allowed.
		B'1'	X	Use of SERIES ID and STATUS FMH-3s allowed.
(16)	7	B'0'		Use of ADD REPLICATE and REPLACE REPLICATE FMH-2s not allowed.
		B'1'	X	Use of ADD REPLICATE and REPLACE REPLICATE FMH-2s allowed.
17	0			Reserved

Byte	Bits	Value	Opt	Discussion
(17)	1	B'0'		Use of QUERY FOR DATA SET FMH-2 not allowed.
		B'1'	X	Use of QUERY FOR DATA SET FMH-2 allowed.
(17)	2	B'0'		Use of CREATE DATA SET, SCRATCH DATA SET, and SCRATCH ALL DATA SETS FMH-2s not allowed.
		B'1'	X	Use of CREATE DATA SET, SCRATCH DATA SET, and SCRATCH ALL DATA SETS FMH-2s allowed.
(17)	3	B'0'		Use of EXECUTE PROGRAM OFFLINE FMH-2 not allowed.
		B'1'	X	Use of EXECUTE PROGRAM OFFLINE FMH-2 allowed.
(17)	4-7			Reserved
(18-19)				Data Stream Flags Note: There is a hierarchy of support within data streams (byte 18, bits 0-3); that is, if horizontal format is supported, so must the full-base and the base data streams.
18	0	B'0'		Base NL (new line) FF (form feed)
		B'1'	X	Full base Includes: base data stream plus BS (backspace) CR (carriage return) INP (inhibit presentation) ENP (enable presentation) LF (line feed) - defaultable HT (horizontal tab) - defaultable VT (vertical tab) - defaultable Defaults are defined in Part 2 Chapter 1.
(18)	1	B'0'		Horizontal format must not be used.
		B'1'	X	Horizontal format may be used. Includes: full base plus SHF (set horizontal format) (with parameters)
(18)	2	B'0'		Vertical format must not be used.
		B'1'	X	Vertical format may be used. Includes: Horizontal format plus SVF (set vertical format) (with parameters)
(18)	3	B'0'		Vertical channel must not be used.
		B'1'	X	Vertical channel may be used. Includes: Vertical format plus VCS (vertical channel select) (with parameters)
(18)	4	B'0'		SLD (set line density) must not be used.
		B'1'	X	SLD may be used.
(18)	5			Reserved
(18)	6	B'0'		BEL (bell) must not be used.
		B'1'	X	BEL may be used.
(18)	7	B'0'		TRN (transparent) and IRS (interchange record separator) must not be used.
		B'1'	X	TRN and IRS may be used.
19	0	B'0'		Secondary half-session will initiate attended mode.
		B'1'	X	Secondary half-session will initiate unattended mode.
(19)	1	B'0'		During session, secondary half-session will not alternate from attended to unattended mode.
		B'1'	X	During session, secondary half-session may alternate between attended and unattended mode.

Byte	Bits	Value	Opt	Discussion
(19)	2-7			Reserved
20				Media flags
(20)	0	B'0'		Document format is not allowed.
		B'1'	X	Document format is allowed.
(20)	1	B'0'		Card format is not allowed.
		B'1'	X	Card format is allowed.
(20)	2	B'0'		Exchange media format is not allowed.
		B'1'	X	Exchange media format is allowed.
(20)	3	B'0'		Disk data management format is not allowed.
		B'1'	X	Disk data management format is allowed.
(20)	4	B'0'		Extended card format is not allowed.
		B'1'	X	Extended card format is allowed.
(20)	5	B'0'		Extended document format is not allowed.
		B'1'	X	Extended document format is allowed.
(20)	6	B'0'		Secondary half-session may send CD at EDS (end of destination selection).
		B'1'	X	Secondary half-session must send CD at EDS.
(20)	7			Reserved

Secondary Half-Session Usage

(Primary Half-Session Receive Capability)

(21-25) Bytes 21-22 contain FM header flags.
Bytes 23-24 contain data stream flags.
Byte 25 contains media flags.

FM Header Flags:

••• Flags for FM header set 0

21	0-7			Reserved
22	0-3			Reserved
(22)	4	B'0'		Use of structured field for ERP not allowed.
		B'1'	X	Use of structured field for ERP allowed.
(22)	5-7	B'000'		Reserved

••• Flags for FM header set 1 or 2

21	0	B'0'		Two destinations may be outstanding. The secondary half-session may interrupt itself once.
		B'1'	X	Three destinations may be outstanding. The secondary half-session may interrupt itself twice.
(21)	1	B'0'		Compacted data must not be sent.
		B'1'	X	Compacted data may be sent.
(21)	2	B'0'		PDIR not allowed.
		B'1'	X	PDIR may be sent by secondary half-session.
(21)	3-7			Reserved
22	0-3			Reserved
	4-7			Reserved

Byte	Bits	Value	Opt	Discussion
••• Flags for FM header set 3				
21	0	B'0'		Two destinations may be outstanding. The secondary half-session may interrupt itself once.
		B'1'	X	Three destinations may be outstanding. The secondary half-session may interrupt itself twice.
(21)	1	B'0'		Compacted data must not be sent.
		B'1'	X	Compacted data may be sent.
(21)	2	B'0'		PDIR not allowed.
		B'1'	X	PDIR may be sent by secondary half-session.
(21)	3	B'0'		Keyed direct data sets not allowed.
		B'1'	X	Keyed direct data sets allowed. (Permits use of ADD, ERASE, and RECORD ID FMH-2s on keyed direct data sets.)
(21)	4	B'0'		Sequential data sets not allowed.
		B'1'	X	Sequential data sets allowed. (Permits use of ADD and ERASE FMH-2s.)
(21)	5	B'0'		Sequential access to addressed direct data sets not allowed.
		B'1'	X	Sequential access to addressed direct data sets allowed. (Permits use of ADD, NOTE, and NOTE REPLY FMH-2s.)
(21)	6	B'0'		Use of SERIES ID and STATUS FMH-3s not allowed.
		B'1'	X	Use of SERIES ID and STATUS FMH-3s allowed.
(21)	7	B'0'		Use of ADD REPLICATE and REPLACE REPLICATE FMH-2s not allowed.
		B'1'	X	Use of ADD REPLICATE and REPLACE REPLICATE FMH-2s allowed.
22	0			Reserved
(22)	1	B'0'		Use of QUERY FOR DATA SET FMH-2 not allowed.
		B'1'	X	Use of QUERY FOR DATA SET FMH-2 allowed.
(22)	2	B'0'		Use of CREATE DATA SET, SCRATCH DATA SET, and SCRATCH ALL DATA SETS FMH-2s not allowed.
		B'1'	X	Use of CREATE DATA SET, SCRATCH DATA SET, and SCRATCH ALL DATA SETS FMH-2s allowed.
(22)	3	B'0'		Use of EXECUTE PROGRAM OFFLINE FMH-2 not allowed.
		B'1'	X	Use of EXECUTE PROGRAM OFFLINE FMH-2 allowed.
(22)	4-7			Reserved
(23-24)				Data Stream Flags Note: There is a hierarchy of support within data streams (byte 23, bits 0-3). That is, if horizontal format is supported, so must the full-base and the base data streams.
23	0	B'0'		Base Includes (for console devices): NL (new line) FF (form feed)
		B'1'	X	Full base Includes: Base data stream plus BS (backspace) CR (carriage return) INP (inhibit presentation) ENP (enable presentation) LF (line feed) - defaultable HT (horizontal tab) - defaultable VT (vertical tab) - defaultable Defaults are defined in Part 2 Chapter 1.

Byte	Bits	Value	Opt	Discussion
(23)	1	B'0'		Horizontal format must not be used.
		B'1'	X	Horizontal format may be used. Includes: Full base plus SHF (set horizontal format) (with parameters)
(23)	2	B'0'		Vertical format must not be used.
		B'1'	X	Vertical format may be used. Includes: Horizontal format plus SVF (set vertical format) (with parameters)
(23)	3	B'0'		Vertical channel must not be used.
		B'1'	X	Vertical channel may be used. Includes: Vertical format plus VCS (vertical channel select) (with parameters)
(23)	4	B'0'		SLD (set line density) must not be used.
		B'1'	X	SLD may be used.
(23)	5			Reserved
(23)	6	B'0'		BEL (bell) must not be used.
		B'1'	X	BEL may be used.
(23)	7	B'0'		TRN (transparent) and IRS (interchange record separator) must not be used.
		B'1'	X	TRN and IRS may be used.
24	0-7			Reserved
25				Media flags
(25)	0	B'0'		Document format is not allowed.
		B'1'	X	Document format is allowed.
(25)	1	B'0'		Card format is not allowed.
		B'1'	X	Card format is allowed.
(25)	2	B'0'		Exchange media format is not allowed.
		B'1'	X	Exchange media format is allowed.
(25)	3	B'0'		Disk data management format is not allowed.
		B'1'	X	Disk data management format is allowed.
(25)	4	B'0'		Extended card format is not allowed.
		B'1'	X	Extended card format is allowed.
(25)	5	B'0'		Extended document format is not allowed.
		B'1'	X	Extended document format is allowed.
(25)	6-7			Reserved

Cryptography Options

26	0-1			Private cryptography options:
		B'00'	X	No private cryptography supported.
		B'01'	X	Private cryptography supported. The session cryptography key and cryptography protocols are privately supplied by the end user.
		B'10'		Reserved
		B'11'		Reserved

Byte	Bits	Value	Opt	Discussion
(26)	2-3			Session-level cryptography options:
		B'00'	X	No session-level cryptography supported.
		B'01'	X	Session-level selective cryptography supported. All cryptography key management is supported by the SSCP and LU; exchange (using +RSP to BIND) and verification (using CRV) of the cryptography session-seed value are supported by the LUs for the session. All FM data requests with EDI (enciphered data indicator) on are enciphered/deciphered.
		B'10'		Reserved
		B'11'	X	Session-level mandatory cryptography supported; same as session-level selective cryptography except all FM data requests are enciphered/deciphered.
(26)	4-7			Session-level cryptography options field length:
		X'0'	X	No session-level cryptography specified. The cryptography option fields (bytes 27-k) are omitted.
		X'9'	X	Session-level cryptography specified. Additional options follow in the next nine bytes.
27	0-1			Session cryptography key encipherment mode:
		B'00'	X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key using a seed value of zero (only value defined).
(27)	2-4			Reserved
(27)	5-7			Cryptography cipher method:
		B'000'	X	Block chaining with seed and cipher text feedback, using the data encryption standard (DES) algorithm (only value defined).
28-k			X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key; an 8-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FM data requests.

Names and End User Data

k+1			Length of BIND sender's (or PLU) name in field k+2-m.
	X'00'		Field k+2-m does not exist.
	X'nn'	X	Field k+2-m contains the number of bytes specified (cannot exceed X'08').
k+2-m		X	BIND sender's (or PLU) name.
m+1			Length of user data in field m+2-n.
	X'00'		Field m+2-n does not exist.
	X'nn'	X	Field m+2-n contains the number of bytes specified.
m+2-n		X	User data
m+2		X	User data key.
	X'00'	X	Structured subfields follow.
	-X'00'	X	First byte of unstructured user data.

Note: The unstructured user data continues in m+3 and goes to byte n. For information on structured subfields, see the BIND information in SNA Format and Protocol Reference Manual: Architecture Logic.

Byte	Bits	Value	Opt	Discussion
n+1				Length of user request correlation field.
		X'00'		Field n+2-p does not exist.
		X'nn'	X	Field n+2-p contains the number of bytes specified.
n+2-p			X	User request correlation field.
p+1				Length of BIND receiver's (or SLU) name in field p+2-r.
		X'00'		Field p+2-r does not exist.
		X'nn'	X	Field p+2-r contains the number of bytes specified (cannot exceed X'08').
p+2-r			X	BIND receiver's (or SLU's) name.

Notes

The following notes apply to the BIND SESSION request:

Note 1. The buffer contention error (081B) response cannot be sent by the primary half-session, since the secondary half-session wins contention.

Note 2. There is never an EB race in LU_T1 because the only time multiple-chain brackets with EB are sent from the secondary half-session is under the HDX-FF BIND option.

Note 3. If BIND allows the secondary half-session to send requests in either exception response form or definite response form, the secondary half-session always uses the definite response form unless it has enough logic to understand additional data-dependent protocols implied by the primary half-session name.

Note 4. If compression/compaction is used, all RUs in a chain being compressed/compacted must contain an SCB as their first byte (after any FM headers). This implies that the first RU of the chain (or the only RU of the chain) resets SCB chaining.

Type 2 LU-LU sessions support data communication between an application program and a single display device using the SNA 3270 data stream. Type 2 sessions allow users to migrate existing customer-written 3270 application programs into an SNA network.

Type 2 sessions associate a single device with the SLU. The logical device within the SLU is visible to the PLU. Consistency is maintained between a given physical device and the LU for the duration of a unit of work, for example, a bracket. The same physical device is always associated with a particular SLU. The device may be dedicated permanently or dedicated only for the duration of a bracket or a session; however, it is always the same physical device. Except where otherwise noted, the description assumes a dedicated display device.

Products that use type 2 sessions use the SNA 3270 data stream. The data stream is introduced in Part 2 Chapter 2. For details of the data stream, see 3270 Data Stream Programmer's Reference. When the following options are implemented, they must conform to the protocols of the SNA 3270 data stream:

1. Selector pen. When a selector pen is implemented, providing the ampersand (&) designator to simulate an Enter key is optional.

2. Extended screen size. The SLU supports either a 480- or 1920-character screen size operation. When the screen is physically larger, the screen wraps at the 480 or 1920 point, depending on the value specified in byte 24 of the BIND SESSION request.

   The SLU may support larger or smaller screen sizes; up to a 4096-byte addressing limit with 14-bit addressing, and up to 65,536 with 16-bit addressing.

3. Screen size switching. An SLU has the option of providing dynamic screen size switching. The screen size may be switched between two values, specified in BIND SESSION during session activation, by the Erase/Write and the Erase/Write Alternate commands. The Erase/Write command selects the default size while the Erase/Write Alternate command selects the alternate size. The screen size is initialized to the default size.

4. Display of lowercase. Lowercase alphabetic characters sent to the display or entered from the keyboard are stored as lowercase but will be displayed as uppercase on those devices that can't display lowercase characters.

5. Magnetic stripe reader. The numeric and alphanumeric character sets are supported.

The above options can be implemented using a base set of data stream functions. The SNA 3270 data stream has additional functions that provide the following functions for LU-LU session type 2 implementations:

1. Screen sizes smaller than 480 characters

2. Partitioning. The display screen can be divided into several rectangular areas. The operator interacts with one area at a time, but may jump from one area to another if necessary.

3. Scrolling. Scrolling allows a presentation space to be larger than the viewing area the operator sees. The operator can use the scrolling function to display different portions of the character data available at the terminal.

4. Character sets, color, and highlighting. The application program can control the character set to be used for a field or character, as well as control the characters color and any emphasis that it should have (such as blinking or underlining).

5. Field validation. Field validation functions that can be provided are (1) checking a field to ensure that the operator filled every character position in it, (2) requiring the operator to enter data in a field, and (3) providing an automatic enter operation as soon as the operator finishes processing of the field by moving the cursor out of the field.

CHARACTERISTICS OF TYPE 2 SESSIONS

Because a type 2 session is most commonly associated with a display and its keyboard, the session has some characteristics that do not apply to other session types. These characteristics are described below.

DEVICE SHARING

Device sharing is the use of a device (or medium) by more than one session. The SLU allows two sharing configurations:

- Printer shared for display printout (copy)
- Display shared between the SSCP-SLU and PLU-SLU sessions

When a printer is shared between sessions, the contents of a screen can be copied to the printer. The allocation and scheduling of the printer is a local (controller) function. When a display is shared, concurrent sessions can be held between the SLU and a PLU and the SLU and its SSCP. The scheduling of messages to the display is an SLU function. In both cases, the sharing is not directly visible to the PLU.

DISPLAY PRINTOUT (COPY)

To provide a hard copy capability, the SLU must have - in addition to the display device - the use of a printer device. Although the printer device could be permanently allocated to one SLU, the more likely configuration is one in which one or more printer devices are shared among a group of SLUs. The physical printer that is shared among several SLUs might also be used by half-sessions of other session types. (The SLU is still considered a single-device SLU since the printer does not appear as a device to the PLU.)

There is no forced print element positioning, for example, NL or FF, as a result of starting or ending a display printout. This does not preclude an automatic new line that is built into a printer operation, or an automatic form feed before or after a display printout.

The request for a hard-copy printout of a display screen may be initiated from the host or by the operator:

- Host initiated. The printout is initiated by the PLU sending a display printout request (write commands with WCC = Start Print). The display printout request must be sent either RQD or RQE,CD-EB. See "Error Conditions" for details on printer unavailable and other error conditions associated with the hard copy function.
- Operator initiated. An operator-initiated printout is limited to the function in which the operator at the display initiates a printout of the operator's screen. The operator-initiated printout should be handled locally (in the controller and not in the SLU). No PLU interaction is required or desired at the time of initiation.

SLU-PLU/SLU-SSCP DISPLAY SHARING

An SLU simultaneously supports a session with a PLU and a session with the SSCP. This requires that the SLU resolve any presentation space contention that may occur between the SSCP-LU and LU-LU sessions. (See "Error Conditions" on page 42 for information on SLU-PLU/SLU-SSCP contention.)

KEYBOARDLESS DISPLAY CONSIDERATIONS

When an SLU receives a request with CD and the SLU is aware that nothing can be sent (for example, keyboardless display), LUSTAT X'0002' (no FM data to transmit) is sent to the PLU.

PROTOCOLS

Flip-Flop and Contention

LU type 2 supports half-duplex flip-flop (HDX-FF) operation. Optionally, it may also support half-duplex contention operation. HDX-FF operation is required if the PLU must control or be aware of the display screen.

Certain situations producing unpredictable results limit the usefulness of contention mode. An example is the support of the read commands. In an HDX flip-flop session, each read command must carry the CD indicator and the PLU must assume that the next chain it receives is the reply to the read command. In an HDX contention session, contention exists at the end of each chain. Therefore, unanticipated inputs can be expected from the PLU. This requires the SLU to be able to accept input from either the device or the PLU. In this mode, the PLU cannot be sure that data received after a read command is in fact the reply to the command. In fact, there is no guarantee that the next SLU-to-PLU chain occurred after the previous PLU-to-SLU chain. Accurate correlation of input and output cannot be made. This mode requires the PLU to treat each flow independently even though the SLU does not.

The remainder of this section is oriented to HDX-FF.

Type 2 sessions use the concept of a formatted screen to reduce the amount of data that must be transferred from the display station to the PLU and vice versa. When the screen is formatted, only those fields defined as modified are transferred in reply to a read modified operation. For traffic from the PLU to the display, the PLU may keep track of the format of the display and therefore reduce the amount of data that must be transferred to the display to alter the image. This approach can only be depended on for HDX-FF operation. If contention is used, the PLU cannot directly control the screen and therefore cannot utilize this technique.

A PLU may send two types of data to the SLU: (1) only an RH, used to control the direction of flow (for example, an RH with change direction or end bracket specified), and (2) a 3270 command sequence with associated data when appropriate. For convenience in describing the control of the screen, two of the 3270 command sequences are named and defined:

- Formatting Write. This consists of an Erase/Write command code followed by a WCC byte. Although the Erase/Write, Erase/Write Alternate, and Write Control Character can technically be considered a formatting write by an SLU, usually one or more SBA/SF attribute sequences are also sent to provide the format. A formatting write assumes the display screen is logically blank and therefore transfers both the constant (nonvarying) part of the display image and those fields that may be modified by subsequent operation action.

- Formatted Write. This consists of a Write command code followed by a WCC byte and one or more 'SBA + data' sequences. The SBA(s) select which field(s) of the formatted screen the associated data applies to.

The state of the screen as understood by the PLU determines the type of operation that can be performed by the PLU in transferring data to the SLU to achieve minimal transmission. A formatted screen needs to receive only modified fields from the PLU. An unformatted screen must receive the constants and literals that make up the display format as well as the modified fields. Except for forcing an unformatted screen following acceptance of BIND SESSION, the SLU enforces no screen states.

EB/CD Relations to SLU States

End Bracket (EB): An SLU that receives an EB usually executes the associated 3270 command (other than RB/RM/RMA) and then goes into the contention state with the keyboard restored and the AID cleared. If the command does not explicitly restore the keyboard (and AID), it forces a restore.

The exceptions to the above are the result of error conditions. On all the error conditions, bracket termination rule 1 applies. The SLU action for PSA and PSE error is described under "SLU Detected Errors" later in this chapter.

An EB chain with a RB/RM/RMA command is considered an error condition. The rationale is that the EB is terminating a unit of work while the RB/RM/RMA command is extending the unit of work and they are thus in conflict.

Change Direction (CD): The normal result of an SLU receiving a CD in an RU chain containing a 3270 command (other than RB/RM/RMA) is for the SLU to execute the command and then go to send state. An RH-only chain with CD does the same except that there is no command to be executed. The keyboard is restored, and the AID is cleared only when explicitly done so by the command. The CD chain with RB/RM/RMA causes the SLU to execute the command, send data with CD, and return to the receive state.

Exceptions to the above are the result of error conditions. (See "SLU Detected Errors" on page 46).

Summary: Figure 3-1 summarizes the EB/CD/PSA relationship for an SLU in receive state. The purpose of the table is to define SLU action if a particular RU chain is received. The appearance of a particular chain in the table does not mean the PLU should or will send it.

Chaining and Segmenting

Type 2 sessions support multiple-element RU chains and segmenting. For an HDX-FF session, RU chains from an SLU carry CD if the RH indicates FMD; an RU chain without CD is allowed if the RH indicates DFC. (For example, an RU chain containing user data is an FMD chain and an LUSTAT request is a DFC chain.)

Only one 3270 command is allowed per RU chain. In addition, when a type 2 half-session receives a chain element that contains a command, the command must be in the first byte of the RU. If required, the WCC must be the next byte following the command and must be in the same RU. There are no restrictions on sending RH-only RUs.

ERROR CONDITIONS

All errors are turned into exception requests by the element of the network that detects the error and sent forward (except for loss of path errors) to the destination LU. The destination LU then processes the exception requests and returns a negative response.

SLU Receives Chain with:			SLU Action					
3270 Command	EB	CD	PSA Pend?	Exec CMD?	Clr AID & Reset Kbd?	Clr PSA?	-RSP Sent	S/R State
W,EW,EWA,EAU,Null	1	-	N	Y(3)	Y	-	-	C
RB,RM,RMA(1)chk	1	-	N	N	N	-	0829	CE1
RB,RM,RMA(1)¬chk	1	-	N	Y	N	-	-	S
W,EW,EWA,EAU	1	-	Y	Y	(2)	Y	082A(4)	CE1
Null	1	-	Y	RM(3)	N	Y	-	S
RB,RM,RMA(1)chk	1	-	Y	N	N	N	0829	CE1
RB,RM,RMA(1)¬chk	1	-	Y	N	N	Y	084A	CE1
W,EW,EWA,EAU,Null	0	1	N	Y(3)	(2)	-	-	S
RB,RM,RMA	0	1	N	Y	N	-	-	S
W,EW,EWA,EAU	0	1	Y	Y	(2)	Y	082A(4)	CE1
Null	0	1	Y	RM(3)	N	Y	-	S
RB,RM,RMA	0	1	Y	N	N	Y	084A	CE1
W,EW,EWA,EAU	0	0	N	Y	(2)	-	-	R
Null	0	0	N	NOP(3)	N	-	-	R
RB,RM,RMA	0	0	N	N	N	-	0829	CE1
W,EW,EWA,EAU	0	0	Y	Y	(2)	Y	082A(4)	CE1
Null	0	0	Y	NOP(3)	N	N	-	R
RB,RM,RMA	0	0	Y	N	N	N	0829	CE1

Abbreviations used:

W = Write
EW = Erase Write
EWA = Erase Write Alternate
EAU = Erase All Unprotected
RM = Read Modified
RMA = Read Modified All
RB = Read Buffer
CD = Change Direction
EB = End Bracket
S = Send
R = Receive
C = Contention
PSA = Presentation Space Altered
CE1 = Contention ERP-1

A hyphen indicates a 'not applicable' or 'don't care'.

A write command (W, EW or EWA) with the WCC set to Start Print (SP) is defined as a request for a printout (hard copy) of the display. See "Display Printout" earlier in this chapter for details of the display printout function. The SLU action shown in the summary table applies, except that the command execution consists of first updating the screen if data accompanies the write command and then performing the printout of the display screen.

When a keyboard is locked, certain 'control' keys may still be operable. For example, in the 3277, the reset key is operable in all SLU states whether or not the keyboard is locked.

Note 1: An RB, RM or RMA received with EB is invalid and should not occur. In one option, the SLU checks for the condition; in the other, the SLU does not check for the condition.

Note 2: Keyboard is restored and AID is cleared only if the command either is EAU or is a W, EW or EWA with WCC = Keyboard Restore.

Note 3: Null means an RU chain with only an RH and therefore no command to execute.

Note 4: The SLU may choose to handle the EW, EWA chain differently. If the SLU implementation is such that a formatting write sequence (EW, EWA usually followed by at least one SF sequence) can be recognized and if the results of prior operator action are overwritten, then the SLU may clear the PSA, not send a negative response, and go to send state if CD is set.

Figure 3-1. SLU action summary (receive state)

SENSE CODE SUMMARY

All state errors (sequence numbers, chaining, brackets) may flow in both directions. Function management data (FMD) errors may be sent by the PLU. The following codes may be sent by the SLU.

Request Reject - FM data (category code X'08'):

0801 Resource not available (printer)
0802 Intervention required (display)
0807 Resource not available, LUSTAT forthcoming (printer)
080B Bracket race error
0813 Bracket bid reject - no RTR forthcoming
0814 Bracket bid reject - RTR forthcoming
081B Receiver in transmit mode
081C Request not executable (display)
0829 Change direction required
082A Presentation space alteration, request executed (contrast with X'084A')
082B Presentation space integrity lost
082D LU busy (SLU resource already in use)
082E Intervention required at LU subsidiary device (printer)
082F Request not executable because of LU subsidiary device (printer)
0831 LU component disconnected
0843 Required FMDS synchronization not supplied (for example, a request received WCC=SP,RQE,¬CD)
084A Presentation space alteration, request not executed (contrast with X'082A')
084C Permanent insufficient resource (the PS buffer resource required by Load PS is not available)
0863 Referenced local character set ID (LCID) not found
0868 No panels loaded
0869 Panel not loaded
0871 Read Partition state error

Request Reject - BIND SESSION (Category code X'08'):

080A Permission rejected - SSCP will not be notified (contrast with X'0845')
0821 Invalid session parameters (response to BIND)
0833 Invalid parameter (with pointer and complemented byte)
0845 Permission rejected - SSCP will be notified (contrast with X'080A')

Request Errors (category code X'10'):

1001 RU data error
1003 Function not supported
  Invalid 3270 command code
  Data following RB, RM, RMA or EAU command
1005 Parameter error
  Invalid address follows SBA, or SBA or SF without parameters
1007 Category not supported (for example, unsolicited FM data request on SSCP-LU session)
1009 Format group not selected (no format group was selected before issuing a Present Absolute Format or Present Relative Format structured field)

The SLU may send the following status codes in LUSTAT:

0001 Component now available (Notes 1 and 2)
0002 No FM data to transmit (Note 3)
0801 Component not available (for example, no resources configured) (Note 1)
081C Component failure (Note 1)
082B Component available, presentation space integrity lost (Notes 1 and 2)
0831 Component disconnected (power off or some other disconnecting condition) (Note 1)

The format of LUSTAT is:

Byte	Value	Definition
0	X'04'	Request code
1-4		Status value and status extension field (two bytes each)
	X'0001'+'cc00'	Component now available (0001) and component identification (cc00).
	X'0002'+'rrrr'	No FMD requests to send and reserved field.
	X'0801'+'cc00'	Component not available (0801) and component identification (cc00)
	X'081C'+'cc00'	Component failure - intervention required (081C) and component identification (cc00)
	X'082B'+'cc00'	Component available but presentation space integrity lost (082B) and component identification (cc00)
	X'0831'+'cc00'	Component disconnected (0831) and component identification (cc00)

Note 1: Values for 'cc' byte are:

X'00' = LU itself rather than a specific LU component.

Otherwise,
Bit 0, set to 1 = Media other than LU
Bits 1-3, LU component class:

'011' Printer
'101' Display
All other values reserved.

Bits 4-7, LU component device address.

Note 2: LUSTAT X'0001' and LUSTAT X'082B' both indicate availability. The sense code the SLU uses depends on the presentation space state. If the presentation space is damaged or suspected of being damaged, LUSTAT X'082B' is used. LUSTAT X'0001' implies that the presentation space was not affected and that resumption, including the resending of a rejected request, will cause no presentation space problem.

Note 3: LUSTAT X'0002' is used if the change direction indicator is received and the SLU is unable to send FM data; it is not used when the SLU is waiting for keyboard input.

The RU chain containing LUSTAT cannot begin a bracket.

CANCEL REQUEST

A sender-detected error occurs when the RU chain sender decides to abnormally terminate the current chain by sending the CANCEL request. This action is controlled by the chain sender and is independent of any action by the chain receiver. If the PLU receives such an error and the PLU is keeping track of screen states, it assumes the display format has been altered and replies with a formatting write.

The PLU may abort the chain it is sending by issuing CANCEL. The meaning of such an error is defined by the operating environment of the session. Since the SLU is free to transmit RUs to the display screen as they are received, receipt of CANCEL leaves the screen in an indeterminate state (that is, up to the last RU before CANCEL). The SLU is not required to update its current state. All type 2 sessions are bound with PLU recovery responsibility. Therefore, the SLU goes into receive state and awaits further PLU action. The PLU has two recovery options:

In most situations, the PLU may retransmit precisely that chain it chose to abort. This results in the first part of the chain being reexecuted and the remainder of the chain being executed as originally intended. This would normally be chosen only if the chain was formatted write, and if the chain started with an explicit SBA order, and if there were no positioning references (such as PT) that were revised later in the chain. In short, the PLU must understand the chain content before attempting to retransmit.

The PLU may transmit a formatting write. This rebuilds the screen format and leaves the SLU screen state as formatted.

Any other action leaves the screen state in an unpredictable format. If the PLU is maintaining screen awareness and chooses to end the current bracket, the PLU must also exercise one of the above options or the PLU may not be able to interpret the first request of the new bracket (if one is begun by the SLU).

PLU DETECTED ERRORS

The PLU may reject an RU chain from the SLU for any valid reason. This includes so-called standard errors as well as FM unique errors (for example, "no such transaction code" from a transaction processing subsystem). When a chain is rejected, the PLU sends a negative response to the SLU and enters undefined state. The SLU responds by sending CANCEL if a chain is still in progress and enters receive state to await PLU action. The PLU then sends an error message to direct the next operator action. Since the PLU has entered an undefined state, this must be a formatting write if screen awareness is to be maintained. The error message may continue the current bracket by sending a CD or a ¬CD, or may end the current bracket by sending EB.

SLU DETECTED ERRORS

The SLU may detect errors in receiving an RU chain. The type of error determines the state of the screen assumed by the PLU. Three different categories are identified:

- State errors. These are sequence numbers, chain, and bracket protocol errors. The SLU sends the correct error response and enters purging chain state. Upon exit from the purging chain state, the SLU awaits PLU action unless bracket termination also occurred, in which case contention is entered. Normal error recovery action is for the PLU to terminate the current session by sending UNBIND SESSION.

- Permanent device errors. This type of error is due to some hardware malfunction between the SLU and the display head (for example, buffer parity error). When the error is detected, the SLU sends a negative response (X'081C'), enters purging chain state, and, upon exit, enters receive or contention state as defined for state errors above.

- Temporary device errors. When this type of error occurs, the SLU sends the appropriate negative response (for example, X'0802' intervention required). When the error condition is cleared, the appropriate LUSTAT is used to report correction of the error. If presentation space integrity has been lost, the SLU sends LUSTAT X'082B'. If presentation space integrity has not been lost, the SLU sends LUSTAT X'0001'.

The following defines the action for certain error conditions:

1. Power off. An SLU may encounter a power-off (display) situation either at BIND or during an established session.

   If a power-off condition exists at BIND, the SLU rejects the BIND SESSION request with a negative response of 0845 or 080A. The 0845 (permission rejected: SSCP will be notified), when sent to reject BIND SESSION, indicates that the SLU will notify the SSCP when another BIND SESSION can be accepted. The 080A (permission rejected: SSCP will not be notified), when sent to reject BIND SESSION, indicates that the LU will not notify the SSCP.

   If a power-off condition occurs during an established session, and the SLU is in the receive state, the SLU sends a negative response (X'0831') to indicate to the PLU that a power-off condition exists. If power-off occurs while the SLU is in the send or contention state, an LUSTAT may be sent (also X'0831'). When power has been restored, the SLU sends the appropriate LUSTAT status code to notify the PLU that the component is available: X'0001' or, if presentation space integrity has been lost, X'082B'. The SSCP may also be notified at this time.

2. Presentation space altered (PSA) (X'082A' or X'084A'). To ensure PLU screen awareness/control when operating HDX-FF, the operator should not alter the screen while the SLU is in receive state. However, if the PLU restores the keyboard while the SLU is in receive state or if the display permits reset and enter capability regardless of SLU state, the screen can be altered. Type 2 sessions allows the reset and enter operation regardless of SLU state and records a presentation space altered (PSA) pending whenever the operator performs a reset and enter operation while the SLU is in receive state. However, the SLU is not required to record a PSA pending on an operator enter operation following an explicit PLU keyboard unlock (Write with WCC = keyboard restore) that leaves the SLU in the receive state.

   Whether the PSA pending results in a negative response depends on what the SLU subsequently receives from the PLU:

   All chains except an RH-only chain and RB/RM/RMA chain: The command is executed. A Write command with WCC = keyboard restore, and Erase/Write command with WCC = keyboard restore, EWA command with WCC = keyboard restore, or an EAU command clears the AID and restores the keyboard. If a PSA pending exists at the completion of the command execution, the SLU sends a negative response X'082A', resets the PSA pending, and awaits PLU recovery action.

   RB/RM/RMA chain: If a PSA pending exists when the RB/RM/RMA chain is received, the command is not executed, and a negative response 084A is sent to notify the PLU of the PSA pending condition. This resets the PSA pending, and the SLU awaits PLU recovery action. If the data is required, another RB/RM/RMA chain must be sent.

   RH-only chain: SLU action depends on setting of EB/CD. (See "EB/CD Relation to SLU States" above for details.) Briefly, if a PSA pending exists when an RH-only chain with EB and with or without CD is received, the SLU clears the PSA pending, negative response is not sent, and the SLU goes to the state appropriate for the EB/CD setting.

   In all the above cases, the normal PLU recovery is to send a formatting write (EW or EWA command - usually with one or more SFs). Since the PSA pending is cleared once the negative response is sent, the SLU attempts to execute commands other than the formatting write. That is, the negative response is not repeated unless the operator causes a new PSA pending.

3. Presentation space error (PSE) (X'082B'). This error condition is set as a result of detecting an error in the presentation space that is not locally recoverable; for example, a regeneration buffer parity error or operator action such as allowing SSCP use of the screen. When the screen size switching is allowed, presentation space integrity includes screen size as well as content. The X'082B' error results from other than the operator action described under the X'082A' error. The X'082B' error is presumed to be transient and therefore recoverable by the PLU sending a

formatting write. However, if the error condition is not actually a transient but repeats, the SLU changes to negative response X'081C'.

When the error is detected while the SLU is in receive state, it is reported via negative response X'082B'. If the error is detected while the SLU is in send or contention state, an LUSTAT is sent (also X'082B').

If a locally initiated display printout fails and results in a damaged or questionable presentation space, the SLU sends negative response X'082B' or the equivalent LUSTAT depending on the SLU state.

Although the formatting write is the normal recovery for both PSA and PSE, note that the clearing of a PSA pending by an RH-only CD or EB chain is not applicable to PSE (X'082B'). When an X'082B' is pending and an EW or EWA command is received, the SLU may reset the X'082B'; that is, not send the negative response or LUSTAT request.

4. Display printout request ('Copy'). The display printout request is accomplished by sending a write command (Write, Erase/Write or Erase/Write Alternate) with WCC = Start Print (bit 4 of WCC) to the SLU. If the write command contains data, the display is updated before the printout operation. The display printout request must be sent as an RQD chain or an RQE, CD ¬EB chain. Since the PLU is responsible for error recovery, the SLU does not have to check for RQE, CD, ¬EB chains. Therefore, if the PLU sends a display printout request as an RQE, ¬CD or RQE, CD, EB chain and follows with a screen update request, the SLU may attempt to update the screen before doing the print. If the SLU does check, an RQE, ¬CD or RQE, CD, EB is rejected with negative response X'0843' (required FMDS synchronization not supplied). (The 'hard copy' function provided by the SLU does not utilize a copy control character.)

   a. Logical printer definition. Several printers may be configured for an SLU's use for display printouts. A display printout request to the SLU could result in a display printout on any of the configured printers, depending on availability and the allocation algorithm implemented. From a PLU view, however, there is only one logical printer regardless of how many physical printers are configured for the SLU.

      A logical printer is considered available if any of the physical printers configured for the SLU are able to immediately execute a display printout request. When none of the configured printers is available, the logical printer assumes the status of the most available of the physical printers. The following are the unavailable categories listed in order of best to worst:

      Short-term busy. A display printout request can be executed after "n" queued-up display printout requests are executed. The maximum delay is 30 minutes. Use of the printer by a session is not allowed.

      Intervention required condition exists.

      Allocated for session use or local use.

      Permanent error condition (or an intervention required situation on an unattended printer).

      No logical printer configured.

   b. Logical printer not available. The SLU may reject a display printout request when the logical printer is not available. The following are the sense codes used:

      X'082E'. The logical printer has an intervention required condition, for example, out of paper, power off, cover interlock open, and so on. This code can be used only if the printer is attended. In general, attended means the condition will likely receive quick attention, such as when an operator is attending the printer or a mechanism exists to alert an operator. If the printer is unattended, the permanently unavailable code is used.

X'0807'. The logical printer is busy for an indeterminate period of time. This may range from a relatively short time when the printer is being used by another SLU for a display printout to many hours if all printers are assigned locally on a long term basis. (The SLU masks short term busy from the PLU by withholding negative response X'0807'.)

X'082F'. Effectively, the logical printer is permanently not available. For example, an unattended printer is out of paper.

X'0801'. No printer configured.

c. LUSTAT requirements. After sending negative response X'082E' or X'0807', an LUSTAT is sent when the condition clears. However, regardless of how many times the request is repeated, only one LUSTAT is sent when the condition does clear. Sending an LUSTAT X'0001' (source = printer) indicates to the PLU that a display printout request will find either an available or short term busy logical printer. Exceptions to the above are:

   The LUSTAT X'0001' (source = printer) reporting the clear of an X'082E' or X'0807' condition is not sent if:

   After sending negative response X'082E' or X'0807' the data traffic subtree is reset or the session terminated.

   After sending the logical negative response X'082E or X'0807' the printer develops a permanent error and is dropped from the configuration. An LUSTAT X'081C' or X'0801' (source = printer) is sent instead of LUSTAT X'0001'.

   The LUSTAT X'0001' (source = printer) does not indicate the PLU will necessarily find an available or short-term busy logical printer if:

   After sending negative response X'082E' or X'0807', the SLU receives any normal FM data request chain other than a display printout request chain.

   After sending LUSTAT X'0001' (source = printer), the logical printer develops an intervention required, permanent error, or a not-configured condition. The display printout request is rejected with the appropriate negative response; that is, X'082E', X'082F' or X'0801'.

   After sending negative response X'0807' or X'082E', the SLU detects a nonprinter exception condition that causes a negative response.

   Generally, implementations do not allow session contention for a printer before honoring outstanding LUSTATs; that is, at least one physical printer is held after having sent the LUSTAT(s) so that the logical printer is available to execute copy requests. However, if as a result of an exception condition developing on the held printer, the logical printer status of 'in session' results, then negative response X'0807' may also be sent.

d. Recovery considerations. The negative responses X'082E', X'0807', X'082F', and X'0801' indicate to the PLU that if the display printout request included screen update data, the screen update was accomplished. If a display printout request (with update data) is sent again, it cannot be guaranteed that the screen will be unchanged. An example where the screen would be changed is if the request was a write to cursor, and the cursor position was altered later in the data stream. Therefore, following the receipt of an LUSTAT associated with a previous negative response X'0807' or X'082E', the PLU may do one of the following:

   Resend the display printout request (with update data) if the PLU is aware that the data stream is such that resending will not alter the screen.

   Resend the display printout request without the update data.

Rebuild the screen followed by display printout request with update data.

e. PSA considerations. If a presentation-space-altered pending condition exists when the copy of the display is transferred to the printer, the printer operation completes and a negative response X'082A' is sent. This response indicates to the PLU that the display printout was accomplished, but the display screen may have been altered. If the printout operation failed, the appropriate negative response would be sent rather than the X'082A' response; however, the PSA condition remains.

5. SLU-PLU/SLU-SSCP. When the SLU cannot accept a PLU request because the SSCP has the use of the screen, the request is rejected with negative response X'082D'. After the SSCP-SLU session relinquishes the screen, the SLU returns an LUSTAT to indicate that the PLU has regained use of the screen. The type of LUSTAT returned depends on the SLU screen sharing implementation. If the SLU maintains screen integrity (such as storing a copy of the screen) during the SSCP interruption, LUSTAT X'0001' (component available) is sent. If the SLU does not maintain screen integrity, LUSTAT X'082B', (component available, screen integrity lost) is sent. Note that when the SLU implementation does not maintain screen integrity, the LUSTAT is sent every time the SSCP-LU relinquishes the screen, regardless of whether or not a negative response X'082D' has been sent previously. The normal PLU recovery following LUSTAT X'082B' would be to rebuild the screen.

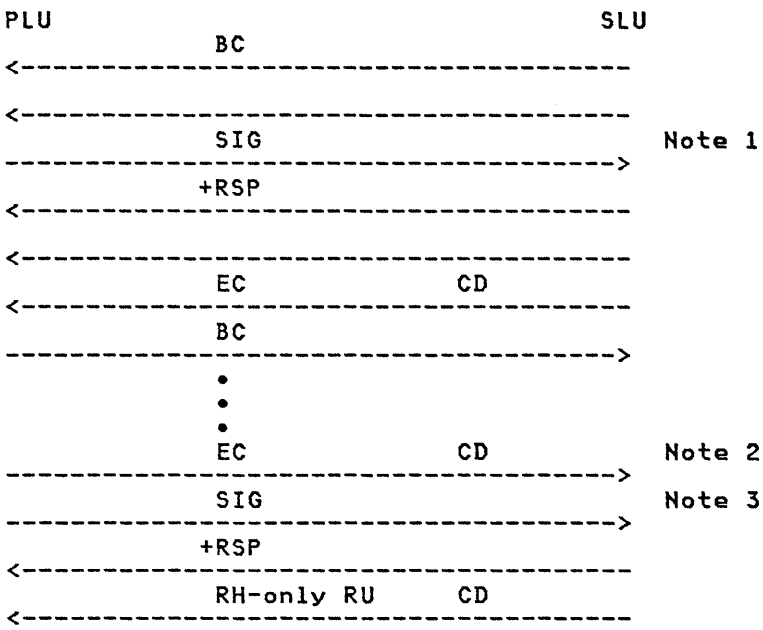
6. SLU busy. The SLU may be unable to accept a request from the PLU because the SLU or its resources are already in use. An example would be if a local test routine was being exercised. A request received when the SLU is busy is rejected with negative response X'082D'. Negative response X'082D' requires that an LUSTAT be sent when the condition clears (see SLU-PLU/SLU-SSCP, item 5, for details).

7. Keyboardless display considerations. When an SLU receives a request with CD and the SLU is aware that nothing can be sent (for example, a keyboardless display), LUSTAT X'0002' is sent to the PLU.

SAMPLE SEQUENCES

The following examples illustrate basic data exchange sequences. The first example shows how a receiver alerts the sender when the receiver has data to send. The session is established and the partners are in brackets. The remaining examples show how the session partners (1) can get from contention between bracket state to in bracket state, and (2) can exchange data between them. The response is not shown unless it is a necessary part of the example.

EXAMPLE 1: Signal from PLU

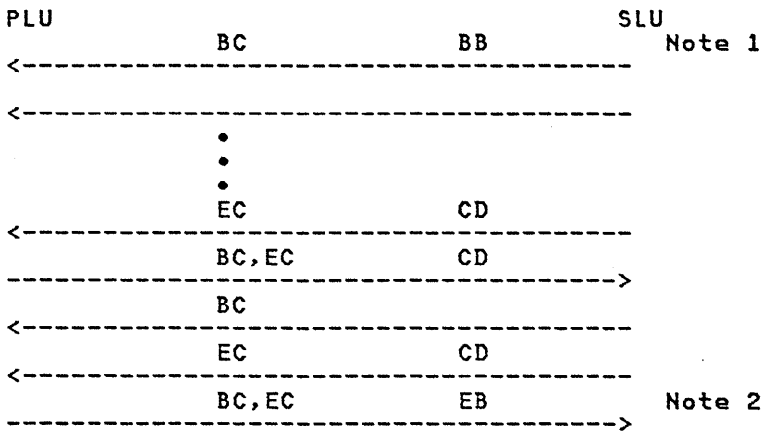


Note 1: The SLU receives SIG request while sending. The +RSP is returned to acknowledge receipt of the signal. The signal is effectively treated as no operation, and the SLU completes sending of the chain. The SLU always sends CD with the end of a data chain.

Note 2: CD allows the SLU to send. The operator starts keying in data.

Note 3: Before the operator initiates sending of data (for example, presses the ENTER key), the PLU sends SIG. The SLU sends +RSP to the signal, locks the keyboard, and sends CD.

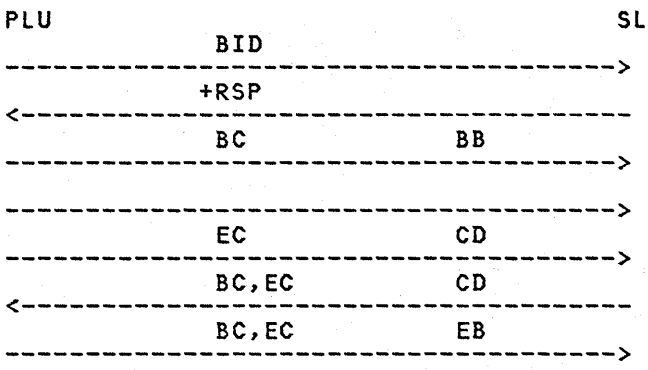
EXAMPLE 2: Bracket/Chain - SLU Initiated (Without Contention)



Note 1: SLU initiates a bracket and sends an RU chain as a result, for example, of ENTER key depression.

Note 2: After the required exchange of chains is completed, the PLU ends the unit of work by sending EB. (An SLU cannot send EB.) The chain with EB may contain data; for example, a write to the screen; or it may be an RH-only chain.

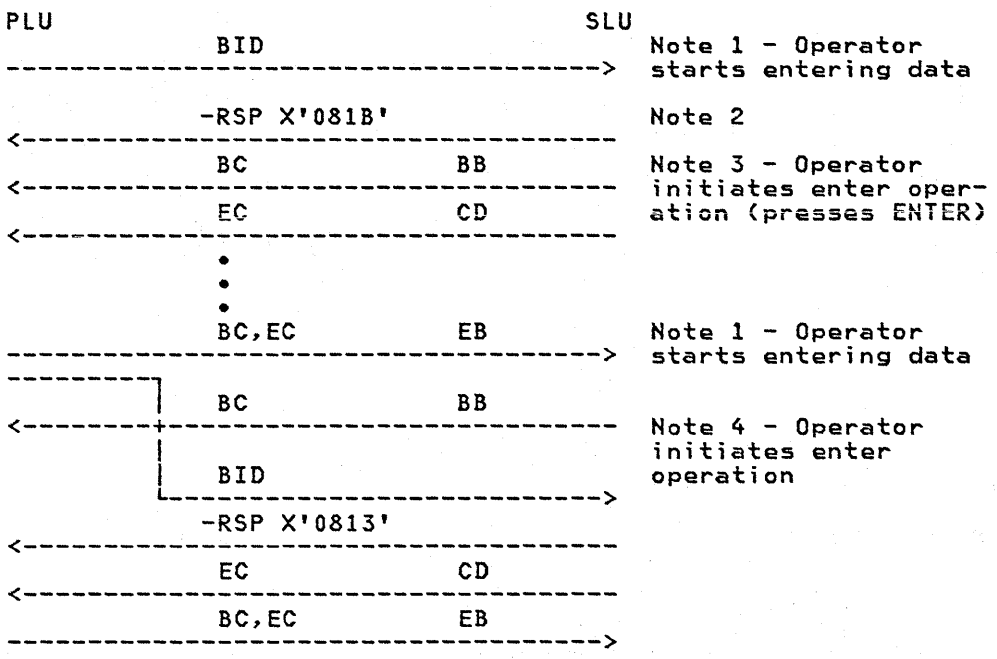
EXAMPLE 3: Bracket/Chain - PLU Initiated (Without Contention)



Note 1: PLU sends BID request to indicate intention to begin a bracket.

Note 2: The positive response to BID allows the PLU to initiate the unit of work with BB.

EXAMPLE 4: Bracket/Chain - PLU/SLU Contention



Note 1: The first keystroke puts the SLU in the send (but not transmitting) state. The SLU remains in BETB state.

Note 2: The SLU will reject a BID (or BB with X'081B' - receiver in transmit mode).

Note 3: The operator initiates an enter operation - for example, presses the ENTER key. The SLU begins a bracket and transmits the entered data.

Note 4: When the operator presses the ENTER key, the SLU goes to in bracket (INB) state. The SLU begins a bracket and starts sending data. The PLU has sent a BID (or BB) before the first chain element was received. The SLU rejects the BID (or BB) with X'0813', bracket bid reject.

BIND SESSION FORMAT FOR LU-LU SESSION TYPE 2

This section defines valid bit settings for the BIND request. The BIND sender selects the values to be used. Any value may be specified except those identified as not valid. All values listed must be supported by the BIND receiver unless the value is optional (identified by X in the Opt column) or not valid.

Byte	Bits	Value	Opt	Discussion
0	0-7	X'31'		Identifies this RU as a BIND request.
1	0-3	X'0'		Specifies the format of the BIND RU. Only one format has been defined: format 0 (zero).
(1)	4-7			Type - denotes the type of BIND to be performed:
		X'0'		Not valid
		X'1'		Nonnegotiable BIND

FM (Function Management) Profile

2	0-7	X'03'		FM profile 3
				The FM profile defines data flow control (DFC) protocols. (See SNA Reference Summary for highlights of the FM profile.)

TS (Transmission Subsystem) Profile

3	0-7	X'03'		TS profile 3
				The TS profile defines transmission control (TC) protocols. (See SNA Reference Summary for highlights of the TS profile.)

FM Usage

(4-7)				See bytes 4 through 7 below.

Primary Half-Session Protocols for FM Data

				Chaining Use:
4	0	B'0'		Primary half-session can send only single element chains.
		B'1'		Primary half-session can send single or multiple element chains.
				Request Mode Selection:
(4)	1	B'0'		Immediate request mode is used. Only one definite response can be outstanding at a time. That response must be received before the primary half-session can send another RU.
		B'1'		Not valid
				Chaining Responses:
(4)	2-3	B'00'		Not valid
		B'01'		Primary half-session can request only exception responses.
		B'10'		Primary half-session can request only definite response.
		B'11'		Primary half-session can request definite or exception responses.

Byte	Bits	Value	Opt	Discussion
(4)	4-5			Reserved
				Compression Indicator:
(4)	6	B'0'		Primary half-session cannot send compressed data.
		B'1'		Not valid
				Send End Bracket Indicator:
(4)	7	B'0'		Not valid
		B'1'		Primary half-session can send EB.

Secondary Half-Session Protocols for FM Data

				Chaining Use:
5	0	B'0'		Not valid
		B'1'		Secondary half-session can send single or multiple element chains.
				Request Mode Selection:
(5)	1	B'0'		Immediate request mode is used. Secondary half-session can issue a request for a single definite response. No further transmission is sent until the secondary half-session receives the requested response.
		B'1'		Not valid
				Chaining Responses:
(5)	2-3	B'00'		Not valid
		B'01'		Secondary half-session can request only exception responses.
		B'10'		Secondary half-session can request only definite responses.
		B'11'		Secondary half-session can request either definite or exception responses.
(5)	4-5			Reserved
				Compression Indicator:
(5)	6	B'0'		Secondary half-session cannot send compressed data.
		B'1'		Not valid
				Send End Bracket Indicator:
(5)	7	B'0'		Secondary half-session cannot send EB.
		B'1'		Not valid

Common Protocols for FM Data

6	0			Reserved
(6)	1	B'0'		Primary half-session and secondary half-session cannot exchange FM headers.
		B'1'		Not valid
				Brackets Usage and Reset State:
(6)	2	B'0'		Not valid
		B'1'		Brackets are used and bracket state manager's reset state is BETB (between brackets).

Byte	Bits	Value	Opt	Discussion
				Bracket Termination Rule Selection:
(6)	3	B'0'		Not valid
		B'1'		Bracket termination rule 1 is used. Bracket termination is controlled by the form of response requested (definite or exception) for the chain containing (¬BB,EB).
				Alternate Code Selection:
(6)	4	B'0'		Alternate code set will not be used.
		B'1'	X	Alternate code set may be used.
				If byte 6, bit 4, is B'1', the code selection indicator (CSI field of the RH (request header)) controls whether the alternate code is to be used: CSI of 0 = EBCDIC CSI of 1 = ASCII
(6)	5-7			Reserved
				Normal-Flow Send and Receive Mode Selection:
7	0-1	B'00'		Not valid
		B'01'		Not valid
		B'10'		Half-duplex flip-flop
				Bracketed half-duplex flip-flop sessions require: Byte 4, bit 2 or 3 must be set to 1 (primary half-session chaining response). Byte 4, bit 7, or byte 5, bit 7, or both must be set to 1 (send EB). Byte 5, bit 2 or 3 must be set to 1 (secondary half-session chaining response). Byte 6, bit 2 must be set to 1 (bracketed session). Byte 7, bit 0 and 1 must be set to 10 (half-duplex flip-flop). Byte 7, bit 3 must be set to 0 (secondary half-session is the contention winner). Byte 7, bit 7 must be set to 0 (reset state is receive for primary and send for secondary).
		B'11'		Reserved
				Recovery Responsibility:
(7)	2	B'0'		The contention loser is responsible for any error recovery attempts for the session.
		B'1'		Not valid
				Contention Winner/Loser:
(7)	3	B'0'		Secondary LU is contention winner and primary LU is contention loser.
		B'1'		Not valid
(7)	4-6			Reserved
				Half-Duplex Flip-Flop (HDX-FF) Reset States:
(7)	7			Reserved

TS Usage

(8-13)	See bytes 8 through 13 below.

Byte	Bits	Value	Opt	Discussion
				Staging Indicator and Pacing Count for Secondary CPMGR (connection point manager) to Primary CPMGR normal flow:
8	0	B'0' B'1'		Pacing in this direction occurs in one stage. Pacing in this direction occurs in two stages.
				Note: The meanings of 0 and 1 are reversed from the staging indicator for primary CPMGR to secondary CPMGR (see byte 12).
(8)	1			Reserved
(8)	2-7	B'nnnnnn'		Secondary CPMGR send pacing count. A value of 000000 means no pacing of requests flowing from the secondary.
9	0-1			Reserved
(9)	2-7	B'nnnnnn'		Secondary CPMGR receive pacing count. A value of 000000 causes the boundary function to substitute the value set by a system definition pacing parameter (if the system definition includes such a parameter) before it sends the BIND RU on to the secondary LU. A value of 000000 received at the secondary LU is interpreted to mean no pacing of requests flowing to the secondary LU.
10		X'ab'	X	Maximum RU size sent on the normal flow by the secondary half-session. This value represents the largest RU that can be sent. It is expressed as a mantissa and an exponent value of 2 by which the mantissa is multiplied.
		X'85'		When the mantissa is 8 and the exponent is 5 (X'85'), the RU size is 256 bytes (8×2^5).
		B'0n...'		When bit 0 is set to zero, no maximum is specified and the remaining bits 1-7 are ignored. When bit 0 is set to one, the byte is interpreted as X'ab'. See Appendix B for these values.
11		X'ab'	X	Maximum RU size sent on the normal flow by the primary half-session. This value represents the largest RU that can be sent by the primary half-session and is specified in the same format as the secondary half-session (byte 10).
		X'85'		When the mantissa is 8 and the exponent is 5 (X'85'), the RU size is 256 bytes (8×2^5).
		B'0n...'		When bit 0 is set to zero, no maximum is specified and the remaining bits 1-7 are ignored. When bit 0 is set to one, the byte is interpreted as X'ab'.
				Staging Indicator and Pacing Count for Primary CPMGR (connection point manager) to Secondary CPMGR normal flow:
12	0	B'0' B'1'		Pacing in this direction occurs in two stages. Pacing in this direction occurs in one stage.
				Note: The meanings of 0 and 1 are reversed from the staging indicator for secondary CPMGR to primary CPMGR (see byte 8).
(12)	1			Reserved
(12)	2-7	B'nnnnnn'		Primary CPMGR send pacing count. A value of 000000 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 000000, it means no pacing of requests flowing from the primary. For single-stage pacing in the primary-to-secondary direction, this field is redundant with, and will indicate the same value as, the secondary CPMGR receive pacing count (see byte 9, bits 2-7 above).

Byte	Bits	Value	Opt	Discussion
13	0-1			Reserved
(13)	2-7	B'nnnnnn'		Primary CPMGR receive pacing count. A value of 000000 means no pacing of requests that flow to the primary. For single-stage pacing in the secondary-to-primary direction, this field is redundant with, and will indicate the same value as, the secondary CPMGR send pacing count (see byte 8, bits 2-7 above).
PS Profile				
(14-25)				See bytes 14 through 25 below.
14	0	B'0'		PS usage field format is the basic format.
		B'1'		Reserved
(14)	1-7	B'0000010'		LU-LU session type 2
PS Usage				
15	0	B'0'		Query not supported
		B'1'	X	Query supported
	1-7			Reserved
16-19				Reserved
				Screen-Size Control:
				Use byte 24 to specify a 480- or 1920-character screen size for the base set of the SNA 3270 data stream. Use bytes 20-24 to specify screen sizes for implementations that use more than the base set of the SNA 3270 data stream. (See Note 1.)
20	0-7		X	Default screen size: rows. This is the number of rows in the screen, stated in binary. Permissible values are from 1 to 255.
21	0-7		X	Default screen size: columns. This is the number of columns in the screen, stated in binary. Permissible values are from 1 to 255.
22	0-7		X	Alternate screen size: rows. This is the number of rows in the screen, stated in binary. Permissible values are from 1 to 255.
23	0-7		X	Alternate screen size: columns. This is the number of columns in the screen, stated in binary. Permissible values are from 1 to 255.
24	0			Reserved
(24)	1-7			Screen size (see Note 1)
		0000000		Not specified
		0000001	X	12 rows by 40 columns (480 characters). (See Note 2.)
		0000010	X	24 rows by 80 columns (1920 characters). (See Note 2.)
		0000011	X	Unspecified screen size. (See Note 1.)
		1111110	X	Static size specified by bytes 20 and 21. (See Note 3.)
		1111111	X	Default size specified by bytes 20 and 21, and alternate size specified by bytes 22 and 23. (See Note 4.)
				All other values reserved.
25				Reserved

Byte	Bits	Value	Opt	Discussion
Cryptography Options				
26	0-1			Private cryptography options:
		B'00'	X	No private cryptography supported.
		B'01'	X	Private cryptography supported. The session cryptography key and cryptography protocols are privately supplied by the end user.
		B'10'		Reserved
		B'11'		Reserved
(26)	2-3			Session-level cryptography options:
		B'00'	X	No session-level cryptography supported.
		B'01'	X	Session-level selective cryptography supported. All cryptography key management is supported by the SSCP and LU; exchange (using +RSP to BIND) and verification (using CRV) of the cryptography session-seed value are supported by the LUs for the session. All FM data requests with EDI (enciphered data indicator) on are enciphered/deciphered.
		B'10'		Reserved
		B'11'	X	Session-level mandatory cryptography supported; same as session-level selective cryptography except all FM data requests are enciphered/deciphered.
(26)	4-7			Session-level cryptography options field length:
		X'0'	X	No session-level cryptography specified. The cryptography option fields (bytes 27-k) are omitted.
		X'9'	X	Session-level cryptography specified. Additional options follow in the next nine bytes.
27	0-1			Session cryptography key encipherment mode:
		B'00'	X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key using a seed value of zero (only value defined).
(27)	2-4			Reserved
(27)	5-7			Cryptography cipher method:
		B'000'	X	Block chaining with seed and cipher text feedback, using the Data Encryption Standard (DES) algorithm (only value defined).
28-k			X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key; an eight-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FM data requests.
Names and End User Data				
k+1				Length of BIND sender's (or PLU) name in field k+2-m.
		X'00'		Field k+2-m does not exist.
		X'nn'	X	Field k+2-m contains the number of bytes specified (cannot exceed X'08').
k+2-m			X	BIND sender's (or PLU) name.
m+1				Length of user data in field m+2-n.
		X'00'		Field m+2-n does not exist.
		X'nn'	X	Field m+2-n contains the number of bytes specified.

Byte	Bits	Value	Opt	Discussion
m+2-n			X	User data.
m+2			X	User data key.
		X'00'	X	Structured subfields follow.
		¬X'00'	X	First byte of unstructured user data.
				Note: The unstructured user data continues in m+3 and goes to byte n. For information on structured subfields, see the BIND information in SNA Format and Protocol Reference Manual: Architecture Logic.
n+1				Length of user request correlation field.
		X'00'		Field n+2-p does not exist.
		X'nn'	X	Field n+2-p contains the number of bytes specified.
n+2-p			X	User request correlation field.
p+1				Length of BIND receiver's (or SLU) name in field p+2-r.
		X'00'		Field p+2-r does not exist.
		X'nn'	X	Field p+2-r contains the number of bytes specified (cannot exceed X'08').
p+2-r			X	BIND receiver's (or SLU) name.

Notes

The following notes apply to the BIND SESSION request:

Note 1. Screen sizes can be specified using byte 24, or byte 24 can refer to the screen sizes defined in bytes 20-21, and, as required, bytes 22-23. If the base set of the SNA 3270 data stream is used, byte 24, bits 1-7, may specify only B'0000000', B'0000001', or B'0000010'. The B'0000000' value permits either a 480 or 1920 screen size. If any functions above the base set of the SNA 3270 data stream are used, byte 24 may specify any of the defined values.

If byte 24, bits 1-7 specify B'0000011', the default screen size is 1920 characters (24 x 80) and the alternate screen size is specified in the Query Reply (Usable Area) structured field.

If the data being loaded in the display buffer exceeds the designated screen limit, the remainder of the data is wrapped, that is, loaded starting at the first buffer position. With one exception, the display screen wraps at the value(s) specified in BIND regardless of the actual size of the screen. The exception is the byte 24, bits 1-7, value of B'0000000'; the screen wraps at the end of the actual screen size (480 or 1920).

Note 2: The screen size is fixed for the duration of the session when this mode is selected. When in this mode, an Erase/Write Alternate command may be rejected with negative response X'1003'.

Note 3: If byte 24, bits 1-7, contain B'1111110', the screen size specified in bytes 20-21 is static. Bytes 22-23 are not used. When in this mode, an Erase/Write Alternate command may be rejected with negative response X'1003'. Valid row and column parameters for default bytes 20-21 can be found in implementation literature. The product of the number of rows (byte 20) and columns (byte 21) must not exceed 4096 for the base set of the SNA 3270 data stream (upper limit of order stream addressing), and 65536 when the base set is extended with additional functions.

Note 4: An alternate screen size may be specified if byte 24, bits 1-7, contain B'1111111' (dynamic screen size switching within a session).

The BIND operation establishes both default (bytes 20-21) and alternate (bytes 22-23) screen sizes to be used within a session. It initializes the unit by setting it in the default screen size mode.

An Erase/Write Alternate command changes the screen size from default to alternate. An Erase/Write command or depression of the CLEAR key changes the screen size from alternate to default.

Valid row and column parameters for default bytes 20-21 and alternate bytes 22-23 can be found in implementation literature. The product of the number of rows and columns must not exceed 4096 for the base set of the SNA 3270 data stream (upper limit of order stream addressing), and 65536 when the base set is extended with additional functions.

Type 3 LU-LU sessions support data communication between an application program and a single printer (without a keyboard) using the SNA 3270 data stream. Type 3 sessions allow users to migrate existing customer-written 3270 application programs into an SNA network.

Type 3 sessions associate a single device with the SLU. The logical device within the SLU is visible to the PLU. Consistency is maintained between a given physical device and the LU for the duration of a unit of work, for example, a bracket. The same physical device is always associated with a particular SLU. The device may be dedicated permanently or dedicated only for the duration of a bracket or a session: however, it is always the same physical device. Except where otherwise noted, the description assumes a dedicated printer.

Products that use type 3 sessions use the SNA 3270 data stream. The data stream is introduced in Part 2 Chapter 2. (For details of the data stream, see 3270 Data Stream Programmer's Reference.) When the following options are implemented, they must conform to the protocols of the SNA 3270 data stream:

1. Buffer size. The SLU supports either a 480- or 1920-character print buffer size. This applies when the physical buffer size is larger than the required basic 480 (12x40) or 1920 (24x80) size. When the buffer is physically larger, the buffer wraps at the 480 or 1920 point, depending on the value specified in byte 24 of the BIND SESSION request.

   The SLU may support larger and smaller buffer sizes; up to a 4096-byte addressing limit with 14-bit addressing, and up to 65,536 with 16-bit addressing.

2. Buffer size switching. In addition to the basic and extended buffer sizes described above, the SLU may provide dynamic buffer size switching. The buffer size may be switched between two values, specified in BIND SESSION, by the Erase/Write and the Erase/Write Alternate commands. The Erase/Write command selects the default size while the Erase/Write Alternate command selects the alternate size. The buffer size is initialized to the default size.

The above options can be implemented using a mandatory set of data stream functions. The SNA 3270 data stream has options that provide additional character set, color, and highlighting functions for LU-LU session type 3 implementations. The application program can control the character set to be used for a field or character, as well as control the characters color and any emphasis that it should have (such as underlining). The following functions are specifically excluded:

Multiple partitions
Scrolling
Field validation

Part 2 Chapter 2 shows which data stream functions are not applicable to type 3 sessions.

CHARACTERISTICS OF TYPE 3 SESSIONS

The profiles that define type 3 sessions differ slightly from those that define type 1 and 2 sessions. These differences are discussed here.

DEVICE SHARING

In addition to providing a dedicated printer configuration, the SLU allows sharing of the printer for local use; for example, for type 2 display printouts as well as for type 3 session use. The use for type 2 printouts may be on a between-session or between-bracket basis. If a request should arrive from the PLU while the printer is in use, the SLU rejects it with a negative response (X'0814') followed by RTR (ready to receive) when the printer is available.

The allocation and scheduling of the shared device(s) is a local (controller) function. The sharing is not directly visible to the PLU and may not be visible to the SLU.

The SLU does not provide any forced print element positioning (for example, NL or FF as a result of an EB or BB), but an implementation optionally may do so.

When a printer is configured for sharing between brackets, the PLU must assume that both the print buffer and print element positioning has changed between brackets. The SLU does not inform the PLU whether the printer was used. The same applies to the between-session interval when sharing between sessions is configured.

PROTOCOLS

Chaining and Segmenting

The SLU receives multiple element chains. Support of segmenting is optional. The SLU does not send FM data chains.

Only one 3270 command is allowed per RU chain. In addition, when an SLU receives a chain element that contains a command, the command must be the first byte of the RU; the WCC must be the next byte following the command and must be in the same RU.

There are no restrictions on sending RH-only RUs to an SLU.

Half-Duplex Flip-Flop

The SLU operates only in HDX-FF mode. It always sends LUSTAT X'0002' with CD to return the flow immediately whenever the PLU sends CD with any request.

RQD and Start Print

All write commands with the start print bit set on in the WCC must be sent either RQD (request definite response) or RQE,CD (request exception response, change direction). The PLU is responsible for enforcement. If the SLU elects to check, however, a request with WCC = start-print,RQE,¬CD is rejected with negative response X'0843' (required FMDS synchronization not provided).

A positive response is not sent for requests with WCC = start-print,RQD until the print operation completes.

If the start print bit is not set on in the WCC, the SLU allows sending the chain RQE,¬CD.

Reading the Buffer

A type 3 printer accepts the same commands as a type 2 display except that Read, Read Modified, and Read Modified All are not accepted.

Because multiple partitions are not allowed, a Read Partition structured field is invalid for any use other than to perform a Query operation. The Read Modified command is accepted to retry a query operation. (The content of the character buffer cannot be read.)

ERROR CONDITIONS

All errors are turned into exception requests by the element of the network that detects the error and sent forward (except for the loss of path errors) to the destination LU. The LU then processes the exception request and returns a negative response.

SENSE CODE SUMMARY

All state errors (sequence numbers, chaining, brackets) may flow in both directions. Function management data (FMD) errors may be sent by the PLU. The following codes may be sent by the SLU.

Request Reject - FM data (category code X'08'):

0801 Resource not available (printer not configured)
0802 Intervention required
0814 Bracket bid reject - RTR forthcoming
081B Receiver in transmit mode
081C Request not executable
082B Presentation space integrity lost
082D LU busy (SLU resource already in use)
0831 LU component disconnected
0843 Required FMDS synchronization not supplied
(for example, a request received WCC=SP,RQE,¬CD)
084C Permanent insufficient resource
(the PS buffer resource required by Load PS
is not available)
0863 Referenced local character set ID (LCID) not found
087F Query Reply parameter changed - RU chain processed
0880 Query Reply parameter changed - RU chain not processed

Request Reject - BIND SESSION (category code X'08'):

080A Permission rejected - SSCP will not be notified
(contrast with X'084A')
0821 Invalid session parameters
0845 Permission rejected - SSCP will be notified
(contrast with X'080A')

Request Error: 10XX

1001 RU data error
1003 Function not supported
Invalid 3270 command code
Data following EAU command
Attempt to read character buffer
1005 Parameter error
1007 Category not supported (for example, FM data request
received on SSCP-LU session)

The SLU may send the following status codes in LUSTAT:

0001 Component now available (notes 1 and 2)
0002 No FM data to transmit
0801 Component not available (for example, no resources configured (note 1)
081C Component failure (note 1)
082B Component available, presentation space integrity lost (notes 1 and 2)
0831 Component disconnected (power off or some other disconnecting condition) (note 1)

The format of LUSTAT is:

Byte	Value	Definition
0	X'04'	Request code
1-4		Status value and status extension field (two bytes each)
	X'0001'+'cc00'	Component now available (0001) and component identification (cc00) (see note)
	X'0002'+'rrrr'	No FMD requests to send and reserved field.
	X'0801'+'cc00'	Component not available (0801) and component identification (cc00)
	X'081C'+'cc00'	Component failure - intervention required (081C) and component identification (cc00)
	X'082B'+'cc00'	Component available, but presentation space integrity lost (082B) and component identification (cc00)
	X'0831'+'cc00'	Component disconnected (0831) and component identification (cc00)

Note 1: Values for 'cc' byte are:

X'00' = LU itself rather than a specific LU component.

Otherwise,
Bit 0, set to 1 = media other than LU
Bits 1-3, LU component class:

'011' Printer
'101' Display
All other values reserved.

Bits 4-7, LU component device address.

Note 2: LUSTAT X'0001' and LUSTAT X'082B' both indicate availability. The sense code the SLU uses depends on the presentation space state. If the presentation space is damaged or suspect of being damaged, LUSTAT X'082B' is used. LUSTAT X'0001' implies that the presentation space was not affected and that resumption, including resending a rejected request, will cause no presentation space problem.

SLU DETECTED ERRORS

The SLU may detect errors in receiving an RU chain. The type of error detected determines the state of the buffer to be assumed by the PLU. Four different categories are identified:

- State, sequence-number, chain, and bracket-protocol errors. The SLU sends the correct error response and enters purging chain state. Upon exit from the purging chain state, the SLU goes to receive state unless bracket termination also occurred (in which case contention is entered). Normal recovery action is for the PLU to terminate the current session by sending UNBIND SESSION.

- Permanent device errors. This type of error is due to some hardware malfunction between the SLU and the printer (for example, buffer parity error). When the error is detected, the SLU sends a negative response (X'081C'), enters purging chain state, and, upon exit, enters receive or contention state as defined for state errors.

- Temporary device errors. When this type of error occurs, the SLU sends a negative response (X'0802' intervention required). When the error condition is cleared, the appropriate LUSTAT is used to report correction of the error. If presentation space integrity has been lost, the SLU sends LUSTAT X'082B'. If presentation space integrity has not been lost, the SLU sends LUSTAT X'0001', component available.

- Power off. An SLU may encounter a printer-off situation either at BIND or during a session.

If a power-off condition exists at BIND, the SLU rejects the BIND with either negative response X'0845' or X'080A'. The X'0845', when sent to reject a BIND, indicates that the SLU will notify the SSCP when a BIND can be accepted. The X'080A', when sent to reject a BIND, indicates that the SLU will not notify the SSCP when a BIND can be accepted.

If a power-off condition occurs during a session, and the SLU is in the receive state, the SLU sends negative response X'0831' to indicate to the PLU that a power-off condition exists. If power-off occurs while the SLU is in the send or contention state, an LUSTAT may be sent (same error code). When power has been restored, the SLU sends the appropriate LUSTAT status code to notify the PLU that the component is available: X'0001' or, if presentation space integrity has been lost, X'082B'.

BIND SESSION FORMAT FOR LU-LU SESSION TYPE 3

This section defines valid bit settings for the BIND request. The BIND sender selects the values to be used. Any value may be specified except those identified as not valid. All values listed must be supported by the BIND receiver unless the value is optional (identified by X in the Opt column) or not valid.

Byte	Bits	Value	Opt	Discussion
0	0-7	X'31'		Identifies this RU as a BIND request.
1	0-3	X'0'		Specifies the format of the BIND RU. Only one format has been defined: format 0 (zero).
(1)	4-7			Type - denotes the type of BIND to be performed:
		X'0'		Not valid
		X'1'		Nonnegotiable BIND

FM (Function Management) Profile

2	0-7	X'03'		FM profile 3
				The FM profile defines data flow control (DFC) protocols. (See SNA Reference Summary for highlights of the FM profile.)

TS (Transmission Subsystem) Profile

3	0-7	X'03'		TS profile 3
				The TS profile defines transmission control (TC) protocols. (See SNA Reference Summary for highlights of the TS profile.)

FM Usage

(4-7)				See bytes 4 through 7 below.

Primary Half-Session Protocols for FM Data

				Chaining Use:
4	0	B'0'		Primary half-session can send only single element chains.
		B'1'		Primary half-session can send single or multiple element chains.
				Request Mode Selection:
(4)	1	B'0'		Immediate request mode is used. Only one definite response can be outstanding at a time. That response must be received before the primary half-session can send another RU.
		B'1'		Not valid
				Chaining Responses:
(4)	2-3	B'00'		Not valid
		B'01'		Primary half-session can request only exception-only responses.
		B'10'		Primary half-session can request only definite response.
		B'11'		Primary half-session can request definite or exception-only responses.

Byte	Bits	Value	Opt	Discussion
(4)	4-5			Reserved
				Compression Indicator:
(4)	6	B'0'		Primary half-session cannot send compressed data.
		B'1'		Not valid
				Send End Bracket Indicator:
(4)	7	B'0'		Not valid
		B'1'		Primary half-session can send EB.

Secondary Half-Session Protocols for FM Data

				Chaining Use:
5	0	B'0'	X	Secondary half-session can send only single element chains.
		B'1'		Secondary half-session can send single or multiple element chains.
				Request Mode Selection:
(5)	1	B'0'		Immediate request mode is used. Secondary half-session can issue a request for a single definite response. No further transmission is sent until the secondary half-session receives the requested response.
		B'1'		Not valid
				Chaining Responses:
(5)	2-3	B'00'		Not valid
		B'01'		Secondary half-session can request only exception responses.
		B'10'		Secondary half-session can request only definite responses.
		B'11'		Secondary half-session can request either definite or exception responses.
(5)	4-5			Reserved
				Compression Indicator:
(5)	6	B'0'		Secondary half-session cannot send compressed data.
		B'1'		Not valid
				Send End Bracket Indicator:
(5)	7	B'0'		Secondary half-session cannot send EB.
		B'1'		Not valid

Common Protocols for FM Data

6	0		Reserved
(6)	1	B'0'	Primary half-session and secondary half-session cannot exchange FM headers.
		B'1'	Not valid
			Brackets Usage and Reset State:
(6)	2	B'0'	Not valid
		B'1'	Brackets are used and bracket state manager's reset state is BETB (between brackets).
			Bracket Termination Rule Selection:
(6)	3	B'0'	Not valid

Byte	Bits	Value	Opt	Discussion
		B'1'		Bracket termination rule 1 is used. Bracket termination is controlled by the form of response requested (definite or exception) for the chain containing (¬BB,EB).
				Alternate Code Selection:
(6)	4	B'0'		Alternate code set will not be used.
		B'1'	X	Alternate code set may be used.
				If byte 6, bit 4, is B'1', the code selection indicator (CSI field of the RH (request header)) controls whether the alternate code is to be used: CSI of 0 = EBCDIC CSI of 1 = ASCII
(6)	5-7			Reserved
				Normal-Flow Send and Receive Mode Selection:
7	0-1	B'00'		Not valid
		B'01'		Not valid
		B'10'		Half-duplex flip-flop
				Bracketed half-duplex flip-flop sessions require: Byte 4, bit 2 or 3 must be set to 1 (primary half-session chaining response). Byte 4, bit 7, or byte 5, bit 7, or both must be set to 1 (send EB). Byte 5, bit 2 or 3 must be set to 1 (secondary half-session chaining response). Byte 6, bit 2 must be set to 1 (bracketed session). Byte 7, bit 7 must be set to 0 (reset state is Receive for primary and Send for secondary).
		B'11'		Reserved
				Recovery Responsibility:
(7)	2	B'0'		The contention loser is responsible for any error recovery attempts for the session.
		B'1'		Not valid
				Contention Winner/Loser:
(7)	3	B'0'		Secondary LU is contention winner and primary LU is contention loser.
		B'1'		Not valid
(7)	4-6			Reserved
				Half-Duplex Flip-Flop (HDX-FF) Reset States:
(7)	7			Reserved

TS Usage

(8-13)				See bytes 8 through 13 below.
				Staging Indicator and Pacing Count for Secondary CPMGR (connection point manager) to Primary CPMGR normal flow:
8				Reserved
9	0-1			Reserved
(9)	2-7	B'nnnnnn'		Secondary CPMGR receive pacing count. A value of 000000 causes the boundary function to substitute the value set by a system definition pacing parameter (if the system definition includes

Byte	Bits	Value	Opt	Discussion
				such a parameter) before it sends the BIND RU on to the secondary LU. A value of 000000 received at the secondary LU is interpreted to mean no pacing of requests flowing to the secondary LU.
10				Reserved
11		X'ab'	X	Maximum RU size sent on the normal flow by the primary half-session. This value represents the largest RU that can be sent. It is expressed as a mantissa and an exponent value of 2 by which the mantissa is multiplied.
		X'85'		When the mantissa is 8 and the exponent is 5 (X'85'), the RU size is 256 bytes (8×2^5).
		B'0n...'		When bit 0 is set to zero, no maximum is specified and the remaining bits 1-7 are ignored. When bit 0 is set to one, the byte is interpreted as X'ab'. See Appendix B for these values.
				Staging Indicator and Pacing Count for Primary CPMGR (connection point manager) to Secondary CPMGR normal flow:
12	0	B'0' B'1'		Pacing in this direction occurs in two stages. Pacing in this direction occurs in one stage.
				Note: The meanings of 0 and 1 are reversed from the staging indicator for secondary CPMGR to primary CPMGR (see byte 8).
(12)	1			Reserved
(12)	2-7	B'nnnnnn'		Primary CPMGR send pacing count. A value of 000000 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 000000, it means no pacing of requests that flow from the primary. For single-stage pacing in the primary-to-secondary direction, this field is redundant with, and will indicate the same value as, the secondary CPMGR receive pacing count (see byte 9, bits 2-7 above).
13				Reserved
PS Profile				
(14-25)				See bytes 14 through 25 below.
14	0	B'0' B'1'		PS usage field format is the basic format. Reserved
(14)	1-7	B'0000011'		LU-LU session type 3
PS Usage				
15	0	B'0' B'1'	 X	Query not supported Query supported
	1-7			Reserved
16-19				Reserved
				Buffer-Size Control:

Byte	Bits	Value	Opt	Discussion
				Use byte 24 to specify a 480- or 1920-character print buffer size for the SNA 3270 data stream. Use bytes 20-24 to specify print buffer sizes for all uses of the extended SNA 3270 data stream (see Note 1). Buffer sizes are specified by row and column to be consistent with buffer sizes in LU type 2. The row and column values have no significance in LU type 3 other than their product defines the buffer's wrap point.
20	0-7		X	Default buffer size: rows. This is the number of rows in the buffer, stated in binary. Permissible values are from 1 to 255.
21	0-7		X	Default buffer size: columns. This is the number of columns in the buffer, stated in binary. Permissible values are from 1 to 255.
22	0-7		X	Alternate buffer size: rows. This is the number of rows in the buffer, stated in binary. Permissible values are from 1 to 255.
23	0-7		X	Alternate buffer size: columns. This is the number of columns in the buffer, stated in binary. Permissible values are from 1 to 255.
24	0			Reserved
(24)	1-7			Buffer size (see Note 1)
		0000000		Not specified
		0000001	X	12 rows by 40 columns (480 characters)(see Note 2)
		0000010	X	24 rows by 80 columns (1920 characters)(see Note 2)
		1111110	X	Static size specified by bytes 20 and 21 (see Note 3).
		1111111	X	Default size specified by bytes 20 and 21, and alternate size specified by bytes 22 and 23 (see Note 4).
				All other values reserved.
25				Reserved

Cryptography Options

26	0-1			Private cryptography options:
		B'00'	X	No private cryptography supported.
		B'01'	X	Private cryptography supported. The session cryptography key and cryptography protocols are privately supplied by the end user.
		B'10'		Reserved
		B'11'		Reserved
(26)	2-3			Session-level cryptography options:
		B'00'	X	No session-level cryptography supported.
		B'01'	X	Session-level selective cryptography supported. All cryptography key management is supported by the SSCP and LU; exchange (using +RSP to BIND) and verification (using CRV) of the cryptography session-seed value are supported by the LUs for the session. All FM data requests with EDI (enciphered data indicator) on are enciphered/deciphered.
		B'10'		Reserved
		B'11'	X	Session-level mandatory cryptography supported; same as session-level selective cryptography except <u>all</u> FM data requests are enciphered/deciphered.
(26)	4-7			Session-level cryptography options field length:
		X'0'	X	No session-level cryptography specified. The cryptography option fields (bytes 27-k) are omitted.

Byte	Bits	Value	Opt	Discussion
		X'9'	X	Session-level cryptography specified. Additional options follow in the next nine bytes.
27	0-1			Session cryptography key encipherment mode:
		B'00'	X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key using a seed value of zero (only value defined).
(27)	2-4			Reserved
(27)	5-7			Cryptography cipher method:
		B'000'	X	Block chaining with seed and cipher text feedback, using the data encryption standard (DES) algorithm (only value defined).
28-k			X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key; an eight-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FM data requests.

Names and End User Data

k+1			Length of BIND sender's (or PLU) name in field k+2-m.
	X'00'		Field k+2-m does not exist.
	X'nn'	X	Field k+2-m contains the number of bytes specified (cannot exceed X'08').
k+2-m		X	BIND sender's (or PLU) name.
m+1			Length of user data in field m+2-n.
	X'00'		Field m+2-n does not exist.
	X'nn'	X	Field m+2-n contains the number of bytes specified.
m+2-n		X	User data.
m+2		X	User data key.
	X'00'	X	Structured subfields follow.
	-X'00'	X	First byte of unstructured user data.
			Note: The unstructured user data continues in m+3 and goes to byte n. For information on structured subfields, see the BIND information in SNA Format and Protocol Reference Manual: Architecture Logic.
n+1			Length of user request correlation field.
	X'00'		Field n+2-p does not exist.
	X'nn'	X	Field n+2-p contains the number of bytes specified.
n+2-p		X	User request correlation field.
p+1			Length of BIND receiver's (or SLU) name in field p+2-r.
	X'00'		Field p+2-r does not exist.
	X'nn'	X	Field p+2-r contains the number of bytes specified (cannot exceed X'08').
p+2-r		X	BIND receiver's (or SLU) name.

Notes

The following notes apply to the BIND SESSION request:

Note 1: Buffer sizes can be specified using byte 24, or byte 24 can refer to the buffer sizes defined in bytes 20-21 and, as required, bytes 22-23.

If the SNA 3270 data stream is used, byte 24, bits 1-7, may specify only B'0000000', B'0000001', or B'0000010'. The B'0000000' value permits either a 480 or 1920 screen size. If any functions above the base set of the SNA 3270 data stream are used, byte 24 may specify any of the defined values.

If the data being loaded in the buffer exceeds the designated buffer limit, the remainder of the data is wrapped, that is, loaded starting at the first buffer position. With one exception, the buffer wraps at the value(s) specified in BIND regardless of the actual size of the buffer. The exception is the byte 24, bits 1-7, value of B'0000000'; the buffer wraps at the end of the actual buffer size (480 or 1920).

Note 2: The buffer size is fixed for the duration of the session when this mode is selected. When in this mode, an Erase/Write Alternate command may be rejected with negative response X'1003'.

Note 3: If byte 24, bits 1-7, contain B'1111110', the buffer size specified in bytes 20-21 is static. Bytes 22-23 are not used. When in this mode, an Erase/Write Alternate command may be rejected with negative response X'1003'.

Valid row and column parameters for default bytes 20-21 can be found in implementation literature. The product of the number of rows (byte 20) and columns (byte 21) must not exceed 4096 for the base set of the SNA 3270 data stream (upper limit of order stream addressing), and 65536 when the base set is extended with additional functions.

Note 4: An alternate buffer size may be specified if byte 24, bits 1-7, contain B'1111111' (dynamic buffer size switching within a session).

The BIND operation establishes both default (bytes 20-21) and alternate (bytes 22-23) buffer sizes to be used within a session. It initializes the unit by setting it in the default buffer size mode.

An Erase/Write Alternate command changes the buffer size from default to alternate. An Erase/Write command or depression of the CLEAR key changes the buffer size from alternate to default.

Valid row and column parameters for default bytes 20-21 and alternate bytes 22-23 can be found in implementation literature. The product of the number of rows and columns must not exceed 4096 for the base set of the SNA 3270 data stream (upper limit of order stream addressing), and 65536 when the base set is extended with additional functions.

Type 4 LU-LU sessions are used for data communications between two terminals or between application programs and single- or multiple-device terminals. Type 4 sessions are similar to type 1 sessions, particularly for data processing applications. In addition, word processing applications can execute using type 4 sessions.

LU_T4 sessions may occur as peer-to-peer sessions or as PLU/SLU sessions. In peer-to-peer sessions, the relationship between half-sessions is symmetrical; that is, the LU at either end of the session is of equal standing (a peer) with the LU at the other end. In PLU/SLU sessions, the relationship is asymmetrical, and the primary LU controls the session.

In peer-to-peer sessions, the operational protocols are symmetrical. This implies, among other things, that an LU must recover from its own errors occurring in data, since there is no primary LU to do so. It further implies that an LU is never the master of the other LU in the session, and that one LU has no recovery responsibility for the other LU's data.

In the peer-to-peer environment, the terms primary LU and secondary LU no longer retain their traditional meaning. For convenience and to preserve the analogy to the terminology used by other LU types, the LU that sends the BIND SESSION request will be called the primary LU, and the LU that receives BIND SESSION will be called the secondary LU. However, these terms are used only in the description of the BIND and UNBIND procedures, since, other than during these procedures, either LU in the session may start any sequence of activity.

In addition to peer-to-peer sessions, LU_T4 implementations are capable of operating in a network controlled by an SSCP using PLU/SLU sessions. To do so, the session partners must conform to the SSCP network and session activation and deactivation procedures. After the activation procedures are accomplished, the partners either:

Consider the other LU in the session to be its peer, or

Operate in a nonpeer environment (as a PLU or SLU in a PLU/SLU session). This mode of operation enables LU_T4 products to be supported by current PLU implementations.

The proper selection of BIND parameters enables an LU_T4 product to select a peer-peer or PLU-SLU session.

LU_T4 is used to manage input/output devices, such as printers, card readers, card punches, and various storage devices. An operator console is often present for further input/output in an interactive mode or for diagnostic messages.

In addition, some LU_T4 implementations perform various data management functions, such as storing data for later retrieval and modification by the sender or for later distribution to multiple destinations. Application programs and application subsystems may use type 4 sessions in order to communicate with the other type 4 implementations in their network.

Product implementors are not required to implement all options of the LU_T4 protocol. The options implemented by a particular product are specified in the BIND that it sends to start a session or in the BIND response that it sends (for a negotiable BIND), or in the mode that it selects for Initiate (to invoke a nonnegotiable BIND).

BIND SESSION REQUEST

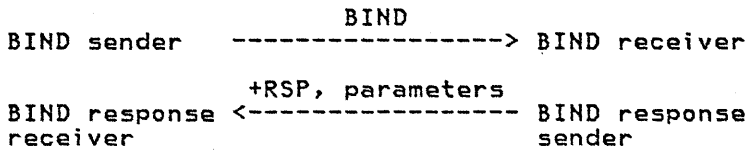
There are two types of BIND -- nonnegotiable and negotiable. The former is used primarily with products using PLU-SLU sessions, while the latter is used with peer-to-peer sessions.

The nonnegotiable BIND has the BIND sender establishing the session parameters without compromise on potentially negotiable characteristics such as maximum RU size and pacing counts. This requires the BIND sender to store the characteristics of each different type of station supported in the SNA communications network.

The negotiable BIND has the BIND sender proposing a set of session parameters, and the BIND receiver adapting where possible and returning a set of response parameters, which together establish the negotiated limits for exchanges in both directions. The sender of a negotiable BIND, upon receipt of the BIND response, checks the returned parameters and accepts the session as negotiated or sends UNBIND to the session partner.

NEGOTIABLE BIND REQUESTS AND RESPONSES

Logical units (LUs) negotiate the protocols they wish to use for LU_T4 sessions by sending to each other a negotiable BIND request and an appropriate response. The first LU to send a BIND request becomes the BIND sender and the other LU becomes the BIND receiver. The BIND sender includes his receive capability and his proposal for his send protocol in his BIND request. The BIND receiver examines the BIND request and either accepts the session parameters by returning a positive response with the same session parameters, or negotiates the session parameters by returning a positive response with different session parameters. This process is illustrated below:



Although other capabilities apply (such as security and user information), the key to establishing a session is the exchange of receive capabilities. Once an LU-LU session is established, an LU must not exceed the receive capability of the session partner -- so the BIND process must ensure that there is a send-receive match. A negotiable BIND may be rejected only on the basis of the session partner's receive capability; that is, a negative response may be sent only when the BIND receiver determines that a viable session cannot be conducted because it cannot send what the BIND sender can receive.

When the BIND receiver sends a positive response but changes the session parameters, the BIND sender may send UNBIND only on the basis of the session partner's receive capability; that is, it cannot send what the BIND receiver can receive.

HALF-SESSION CHARACTERISTICS

FM, TS, AND PS PROFILES

LU_T4 uses FM profile 7 and TS profile 7. The SNA requests that are allowed with each profile are found in SNA Reference Summary.

LU_T4 uses a PS profile that allows:

- Function management headers. LU_T4 implementations can run with or without FM headers. FM header functions are discussed in Part 2 Chapter 4; considerations and restrictions that are unique to LU_T4 are discussed under "FM Header Processing" on page 78.
- Data streams. Functions listed in Part 2 Chapter 1 are allowed except where restricted under "Data Stream Profiles" on page 84.

TC (TRANSMISSION CONTROL) CONSIDERATIONS

Sequence numbers are not used with TS profile 7. Transmission control services, such as pacing and cryptography, are summarized in SNA Concepts and Products and defined in SNA Format and Protocol Reference Manual: Architecture Logic.

DFC (DATA FLOW CONTROL) CONSIDERATIONS

The send-receive mode used by LU_T4 is half-duplex flip-flop with no contention error processing state.

Data flow control services, such as chaining, brackets, and send/receive modes, are summarized in SNA Concepts and Products and defined in SNA Format and Protocol Reference Manual: Architecture Logic. The following paragraphs give additional considerations.

SEND-RECEIVE MODES

LU_T4 implementations must use half-duplex flip-flop as the send-receive mode. Half-duplex contention and full duplex are not allowed.

BRACKET PROTOCOLS

LU_T4 implementations may choose to use brackets to delimit a unit of work. Either half-session may be the bracket first speaker (contention winner) as defined in BIND SESSION byte 7, bit 3.

FM profile 7 does not allow the BID request. To avoid confusion during a bracket contention condition (both session partners attempting to begin a bracket at the same time), the contention loser should send the RU that begins the bracket BB,RQD (request definite response).

Destination selection cannot span brackets. Multiple destinations may be selected within a bracket.

CHANGE-DIRECTION (CD) MANAGEMENT

Use of CD Indicator (CDI): The sending of the CDI should occur under the following circumstances:

- An LU has reached an end-of-send data-transition condition (including any keyboarded data on the LU flow).
- An LU has received SIG X'0001xxxx' (request change direction). The procedures that must be followed when returning the CDI are discussed below under "Signal Codes."
- An LU has been session-activated in the send state, but has no data traffic to send.
- At session activation, the BIND sender receives a BIND response (or a CINIT in a PU type 5 node) with an indication that the BIND receiver must send data first.

The sending of the CDI must occur under the following circumstances:

- When there is no data to send and the BIND parameters specify that the session partner must receive CD on every EDS (end of destination

selection). This includes the implied EDS at the end of a transmission to the console.

- When the LU receives CD and the LU doesn't have the capability to send data (for example, a printer). In this case, the CDI is returned as part of LUSTAT X'0002'.

The CDI may be returned as part of LUSTAT X'0007'.

An LU should send available FM data, if any, when the CDI is received. If there is no FM data pending, the LU must wait for additional FM data to send or for one of the conditions described above.

Use of SIG (Signal): An LU that is capable of sending FM data must be capable of sending SIG X'00010000' (request change direction), under operator control or automatically, when in receive state with FM data to send.

The sender of SIG X'00010000' must return the CDI when it has no more data to send.

The receiver of SIG X'00010000' must not send SIG X'00010000' after sending the CDI until the end-chain indicator (ECI) has been returned. This prevents a signal ping/pong condition from occurring with no data flowing.

Data Direction Resolution: The BIND parameters indicate whether the BIND sender or the BIND receiver will send data first at the beginning of the session (see byte 24, bit 2). The BIND sender is always initialized to the send state. If this bit indicates that the BIND receiver sends data first (value=1), then the BIND sender always begins with CDI or, when applicable, the EBI bit on a null RU.

Signal Codes: The SIG request permits an expedited signal to be sent through the network, regardless of the status of the normal flows. It carries 4 bytes of information set by the sending LU. The signal information has meaning only to the LU and is simply passed by the network. Formats are shown and fields are described below.

RU Formats:

Request	byte 0	= X'C9'
	bytes 1-2	= Signal field
	bytes 3-4	= Signal extension field
Response	byte 0	= X'C9'

Sig. Fld.	Sig. Extn. Fld.	Description
0000		No-op: Signal extension field may have user-defined information.
	0001	Signal operator: The sending LU wants to notify the operator at the receiving LU to take some type of action. Notification can take the form of ringing a bell, turning on a light, and so on.
	0002	Component now available.
	0005	Signal operator reset: The sending LU wants to notify the operator at the receiving LU to reset the function or action taken as a result of receiving SIG with a Signal Extension field of X'0001'.
	0006	Component stopped as a result of service representative or operator action. No data lost.
	0007	Start component. Used by LU receiving intervention required (SIG X'0003uuuu') to restart the flow.
		All other codes are reserved.

0001 Request change direction: The sending LU is in the receive state and wants to be in the send state. An LU can send data on the normal flow only after it has received an RU on the normal flow with the change direction indicator (CDI) set on.

0000 Send CDI immediately: The receiver of request change direction, if it is the half-session without recovery responsibility, is permitted to clear the active destination from its send stack, by abnormal termination if necessary, in conjunction with sending the CDI. Normal clearing of the stack comprises completing the current chain and deselecting with the END (EDS) FMH-1. Abnormal clearing of the stack comprises completing the current chain with CANCEL, deselecting with the END-ABORT (ADS) FMH-1 (or sending LUSTAT (ABORT) if the default destination has been selected). In addition, the half-session with recovery responsibility may also send a single chain console message in conjunction with sending CDI after clearing its stack.

If received by the half-session with recovery responsibility, normal and abnormal termination remain the same, but the half-session may, after clearing its stack, send a single-chain console message with the CDI.

0001 Send CDI as soon as possible: The receiving LU should send CDI but not by prematurely terminating the work in progress.

All other codes are reserved.

0002 Request assistance: The signal extension field is used to indicate what assistance is desired.

uuuu User defines what type of assistance is needed.

0003 Intervention required, no data lost: The signal extension field is used to indicate what assistance is required.

uuuu User defines what type of assistance is needed.

All other signal field and signal extension field codes are reserved.

If the receiver does not recognize the code in the signal field, or if the receiver recognizes the code but cannot adequately perform the function required by the signal field or signal extension field, then the receiver may indicate this inability to the sender by negatively responding to SIG with sense code X'1003' (function not supported).

Use of LUSTAT X'0002': An LU that does not have the capability to send FM data (an output-only component) sends LUSTAT X'0002' whenever it receives a change direction indicator (CDI). The CDI is set on in the returned LUSTAT command so that the LU remains in the receive state.

When an LU receives LUSTAT X'0002', if it is able and ready to send data, it sends it; if it is not ready to send data it either sends UNBIND or RSHUTD, or it remains in the send state until it is ready to send more data.

Use of LUSTAT X'0007': An LU that has the capability to send FM data, sends LUSTAT X'0007' when it no longer has such data to send. It is sent at the time that the next FM data would have been sent (when LUSTAT X'0007' sender is in HDX-FF send or contention state). There is no forced turnaround of the HDX-FF state (that is, Signal RCD) in order to send LUSTAT X'0007'.

The implementation must define when the LU has no more data to send. For example, an attended terminal might pause to give the operator the opportunity to enter a new send job after the last one had been transmitted.

When an LU receives LUSTAT X'0007', if it is able and ready to send data it sends it; if it is not able to send data, it sends either UNBIND or RSHUTD; if it is able to send data but currently has no data to send, it may wait for data to send or receipt of SIG (RCD), or it may return the opportunity to the session partner (CD or EB). The following FSM and flow diagram provide a send check to prevent unlimited exchange of LUSTAT X'0007'.

Specifying the Change-Direction Mode Using BIND: A parameter in the BIND SESSION request specifies if the change direction indicator (CDI) is to be sent on every chain that carries an end of destination selection (EDS or BEDS) FMH-1.

There are two ways that this is done. The first way is to send more than one EDS or BEDS without a change in direction. The second way is to set the CDI on in the chain that contains EDS or BEDS.

In the first way, the LU can begin and end many destinations and then end with the CDI set on. The receiver of data then separates the data sets for processing. This is done in a manner that satisfies the requirements of the specific receiver of data in the session.

In the second way, the LU can begin and end only one destination per chain. The LU must send CD in the chain that contains EDS or BEDS. This includes sending CD with the implied EDS of a headerless transmission to the console. If brackets are used, the LU must not arbitrarily send multiple data sets in a bracket since the various receivers of data with which it communicates may use a bracket for varying purposes, some of which require that the receiver of data control the content of the bracket.

PS (PRESENTATION SERVICES) CONSIDERATIONS

DATA STREAMS

LU_T4 half-sessions use the SNA character string (SCS). The sets of controls that are valid are discussed under "Data Stream Profiles" on page 84.

The data streams that are valid on a session are specified in BIND SESSION bytes 15 through 22. The data stream to be sent to a specific destination is specified in the FMH-1 DSP select field. Valid settings for this field are discussed under "DSP Select Field of FMH-1" on page 82.

FM HEADER PROCESSING

Use of FMH-1 in LU T4

The FMH-1 serves four primary functions: defining the beginning and ending of jobs, specifying the data stream format, specifying the output medium required for the job, and specifying points of suspension and resumption of jobs.

There are restrictions that LU_T4 places on the use of FMH-1 in addition to the general rules imposed by Part 2 Chapter 4:

1. Only the LU that begins a selection may suspend, resume, end, or abort it. An end bracket indicator sent by either session partner ends an active destination selection.

2. A destination begun with begin destination selection (BDS) may be terminated either by end destination selection (EDS) or end-abort destination selection (ADS). In either case, a positive response indicates the ending of the destination selection.

   Termination with EDS indicates normal completion of the destination selection to both LUs.

   Termination with ADS indicates that the destination selection has been prematurely terminated. The reason for termination and mechanisms for recovery are not addressed; however, responsibility for recovery action remains with the applicable LU (BIND sender or RU sender as selected in BIND).

3. FM data sent without an FMH-1 is routed to the console when FMH usage has been indicated in BIND and when there is no active destination. If no output device on the LU has been designated as a console, the RU is rejected (sense data X'0801').

4. All RU chains that carry an FMH-1 must be sent RQD (request definite response), regardless of the presence or absence of CDI.

5. All FMH-1s must be in the first or only RU of a chain. Except for BEDS, FMH-1s may not be followed by data. All FMH-1s may be followed by concatenated FMHs; they may not be preceded by either FMHs or data.

6. BEDS must be in the first or only request of a chain. It may be followed by concatenated FMHs or data, within the RU or in subsequent RUs, but it may not be preceded by either FMHs or data.

7. LU_T4 uses FMH-1 to control the beginning, ending, suspension, and resumption of destinations. The following subsets of FMH-1 support are defined and selected via BIND:

ONE-LEVEL STACK: An active destination may not be suspended in the same direction. An active destination may be implicitly suspended by allowing FM data (with or without an FMH) to flow in the opposite direction. If the RU carries an FMH selecting any destination other than the console, the data may be negatively responded to (sense code X'0801').

The restriction of allowing only console data flowing in the opposite direction to suspend an active destination is an LU_T4 restriction, not a stack manager restriction.

Example:

1. BC, EC, FMH (BDS, Print 1)
   ---------->

2. BC, EC, FMH (SDS)
   ----------> INVALID

3. SIG (RCD)
   <----------

4. BC, EC, FMH (EDS), CD
   ---------->

5. BC, EC, data, CD
   <----------

6. BC, EC, data
   ---------->

Notes: The numbers on the following notes correspond to the numbers in the example above.

1. Printer is selected.

2. This is invalid since only one level of destination is allowed in a given direction and the suspend destination selection implies a second selection.

3. Request for CDI.

4. Flow turnaround. The sending half-session must terminate the destination before sending the CDI.

5. This may be with or without an explicit FM header. If this chain contains an explicit FM header that selects any destination other than the console, the chain may be rejected (sense code X'0801').

6. Data to the printer continues.

TWO-LEVEL STACK: Two levels of selection are allowed. The highest (last) selection, however, must be the default header for console.

Example 1:

1. BC, EC, FMH (BDS, Print 1)
   ---------->
2. BC, EC, FMH (SDS)
   ---------->
3. BC, EC, FMH (BDS, Card 2)
   ----------> INVALID
4. BC, EC, data
   ---------->
5. BC, EC, data
   ---------->
6. BC, EC, FMH (RDS)
   ---------->
7. BC, EC, data
   ---------->

Notes for Example 1:

1. Printer is selected.
2. Printer is suspended.
3. This selection is invalid since highest (second) level of interruption must be default header.
4. Since the printer is suspended, this headerless data is directed to default (console).
5. See Note 4.
6. Printer is resumed.
7. Data to printer.

Example 2:

1. BC, EC, FMH (BDS, Print 1)
   ---------->
2. SIG (RCD)
   <----------
3. BC, EC, CD
   ---------->
4. BC, EC, FMH (BDS, Card 2)
   <----------
5. BC, EC, data
   <----------
6. BC, EC, data
   <----------
7. BC, EC, FMH (SDS)
   <----------
8. BC, EC, data
   <----------
9. BC, EC, FMH (RDS)
   <----------
10. BC, EC, data
    <----------
11. BC, EC, CD, FMH (EDS)
    <----------
12. BC, EC, data
    ---------->

Notes for Example 2:

1. Printer is selected.
2. Flow turnaround requested.
3. Flow turnaround granted.
4. First level of selection (card) from opposite direction.
5. Card data.
6. Card data.
7. Card suspension.
8. Data to default header for console. Second level of interruption must be console only (via default).
9. Card resumption.
10. Data to card.
11. Card ended and flow turned around.
12. Data to printer continued.

THREE-LEVEL STACK: Three levels of interruption are allowed. The highest (last) selection must be the console (explicit or default header).

COMPRESSION AND COMPACTION: Compression of FM data using SCBs and compaction of FM data using SCBs are optionally supported by implementations of LU_T4. The use of either compression or compaction requires FMH-1 for beginning and ending the compressed or compacted data. Either LU in the session may establish or change the compaction table to be used by sending a Compaction Table FMH-3. The rules for FMH-3 are discussed below under "Use of FMH-3 in LU_T4."

DSP SELECT FIELD OF FMH-1: The DSP Select field of the FMH-1 is coded as follows for LU_T4:

Byte 3

Bits 4-7	DSP Select
X'0'	Default DSP
X'1'	Base DSP
X'2'	General DSP
X'3'	Job DSP
X'4'	WP raw-form DSP
X'5'	WP exchange diskette

All other values are reserved.

Use of FMH-2 in LU T4

The previous section showed how the FMH-1 was used to define the beginning and ending of jobs, specify the data stream format, specify the output medium required for the job, and specify points of suspension and resumption of jobs. FMH-2 provides processing parameters for FM data. These parameters include presentation services specifications, horizontal tab stops, vertical line spacing, output type style, and format-control information.

Use of FMH-3 in LU T4

Either LU in the session may establish or change the compaction table by sending a compaction table FMH-3. The rules associated with this header are:

1. Until one of the LUs sends a compaction table header, compacted data is not allowed.

2. A compaction table becomes effective when the positive response to the compaction table header is issued and remains effective until another becomes effective or until the end of the session.

3. The effective compaction table applies to all FM data sent in either direction during its period of effectiveness. This includes data sent from either LU, and it includes data sent to destinations that were suspended when the compaction table became effective but are resumed during its period of effectiveness.

4. Since LU_T4 supports only HDX-FF in brackets, there is no race condition between the two LUs as they try to establish compaction tables.

5. LU_T4 supports the QUERY FOR COMPACTION TABLE FMH-3, requesting that the other LU establish the compaction table.

6. The first RU of a chain must start with a string control byte (SCB). The SCB follows the FMH-2, if present, unless the next byte is another FMH-2. The SCB and count may reflect spanning across RUs after the first SCB. The last SCB in the last RU points to one byte beyond the length of the RU except for repeated prime character (see Part 2 Chapter 5).

GENERAL CONSIDERATIONS

INTERVENTION-REQUIRED CONDITIONS

When an intervention-required condition occurs, such as a printer being out of paper, the LU may handle the condition using the following options:

- Time out. Start a timer of between 0 and 10 minutes' duration. During this time, manually correct the condition. If the condition cannot be corrected, the LU sends a negative response with the sense code defining the original problem.

  If there is a requirement to notify the session partner of an intervention-required condition early, option 3 may be used in conjunction with option 1.

- Send signal and timeout. Send SIG X'0003uuuu' (intervention required - no data loss), which permits an expedited signal to be sent through the network, regardless of the status of the normal flow. Start a timer of between 0 and 10 minutes' duration. During this time, manually correct the condition. If the condition can be corrected, send SIG X'00000002' (component now available). If the condition cannot be corrected by the end of the time period, then send a negative response with the sense code defining the original problem.

  In a peer-to-peer environment, the need for a timer is optional. In a nonpeer environment, the timer is mandatory.

- Send negative response X'0802' and LUSTAT X'0001'. This response is precluded in a peer-to-peer environment; in a nonpeer environment, it is allowed.

  The half-session responsible for recovery, upon receipt of a negative response, enters (S, ¬R) state and must send to continue ERP activity.

ATTENDED AND UNATTENDED MODES OF OPERATION

An attended session has an operator available to satisfy intervention-required conditions and otherwise respond to requests for operator action. When operating in the unattended mode, it must be assumed that an operator is not available to respond to requests, and the effects of this mode are described below.

- The session partners need not display console messages to an operator on line, but may file them for future operator interpretation. This filing and future action by the LU is not architected.

- If CD is sent to the LU and the LU does not have the capability to send FM data, it must send LUSTAT X'0002rrrr'.

- If CD is sent to the LU, and the LU has the capability to send but has no FM data to send, it must send one of the following:

  LUSTAT X'0002rrrr' or X'0007rrrr' (no data to send)
  RU without data but with CDI set on
  UNBIND
  RSHUTD

- Since intervention required (SIG X'0003uuuu') cannot be executed by the programmed operator, a request not executable response (negative response X'081Cuuuu') is issued by the LU.

- Card data may or may not be written to disk on receipt for later punching. (This is also true of attended operation.) If not written to a disk, a

request not executable response (X'081Cuuuu') is issued when the card receipt processor runs out of cards.

- The combination of unattended operation and request not executable (receipt of negative response) is not architected and the receiver may take any action required.

An LU may send the following status codes in LUSTAT if the BIND parameters permit switching between attended and unattended modes:

0003 - Entering attended mode of operation
0004 - Entering unattended mode of operation

DATA STREAM PROFILES

INTRODUCTION

Data stream profiles (DSPs) define the presentation space attributes of FM data. The repertoire of DSPs that a half-session is capable of receiving is indicated in the BIND images exchanged at session initiation (see BIND SESSION bytes 15-16 and 19-20). The repertoire of DSPs indicated in BIND implies that the attributes of each of these data streams are fully supported by the half-session when it receives FM data. As a receiver, the half-session processes these data streams as defined in Part 2 Chapter 1 and in the product libraries of products that have implemented LU_T4 half-sessions.

FM Headers: During the life of the session, FM data exchanges are controlled by FM headers. DSP selection for a particular FM data exchange is indicated by the DSP field (byte 3) when the destination is selected using FMH-1.

During a session, when a data stream selection has been made with a BEGIN FMH-1 (DSSEL field = B'010') and accepted by the receiver, the attributes specified in the FMH-1 are assumed on all subsequent data exchanges for that destination selection, until the END or END/ABORT FMH-1 or end bracket is sent.

The DSP selection is made by the sender of the BEGIN or BEGIN/END FMH-1 to indicate the desired attributes for processing the associated FM data. The receiver of the FMH-1, by accepting the DSP selected, processes the FM data that follows exactly as defined in the DSP definition. The receiver may also accept data whose attributes do not conform to the selected DSP. If the receiver can process the data as indicated by these additional attributes, it should be done; if not, then either the standard default action should be invoked or an error response returned to the sender.

When a DSP selection is made by the sender that does not conform to the repertoire agreed to at BIND, the receiver may reject the data stream (sense code X'1008400E'), or the receiver may attempt to process the data stream using defaults where unable to execute as indicated.

All presentation space controls of the SCS data stream are in effect upon receipt, from BDS to the corresponding EDS. The EDS for the selected destination resets all SCS controls to their default values. ADS and other abnormal terminations of the selected destination also reset the SCS controls to their default values. This implies that the controls are in effect as long as the destination is on the stack.

In summary, data stream selection is performed by the sender of data and enforcement of the selected data stream is performed by the receiver of data. The receiver can also exercise an option on how to handle attributes that do not conform to the selected data stream.

Base Data Stream Profile: The base DSP is the minimum SCS control and graphic character set and is mandatory for all products to assure a minimum level of compatibility. The graphic character set of the base DSP, with no SCS

control characters allowed, is used in the user field of BIND, in any exchange of security information, in any data set name field of an FMH-1 and in store-and-forward addressing for document distribution.

The base DSP is the required character set to be used for sending and receiving any console or supervisory services messages, although a less restrictive character set may be used between products if successful use has been assured by the BIND parameters.

Two SCS controls can appear in the Base DSP: New Line (NL) and Form Feed (FF). Neither has a default. FF is meaningless in continuous-roll paper-fed printers and is treated as NL.

The graphic character set for base DSP is the 48-character set which includes space, the uppercase alphabet, numerals, and 12 special characters.

Use of the base DSP in user fields, FM headers, and in supervisory services or console messages which may be interpreted by machine must conform to the base DSP control and character set. However, it is recommended that coordinating messages exchanged between operators should not be rejected for failure to comply with this character set restriction; the receiving operator may clarify any ambiguity by a continuing interactive exchange. Thus, any graphic character that is not understood should be defaulted.

The sender should not assume that the receiver makes any distinction between uppercase and lowercase alphabetic characters, either in printing or in comparing for correct security, data set name, etc. Any control character that is not understood should be defaulted.

MEDIA

Console: The console in LU_T4 is an SCS device. The set of SCS control characters accepted by a given console is specified in BIND (bytes 17 and 22) for LU-LU sessions and by CINIT (Control Initiate) for the SSCP-PLU session). CINIT is defined in SNA Format and Protocol Reference Manual: Architecture Logic.

There are several sets of control characters that are supported by type 4 consoles -- a base set (base data stream profile) supported by all consoles and extension sets that are optionally and independently supported.

Document: The only operational difference between printer and console that affects the session is that the console presentation surface is shared between the SSCP-LU session and the LU-LU sessions. (See "Console" above.) The printer uses only an LU-LU session.

Card: The card data stream is an SCS data stream. The ERCL (exchange record length) field in the BEGIN or BEGIN/END FMH-1 specifies the point at which an auto-IRS (automatic interrecord separator) is applied. If zero is specified, the point at which the auto-IRS is applied is defined by prior agreement between the session partners.

A chain consists of one or more card images. LU_T4 allows a card to begin in one RU and be continued in the next RU, as long as the RUs are in the same chain. These RUs are known as card-spanning RUs. Cards may not be spanned across chains. The sending LU must enforce this restriction.

Card images within a chain are normally delimited by IRS. There are, however, two other implicit delimiters. The maximum presentation position (MPP), as specified in the BEGIN or BEGIN/END FMH-1 or understood by the session partners, is optionally applicable. The end of the chain is required and always applies. If either of these limits is reached before an IRS is found, the current card image is complete. In the case of MPP having been reached, if an IRS is found at MPP+1, it is associated with the card image just completed. If it is not an IRS, it is associated with the next card image. This is known as automatic IRS.

Transparency may also be used for card data. There is no restriction on the appearance of a transparent string within the card data. The TRN escape sequence (TRN and count) may appear at any point with respect to card image

boundaries, and the count may run out at any point. The count field may describe a number of bytes that constitute all or part of one or more card images. The auto-IRS function, if applicable, is still active even if the transparency count has not run out. The transparency count, like card images, may span RUs but not chains.

Exchange: The sender of data, for application oriented reasons, may require that the receiver store the data on a removable medium that can be transported to another machine for processing. When this requirement exists, the sender selects exchange medium in the FMH-1 that begins the job. The FMH-1 is also used to select the data organization on the medium. The architected exchange media place some preformatting requirements on the data sender and some output medium and formatting requirements on the receiver, which are discussed below.

Basic Exchange: The sender must ensure that the data meets the EBCDIC or ASCII standard (as selected in BIND and the RH) and that the value in the ERCL (exchange record length) field in the FMH-1 is less than or equal to 128. The receiver creates from this data a data set which it stores on an exchange medium (of any physical type) that has been formatted according to Basic Exchange standards. The algorithm for doing so is:

1. The entire stream of data in a chain is considered as a unit.
2. The data is scanned until the enchange record length is reached.
3. The scanned data becomes the content of a logical record in the data set.
4. Scanning continues with the next byte of the data stream and steps 2 and 3 are repeated.

Implied Media Support for Data Processing Media: The following DSP support indicated in the BIND implies media support in FMH-1:

1. At least job DSP supported for printer (byte 15, 19, or both) implies document media support.
2. At least base DSP supported for console (byte 17, 21, or both) implies console media support.
3. Card DSP support (byte 16, 20, or both) implies card media support.
4. Basic exchange DSP support (byte 16, 20, or both) implies exchange medium support.
5. Any word processing DSP support implies WP media 1, 2, 3, and 4.

DATA PROCESSING DATA STREAM PROFILES

All profiles described below apply to data processing data stream profiles and not word processing data stream profiles.

The profiles are:

1. Base data stream (see "Base Data Stream Profile" above)

Control character	Abbreviation
Form feed	FF
New line	NL

2. General data stream. The general data stream contains the base data stream, plus:

Control character	Abbreviation
Carriage return	CR
Line feed	LF
Bell	BEL
Presentation position (absolute vertical, relative horizontal)	PP
Set graphic error action	SGEA
Set chain image	SCI
Set horizontal format (MPP)	SHF
Set vertical format (MPL)	SVF

3. Job data stream. The job data stream contains the base data stream, plus:

Control character	Abbreviation
Carriage return	CR
Line feed	LF
Set horizontal format (with parameters)	SHF
Horizontal tab	HT
Set vertical format (with parameters)	SVF
Vertical tab	VT
Backspace	BS
Vertical Channel Select (with parameters)	VCS
Transparency (with count and data)	TRN
Interchange record separator	IRS

WORD PROCESSING (WP) DATA STREAM PROFILES

WP text is a sequential stream of graphics and format controls. The text is ordered into lines and pages and separated by line- and page-ending SCS controls.

Base Data Stream Profile: The WP base data stream profile is identical to the DP base data stream profile as described in "Base Data Stream Profile" above.

Word Processing Raw-Form Text Data Stream Profile: The WP raw-form text DSP is used where further revision, processing, or reformatting may be done by a recipient. Because of the use of the extended WP control character set, this DSP can not be represented in a 7-bit code.

Since word processing is to be performed subsequent to transmission, recording on a machine-readable device is mandatory unless the entire extent of the processing is done concurrently with reception. Also, alphanumeric control information may be included to indicate to the recipient the intent of the originator. To interchange raw-form text, the originator must know the characteristics of the target machine and represent such control information in a manner that will be acceptable to the recipient(s).

In the absence of format-control information included within WP raw-form text, the default values shown in Figure 5-1 are recommended as standard.

Since further revision, processing, and reformatting may be performed subsequent to transmission, the choice of parameters may be made during such processing without relying on the exchange of format-control information from the originator. The SCS control characters that can appear in a WP raw-form DSP are listed in Figure 5-2 along with their allowable defaults.

No error condition may be returned to the sender if any other control character appears in the data stream, either as a single character or as initiating a multiple-character control sequence, since it is assumed that the recipient may readily correct any such errors in the process of final revision and processing. Instead, a substitute (SUB) control character (or other alternate control character) may be recorded in lieu of an unsupported control

Parameter	Recommended Default
Paper source selection	Primary
Keyboard identification	US 1
Type style, pitch	Elite, 12
Quantity	1
Top margin	1 inch
Left margin	1.5 inches
Vertical line spacing	6 per inch
Horizontal tab settings	Every 5 characters from the left margin (position 5, 10, 15, etc.)
Printing line width	6 inches
Headings	0.5 inches from top of page
Page numbers/footings	Centered, 0.5 inches from bottom of page

Figure 5-1. Defaults for WP raw-form text DSP.

character, as appropriate, to assist in subsequent correction. In accepting a WP raw-form text DSP, acceptance of space (SP), numeric space (NSP), required space (RSP), syllable hyphen (SHY), and the full 96-character WP graphic set is mandatory. Use of a substitute graphic is permitted where the print element (or equivalent) includes fewer than 96 graphics, and no error may be indicated if any other graphic character appears in the data stream, as it is assumed that the recipient may readily revise the text to make a correction.

For fixed-pitch printing, those control and graphic characters that are meaningful only in proportional escapement printing are defaulted as shown in Figure 5-3.

Function	Abbreviation	Magnetic Card Code	Default	EBCDIC	
Backspace	(BS)	4A	*	16	
Carrier return	(CRE)(NL)	2B	*	15	
Horizontal tab	(HT)	2A	Space	05	
Indent tab	(IT)	6A	Tab	39	
Index	(INX)(LF)	0B	*	25	
Index return	(IR)	4B	CR	33	
Null	(NUL)	33	*	00	
Numeric backspace	(NBS)(EBS)	52	BS	36	(1)
Numeric space	(NSP)(ESP)	78	SP	E1	
Page end	(PE)(FF)	7A	NL	0C	(2)
Repeat	(RPT)	7B	STOP	0A	
Required carrier return	(RCR)(RNL)	6B	CR	06	
Required page end	(RPE)(RFF)	(1A,7A)	PE	3A	
Stop	(STP)	32	*	2F	(3)
Subscript	(SBS)	23	NULL	38	
Substitute	(SUB)	13	*	3F	
Superscript	(SPS)	03	NULL	09	
Switch	(SW)	22	STOP	2A	
Unit backspace	(UBS)	42	NULL	1A	(1)
Word underscore	(WUS)	12	NULL	23	

*Base Code must support

Note 1: These control characters are meaningful only in proportional escapement printing. They are permitted in a DSP where the destination is a printer which does not support proportional escapement control characters, and defaults as indicated are permitted only in fixed-pitch printing.

Note 2: Page end defaults to new line only in printers using roll or continuous-paper feed.

Note 3: More than one default is permitted.

Figure 5-2. SCS controls for WP raw-form text DSP.

Character		Default
Numeric backspace	(NBS)	Backspace
Numeric space	(NSP)	Space
Unit backspace	(UBS)	No-op

Figure 5-3. Defaults for proportional escapement characters in fixed-pitch printing.

SENSE CODES AND SENSE DATA

SENSE CODES FOR LU_T4 SESSIONS

The following sense codes may be sent by LU_T4 half-sessions. Except for request error X'1008', complete descriptions are in SNA Format and Protocol Reference Manual: Architecture Logic.

Request Reject: 08xx

0801 Resource not available
0802 Intervention required
0809 Mode inconsistency
080A Permission rejected -- SSCP not notified

080D LU contention
080E LU not authorized
080F End user not authorized
0811 Break
0812 Insufficient resource
0813 Bracket bid reject
0815 Function active
081C Request not executable
0821 Invalid session parameters
0824 Unit of work aborted
0825 Component not available
0827 Intermittent error -- retry requested
0829 Change direction required
082D LU busy

Request Errors: 10xx

1001 RU data error
1002 RU length error
1005 Parameter error
1008 Invalid FM header

When 1008, invalid FM header, is returned, bytes 2 and 3 of the sense may contain X'0000' or may optionally contain a detailed explanation of the FM header error, as follows:

1008 0801 Invalid function code parameters
1008 0803 Forms function cannot be performed
1008 0805 Unable to perform copy function
1008 0806 Compaction table outside supported set
1008 0807 Invalid PDIR identifier
1008 0808 Train function cannot be performed
1008 0809 FCB load function cannot be performed
1008 080A FCB load function not supported
1008 080B Invalid compaction table name
1008 080C Invalid access
1008 080D Invalid RECLEN
1008 080E Invalid NUMRECS
1008 080F Data set in use
1008 0810 Data set not found
1008 0811 Invalid password
1008 0812 Function not allowed for destination
1008 0813 Record too long
1008 0814 Data set full
1008 0815 Invalid RECID
1008 0817 Invalid VOLID format
1008 0818 Number of logical records per chain exceeded
1008 0819 Data set exists
1008 081A No space available
1008 081B Invalid VOLID
1008 081C Invalid DSACCESS
1008 081D Invalid RECTYPE
1008 081E Insufficient resolution space
1008 081F Invalid key technique
1008 0820 Invalid key displacement
1008 0821 Invalid key
1008 0822 Invalid N (number)
1008 0823 Invalid KEYIND
1008 0824 Invalid SERID
1008 0826 Invalid RECID format
1008 0827 Password not supplied
1008 0828 Record ID not supplied
1008 0829 Volume ID not supplied
1008 082A Invalid PGMNAME
1008 2001 Invalid destination -- active
1008 2002 Invalid destination -- inactive
1008 2003 Invalid destination -- suspended
1008 2004 Invalid Suspend-Resume sequence
1008 2005 Interruption level violation
1008 2006 Invalid Resume properties
1008 2007 Destination not available

1008 2008	Invalid End sequence
1008 2009	Invalid FM header length
1008 200A	Invalid field setting
1008 200B	Invalid destination
1008 200C	Invalid ERCL
1008 200D	Invalid DST
1008 200E	Invalid concatenation
1008 200F	FM data not allowed for header
1008 2010	BIND FM header set violation
1008 2014	FM header not sent concatenated
1008 2019	Invalid stack reference indicator (SRI)
1008 201A	Unable to accept CMI modification
1008 201B	Unable to accept CPI modification
1008 201C	Unable to accept ERCL modification
1008 4001	Invalid FMH type
1008 4002	Invalid FMH code
1008 4003	Compression not supported
1008 4004	Compaction not supported
1008 4005	Basic exchange not supported
1008 4006	Only basic exchange supported
1008 4007	Medium not supported
1008 4008	Code selection compression violation
1008 4009	FMHC not supported
1008 400A	Demand select not supported
1008 400B	DSNAME not supported
1008 400C	Invalid media subaddress field
1008 400D	Insufficient resources to perform function requested
1008 400E	DSP select not supported

LUSTATS FOR LU_T4 SESSIONS

Byte 3 of the LUSTAT RU identifies the component class and device address of the LU. Because there are more component classes in LU_T4 than the component-medium field can identify, a byte-3 escape character of X'FF' indicates that byte 4 must be used to identify the component class and device address.

You may use byte 3 of the LUSTAT RU if the component class you desire is defined by byte 3. By so doing, the LUSTAT RU for LU_T4 is compatible with the other LU-LU session types.

The following LUSTAT codes may be sent by LU_T4 half-sessions:

0000	NOP
0001	Component now available
0002	No FM data to transmit
0003	Entering attended mode of operation
0004	Entering unattended mode of operation
0007	Sender currently has no data to transmit
0802	Component failure -- intervention required
081C	Component failure -- permanent error
0824	Function canceled

The format of LUSTAT is:

Byte	Value	Definition
0	X'04'	Request code
1-4		Status value and status extension field (two bytes each)
	X'0000'+'uuuu'	User status (no system defined status) and user-defined field (uuuu)
	X'0001'+'ccdd'	Component now available (0001) and component identification (ccdd)
	X'0002'+'rrrr'	No FMD requests to send and reserved field.

X'0003'+'ccdd' Entering attended mode of operation (0003) and component identification (ccdd)

X'0004'+'ccdd' Entering unattended mode of operation (0004) and component identification (ccdd)

X'0007'+'rrrr' Sender currently has no data to transmit (0007) and reserved field (rrrr)

X'0802'+'ccdd' Component failure - intervention required (0802) and component identification (ccdd)

X'081C'+'ccdd' Component failure - permanent error (081C) and component identification (cc00)

X'0824'+'rrrr' Function canceled (0824) and reserved (rrrr)

Values for 'cc' byte are:

X'00' = LU itself rather than a specific LU component
X'FF' = Use 'dd' byte rather than 'cc' byte to select LU component

Otherwise,
Bit 0, set to 1 = Media other than LU
Bits 1-3, LU component class:

'000' Console
'001' Exchange
'010' Card
'011' Document
All other values are invalid for LU_T4.

Bits 4-7 are the LU component device address.

If LUSTAT byte 3 is X'00', it refers to the LU itself rather than a specific LU component.

If LUSTAT byte 3 is X'FF', then the values for byte 4 are:

Bits 0-3 are the LU component-medium class
(DSP in parentheses below is default DSP).

'0000' Console
'0001' Exchange
'0010' Card
'0011' Document
'0111' Data set name selects destination
'1000' WP medium 1 (raw-form text)
'1001' WP medium 2 (raw-form text)
'1010' WP medium 3 (raw-form text)
'1100' WP medium 4 (raw-form text)
All other values are invalid for LU_T4.

Bits 4-7 are the LU component device address.

LU_T4 does not allow BB (begin bracket) on LUSTAT.
If any of the four WP medium select code points are supported (used) by an LU then all four WP medium codes must be supported. Each WP medium select code point must be supported in one of the following ways:

1. The WP medium select code is assigned a specific medium class, for example, printer, diskette, and so on. Assignment of a WP medium in this manner does not affect the default DSP for WP media; the default remains WP raw form.

2. The WP medium select code is defaulted to one of the classes assigned in 1 above.

An implementation may provide a user override capability for these medium class assignments (for example, the implementation assignment for WP Media 1 to diskette).

If none of the four WP medium select codes is supported, then the receiver of an FMH-1 indicating a WP media select should negatively respond with sense code X'1008'.

EXAMPLES OF DATA DELIMITING SEQUENCES

This section contains information that will help you better understand previously discussed LU_T4 architecture. The sequences in the Figures 5-4 through 5-7 below are illustrative only. Other sequences are allowed.

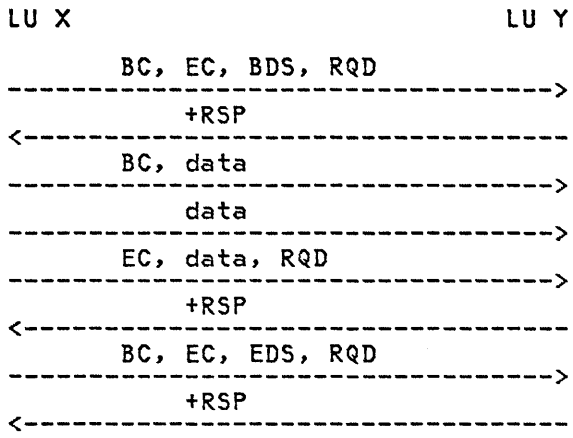


Figure 5-4. Basic data delimiting.

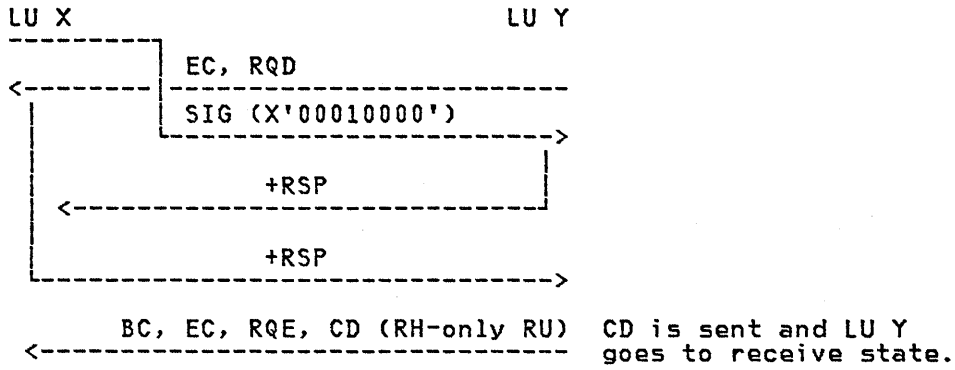


Figure 5-5. SIG (request to send) while between chains.

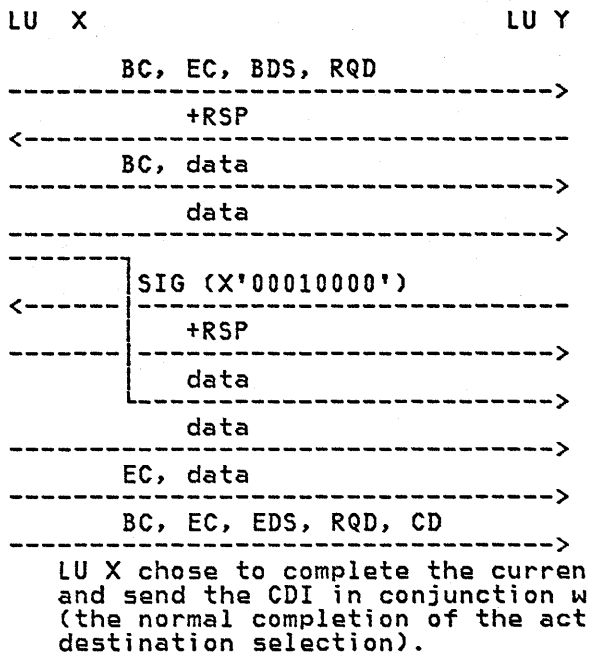


Figure 5-6. SIG (request to send) usage.

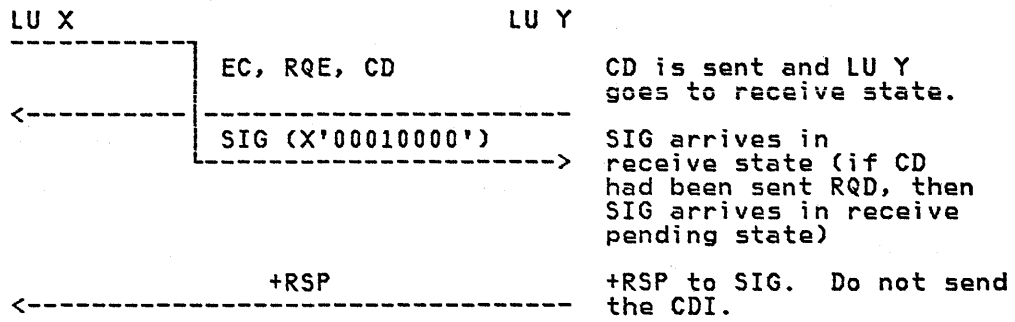


Figure 5-7. SIG (request to send) while in receive state.

BIND SESSION FORMATS FOR TYPE 4 LU-LU SESSIONS

This section defines valid bit settings for the BIND request. The BIND for LU_T4 can be sent as a negotiable or nonnegotiable request. In a negotiable BIND, the BIND sender and BIND receiver can negotiate to establish session protocols. In the nonnegotiable BIND, the BIND sender specifies the session parameters.

The formats of both the nonnegotiable BIND and the negotiable BIND request and response are shown below. For more information on the relationship between the types of BIND, see "BIND Request Parameters" at the beginning of this chapter.

NONNEGOTIABLE BIND

Byte	Bits	Value	Discussion
0	0-7	X'31'	Identifies this RU as a BIND request.
1	0-3	X'0'	Specifies the format of the BIND RU. Only one format has been defined: format 0 (zero).
(1)	4-7		Type - denotes the type of BIND to be performed:
		X'1'	Nonnegotiable BIND

FM (Function Management) Profile

2	0-7	X'07'	FM profile 7 must be used. The FM profile defines data flow control (DFC) protocols. (See SNA Reference Summary for highlights of the FM profile.)

TS (Transmission Subsystem) Profile

3	0-7	X'07'	TS profile 7 must be used. The TS profile defines transmission control (TC) protocols. (See SNA Reference Summary for highlights of the TS profile.)

FM Usage

(4-7)			See bytes 4 through 7 below.

BIND Sender's Send Protocols

			Chaining Use:
4	0	B'0' B'1'	Not valid BIND sender can send single or multiple element chains.
			Request Mode Selection:
(4)	1	B'0'	Immediate request mode is used. Only one definite response can be outstanding at a time. That response must be received before the BIND sender can send another RU.

Byte	Bits	Value	Discussion
		B'1'	Delayed request mode is used. Multiple definite responses can be outstanding at one time. Note: If delayed request mode is used, the RU sender must be responsible for recovery.
			Chaining Responses:
(4)	2-3	B'00' B'01' B'10' B'11'	Not valid Not valid Not valid BIND sender can request definite or exception responses.
(4)	4-5		Reserved
			Compression Indicator:
(4)	6	B'0' B'1'	BIND sender cannot send compressed data. BIND sender can send compressed data. Note: B'1' is valid only if the code is EBCDIC (see byte 6 bit 4 and byte 23 bits 4-7).
			Send End Bracket Indicator:
(4)	7	B'0' B'1'	BIND sender will not send EB (see byte 6 bit 2) BIND sender can send EB.

BIND Receiver's Send Protocols

			Chaining Use:
5	0	B'0' B'1'	Not valid BIND receiver can send single or multiple element chains.
			Request Mode Selection:
(5)	1	B'0' B'1'	Immediate request mode is used. BIND receiver can issue a request for a single definite response. No further transmission is sent until the BIND receiver receives the response. Delayed request mode is used. BIND receiver allows several definite responses to be outstanding at one time. Note: If delayed request mode is used, the RU sender must be responsible for recovery.
			Chaining Responses:
(5)	2-3	B'00' B'01' B'10' B'11'	Not valid Not valid Not valid BIND receiver can request either definite or exception responses.
(5)	4-5		Reserved
			Compression Indicator:
(5)	6	B'0' B'1'	BIND receiver cannot send compressed data. BIND receiver can send compressed data. Note: B'1' is valid only if the code is EBCDIC (see byte 6 bit 4 and byte 23 bits 4-7).
			Send End Bracket Indicator:
(5)	7	B'0' B'1'	BIND receiver cannot send EB (see byte 6 bit 2) BIND receiver can send EB.

Byte	Bits	Value	Discussion
Common Protocols for FM Data			
6	0		Reserved
(6)	1	B'0'	FM headers not used.
		B'1'	BIND sender and BIND receiver can exchange FM headers.
			Brackets Usage and Reset State:
			Note: Brackets are optional in LU_T4 sessions.
(6)	2	B'0'	When brackets are used (byte 4, bit 7 = 1 or byte 5, bit 7 = 1), the bracket state manager's reset states are INB (in brackets). Brackets are not used if neither BIND sender nor receiver will send EB (when byte 4, bit 7 = 0 and byte 5, bit 7 = 0).
		B'1'	Not valid
			Bracket Termination Rule Selection:
(6)	3	B'0'	Not valid
		B'1'	Bracket termination rule 1 is used.
			Note: Bit 3 is reserved if brackets are not used by the session.
			Alternate Code Selection:
(6)	4	B'0'	Alternate code set will not be used.
		B'1'	Alternate code set may be used. (See byte 23 bits 4-7.)
(6)	5-7		Reserved
			Normal-Flow Send and Receive Mode Selection:
7	0-1	B'00'	Not valid
		B'01'	Not valid
		B'10'	Half-duplex flip-flop
		B'11'	Reserved
			Recovery Responsibility:
(7)	2	B'0'	The contention loser is responsible for any error recovery attempts for the session (asymmetric recovery).
		B'1'	Symmetric responsibility for recovery. The sender of data is responsible for recovery.
			Contention Winner/Loser:
(7)	3	B'0'	BIND receiver is contention winner, and BIND sender is contention loser.
		B'1'	BIND sender is contention winner, and BIND receiver is contention loser.
			Note: Byte 7, bit 3, is reserved if there is symmetric responsibility for recovery (when byte 7, bit 2 = 1). For asymmetric recovery (when byte 7, bit 2 = 0), the session partner responsible for recovery is the contention loser identified in byte 7, bit 3.
(7)	4-6		Reserved
			Half-Duplex Flip-Flop (HDX-FF) Reset States:
(7)	7	B'0'	Not valid
		B'1'	HDX-FF reset state is SEND for the BIND sender and RECEIVE for the BIND receiver (that is, the BIND sender sends normal-flow requests first after session activation or reset).

Byte	Bits	Value	Discussion
TS Usage			
(8-13)			See bytes 8 through 13 below.
			Staging Indicator and Pacing Count for BIND-receiver CPMGR (connection point manager) to BIND-sender CPMGR normal flow:
8	0	B'0' B'1'	Pacing in this direction occurs in one stage. Pacing in this direction occurs in two stages.
			Note 1: When the session involves a boundary function, pacing may occur in two stages. For the definition of a boundary function, see SNA Format and Protocol Reference Manual: Architecture Logic.
			Note 2: The meanings of 0 and 1 are reversed from the staging indicator for BIND-sender CPMGR to BIND-receiver CPMGR (see byte 12).
(8)	1		Reserved
(8)	2-7	B'nnnnnn'	BIND-receiver CPMGR send pacing count. A value of 000000 means no pacing of requests flowing from the BIND receiver.
9	0-1		Reserved
(9)	2-7	B'nnnnnn'	BIND-receiver CPMGR receive pacing count. A value of 000000 causes the boundary function to substitute the value set by a system definition pacing parameter (if the system definition includes such a parameter) before it sends the BIND RU on to the BIND receiver. A value of 000000 received at the BIND receiver is interpreted to mean no pacing of requests flowing to the BIND receiver.
			Maximum RU Size:
10		X'ab'	Maximum RU size sent on the normal flow by the BIND receiver. This value represents the largest RU that can be sent. It is expressed as a mantissa and an exponent value of 2 by which the mantissa is multiplied. For example, when the mantissa is 8 and the exponent is 5 (X'85'), the RU size is 256 bytes (8×2^5).
			When bit 0 is set to zero, no maximum is specified and the remaining bits 1-7 are ignored. When bit 0 is set to one, the byte is interpreted as X'ab'. (See Appendix B for these values.)
11		X'ab'	Maximum RU size sent on the normal flow by the BIND sender. This value represents the largest RU that can be sent by the BIND sender and is specified in the same format as the BIND receiver (byte 10).
			Staging Indicator and Pacing Count for BIND-sender CPMGR to BIND-receiver CPMGR normal flow:
12	0	B'0' B'1'	Pacing in this direction occurs in two stages. Pacing in this direction occurs in one stage.
			Note: The meanings of 0 and 1 are reversed from the staging indicator for BIND-receiver CPMGR to BIND-sender CPMGR (see byte 8, notes 1 and 2).
(12)	1		Reserved
(12)	2-7	B'nnnnnn'	BIND-sender CPMGR send pacing count. A value of 000000 means no pacing of requests flowing from the BIND sender.
			For single-stage pacing in the BIND sender to BIND receiver direction, this field is redundant with, and will indicate the

Byte	Bits	Value	Discussion
			same value as, the BIND-receiver CPMGR receive pacing count. (See byte 9, bits 2-7 above.)
13	0-1		Reserved
(13)	2-7	B'nnnnnn'	BIND-sender CPMGR receive pacing count. A value of 000000 means no pacing of requests that flow to the BIND sender. For single-stage pacing in the BIND receiver to BIND sender direction, this field is redundant with, and will indicate the same value as, the BIND-receiver CPMGR send pacing count. (See byte 8, bits 2-7 above.)

PS Profile

(14-25)			See bytes 14 through 25 below.
14	0	B'0'	PS usage field format is the basic format.
		B'1'	Reserved
(14)	1-7	B'0000100'	LU-LU session type 4

PS Usage

15-18			BIND SENDER TO BIND RECEIVER DIRECTION OF FLOW
15			Printer data stream profiles (Profiles are defined under "Data Stream Profiles.")
	0		Base DSP
		0	Not supported
		1	Supported
	1		General data stream
		0	Not supported
		1	Supported
	2		Job data stream
		0	Not supported
		1	Supported
	3		Reserved
	4		WP raw form data stream
		0	Not supported
		1	Supported
	5-7		Reserved
			Note: Base DSP with NL and FF is part of all other printer DSPs. If bit 0 = 0, no printer is supported on BIND receiver. If bit 1, 2, or 4 is set to 1, bit 0 must also be 1.
16			Additional data stream profiles (X'00' = no additional media)
			For bits 0 through 7 below:
		0	Not supported
		1	Supported
	0		Reserved
	1		Card
	2		Reserved
	3		Basic exchange

Byte	Bits	Value	Discussion
	4		Reserved
	5		Word processing exchange diskette
	6		Reserved
	7		Reserved
17			Console data stream profiles
	0-3		Console definition
	0		Base DSP (see Note 1)
		0	No console supported (see Note 2)
		1	Supported
	1		General data stream
		0	Not supported
		1	Supported (see Note 3)
	2		Job data stream
		0	Not supported
		1	Supported (see Note 3)
	3-7		Reserved
			Note: Base DSP with NL and FF is part of all other console DSPs. If bit 0 = 0, no console is supported on BIND receiver. If bit 1 or 2 is set to 1, bit 0 must also be 1.
18			FM and FMH usage
	0		Supervisory services FM data
		0	Not supported
		1	Supported
	1-2		Destination-selection stack
		00	1-level stack
		01	2-level stack
		10	Reserved
		11	3-level stack
	3		Compaction
		0	Not supported
		1	Supported
			B'1' is valid only if the code is EBCDIC. (See byte 6 bit 4 and byte 23 bits 4-7.)
	4		PDIR for all media (PDIR is peripheral data information record)
		0	Not supported
		1	Supported
	5		Reserved
	6		QUERY FOR DATA SET FMH-2
		0	Not supported
		1	Supported
	7	0	BIND receiver need not receive CD on every EDS (end of destination selection)
		1	BIND receiver must receive CD on every EDS
19-22			BIND RECEIVER TO BIND SENDER DIRECTION OF FLOW (Same format as bytes 15-18.)
19			Printer data stream profiles
	0		Base DSP
		0	Not supported
		1	Supported

Byte	Bits	Value	Discussion
	1		General data stream
		0	Not supported
		1	Supported
	2		Job data stream
		0	Not supported
		1	Supported
	3		Reserved
	4		WP raw form data stream
		0	Not supported
		1	Supported
	5-7		Reserved
			Note: Base DSP with NL and FF is part of all other printer DSPs. If bit 0 = 0, no printer is supported on BIND sender. If bit 1, 2, or 4 is set to 1, bit 0 must be 1.
20			Additional data stream profiles (X'00' = no additional media)
			For bits 0 through 7 below:
		0	Not supported
		1	Supported
	0		Reserved
	1		Card
	2		Reserved
	3		Basic exchange
	4		Reserved
	5		Word processing exchange diskette
	6		Reserved
	7		Reserved
21			Console data stream profiles
	0-3		Console definition
	0		Base DSP
		0	Not supported
		1	Supported
	1		General data stream
		0	Not supported
		1	Supported
	2		Job data stream
		0	Not supported
		1	Supported
	3-7		Reserved
			Note: Base DSP with NL and FF is part of all other console DSPs. If bit 0 = 0, no console is supported on BIND sender. If bit 1 or 2 is set to 1, bit 0 must also be 1.
22			FM and FMH usage
	0		Supervisory services FM data
		0	Not supported
		1	Supported
	1-2		Destination-selection stack
		00	1-level stack
		01	2-level stack
		10	Reserved
		11	3-level stack

Byte	Bits	Value	Discussion
	3		Compaction
		0	Not supported
		1	Supported
	4		PDIR for all media
		0	Not supported
		1	Supported
	5		Reserved
	6		QUERY FOR DATA SET FMH-2
		0	Not supported
		1	Supported
	7	0	BIND sender need not receive CD on every EDS (end of destination selection)
		1	BIND sender must receive CD on every EDS
23			Code selection
	0-3		Repertoire
			Note: Bits 0-3 are not defined for the nonnegotiable BIND. Use bits 4-7 instead.
	4-5		Code 0 (main code) selection. Specifies the encoding of character fields in code 0 during the session.
		00	EBCDIC
		01	ASCII or ISCII or ITA #5
			Other values reserved
	6-7		Code 1 (alternate code) selection. Selects the alternate code for the session.
		00	EBCDIC
		01	ASCII or ISCII or ITA #5
			Other values reserved
			If byte 6 bit 4 is B'1', the code selection indicator (CSI field of the RH (request header)) controls whether the alternate code is to be used.
24			General characteristics
	0-1		Reserved
	2	0	BIND sender may send data first
		1	BIND receiver must send data first
	3		Reserved
	4	0	BIND receiver will initiate attended mode (Where operator is available to handle intervention-required and other operator responses.)
		1	BIND receiver will initiate unattended mode
	5	0	During session BIND receiver will not alternate between attended and unattended mode
		1	During session BIND receiver may alternate between attended and unattended mode
	6-7		Reserved
25	Reserved		

Cryptography Options

Byte	Bits	Value	Opt	Discussion
				An X in OPT column indicates that value is optional.
26	0-1			Private cryptography options:
		B'00'	X	No private cryptography supported.
		B'01'	X	Private cryptography supported. The session cryptography key and cryptography protocols are privately supplied by the end user.
		B'10'		Reserved
		B'11'		Reserved
(26)	2-3			Session-level cryptography options:
		B'00'	X	No session-level cryptography supported.
		B'01'	X	Session-level selective cryptography supported. All cryptography key management is supported by the SSCP and LU; exchange (using +RSP to BIND) and verification (using CRV) of the cryptography session-seed value are supported by the LUs for the session. All FM data requests with EDI (enciphered data indicator) on are enciphered/deciphered.
		B'10'		Reserved
		B'11'	X	Session-level mandatory cryptography supported; same as session-level selective cryptography except all FM data requests are enciphered/deciphered.
(26)	4-7			Session-level cryptography options field length:
		X'0'	X	No session-level cryptography specified. The cryptography option fields (bytes 27-k) are omitted.
		X'9'	X	Session-level cryptography specified. Additional options follow in the next nine bytes.
27	0-1			Session cryptography key encipherment mode:
		B'00'	X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key using a seed value of zero (only value defined).
(27)	2-4			Reserved
(27)	5-7			Cryptography cipher method:
		B'000'	X	Block chaining with seed and cipher text feedback, using the Data Encryption Standard (DES) algorithm (only value defined).
28-k			X	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key; an eight-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FM data requests.

Names and End User Data

k+1		Length of BIND sender's name in field k+2-m.
	X'00'	Field k+2-m does not exist.
	X'nn'	Field k+2-m contains the number of bytes specified (cannot exceed X'08').
k+2-m		BIND sender's LU name.
m+1		Length of user data in field m+2-n.
	X'00'	Field m+2-n does not exist.
	X'nn'	Field m+2-n contains the number of bytes specified.

Byte	Bits	Value	Opt	Discussion
m+2-n			X	User data.
m+2			X	User data key.
		X'00'	X	Structured subfields follow.
		¬X'00'	X	First byte of unstructured user data.
				Note: The unstructured user data continues in m+3 and goes to byte n. For information on structured subfields, see the BIND information in SNA Format and Protocol Reference Manual: Architecture Logic.
n+1				Length of user request correlation field.
		X'00'		Field n+2-p does not exist.
		X'nn'		Field n+2-p contains the number of bytes specified.
n+2-p				User request correlation field.
p+1				Length of BIND receiver's name in field p+2-r.
		X'00'		Field p+2-r does not exist.
		X'nn'		Field p+2-r contains the number of bytes specified (cannot exceed X'08').
p+2-r				BIND receiver's LU name.

Definitions for Nonnegotiable BIND

The following definitions apply to bytes 14 to 25 of the nonnegotiable BIND request:

1. LU-LU session type (byte 14):
   This field specifies LU session type 4 and thus identifies the format and field definitions for bytes 15 through 25.

2. BIND sender to BIND receiver flow (bytes 15 to 18):
   This field defines the proposed session parameters for the BIND sender's send capability and, therefore, the BIND receiver's receive capability. Acceptance of the BIND by the BIND receiver means that these session parameters apply to this direction of transmission.

3. BIND receiver to BIND sender flow (bytes 19 to 22):
   This field defines the proposed session parameters for the BIND receiver's send capability and, therefore, the sender's receive capability. Acceptance of the BIND by the BIND receiver means that these session parameters apply to this direction of transmission.

4. Code selection (byte 23):
   The code selection repertoire (bits 0 to 3) is not defined in the nonnegotiable BIND. Any setting of bits in this field conveys no meaningful information.

   The code 0 selection field (bits 4 and 5) defines the code for use in the session as the main code (that is, for an RU in which the code selection indicator in the RH is set to B'0').

   The code 1 selection field (bits 6 and 7) defines the code for use in the session as the alternate code (that is, for an RU in which the code selection indicator is set to B'1').

   The code selection indicator in the request header (RH) allows for an individual RU to be represented in an alternate code. If the indicator is B'0', the main code is used; if B'1', the alternate code is used. For those RUs containing FM headers, the header is never subject to alternate coding.

If alternate code is not to be used in the session (that is, if bit 4 of byte 6 of the BIND request is set to B'0'), then the code 1 selection field is meaningless.

5. General characteristics (byte 24):
This field defines general characteristics proposed for the session. Acceptance of the BIND by the BIND receiver means that these parameters apply to the session.

NEGOTIABLE BIND REQUEST

Byte	Bits	Allowable request settings	Allowable response settings	Discussion
0	0-7	X'31'	X'31'	BIND request
1	0-3	X'0'	X'0'	Format
	4-7	X'0'	X'0'	Negotiable BIND
2	0-7	X'07'	X'07'	FM profile 7
3	0-7	X'07'	X'07'	TS profile 7

FM Usage

4		BIND Sender's Send Protocols	BIND Receiver's Receive Protocols	
	0	B'1'	B'1'	Single and multiple element chains
	1	B'0'	B'0'	B'0' = Immediate request mode
		B'1'	B'0',B'1'	B'1' = Delayed request mode (Requires symmetric error recovery.)
	2-3	B'11'	B'11'	Definite or exception-only responses
	4-5			Reserved
	6	B'0'	B'0',B'1'	B'0' = BIND sender cannot send compressed data
		B'1'	B'0',B'1'	B'1' = BIND sender can send compressed data
	7	B'0'	B'0'	B'0' = BIND sender will not send EB
		B'1'	B'0',B'1'	B'1' = BIND sender can send EB
5		BIND Sender's Receive Protocols	BIND Receiver's Send Protocols	
	0	B'1'	B'1'	BIND receiver sends single or multiple element chains
	1	B'0'	B'0'	B'0' = Immediate request mode
		B'1'	B'0',B'1'	B'1' = Delayed request mode (Requires symmetric error recovery.)
	2-3	B'11'	B'11'	Definite or exception-only responses
	4-5			Reserved

Byte	Bits	Request settings	Response settings	Discussion
	6	B'0'	B'0'	B'0' = BIND receiver cannot send compressed data
		B'1'	B'0',B'1'	B'1' = BIND receiver can send compressed data
	7	B'0'	B'0'	B'0' = BIND receiver cannot send EB
		B'1'	B'0',B'1'	B'1' = BIND receiver can send EB
6		Request Settings	Response Settings	Common Protocols for FM Data
	0			Reserved
	1	B'0'	B'0'	B'0' = FM headers not used
		B'1'	B'0',B'1'	B'1' = BIND sender and BIND receiver can exchange FM headers
	2	B'0'	B'0'	When brackets are used, bracket state manager's reset state is "in brackets."
	3	B'1'	B'1'	Bracket termination rule 1
	4	B'0'	B'0'	B'0' = Alternate code set will not be used
		B'1'	B'0',B'1'	B'1' = Alternate code set may be used
	5-7			Reserved
				Note: Byte 6 bit 3 is reserved if brackets are not used by the session. If brackets are not used and asymmetric recovery responsibility is indicated (byte 7, bit 2), then byte 7 bit 3 indicates which session partner is responsible for recovery.
7	0-1	B'10'	B'10'	Half-duplex flip-flop
	2	B'0'	B'0',B'1'	B'0' = Contention loser responsible for recovery (asymmetric recovery)
		B'1'	B'1'	B'1' = Symmetric recovery responsibility
	3	B'0'	B'0'	B'0' = BIND receiver is contention winner and BIND sender is contention loser
		B'1'	B'0',B'1'	B'1' = BIND sender is contention winner and BIND receiver is contention loser
	4-6			Reserved
	7	B'1'	B'1'	HDX-FF reset state is send for BIND sender and receive for BIND receiver
				Note: Byte 7 bit 3 is reserved if there is symmetric responsibility for recovery (when byte 7 bit 2 = 1) and brackets are not used.

TS Usage

8	0	B'0'	B'0'	One-stage pacing
		B'1'	B'1'	Two-stage pacing
	1			Reserved
	2-7	B'a'	B'a'	BIND-receiver CPMGR send pacing count
				Note 1: When the session involves a boundary function, pacing may occur in two stages. For the

Byte	Bits	Request settings	Response settings	Discussion
				definition of a boundary function, see SNA Format and Protocol Reference Manual: Architecture Logic.
				Note 2: The meanings of 0 and 1 are reversed from the staging indicator for BIND-sender CPMGR to BIND-receiver CPMGR (see byte 12).
9	0-1			Reserved
	2-7	B'b'	B'≤b'	BIND-receiver CPMGR receive pacing count
				Note: The BIND receiver may reduce the value, but it may not be reduced to zero unless the BIND receiver does not support pacing in the BIND sender to BIND receiver direction of flow. If this value is reduced and byte 12 bit 0 is one, the BIND sender's send pacing count (byte 12 bits 2-7) must be reduced to the same value.
10	0-7	X'ab'	X'ab'	Maximum RU size sent by the BIND receiver. The receiver cannot change the value. (See Appendix B for these values.)
11	0-7	X'ab'	X'≤ab'	Maximum RU size sent by the BIND sender (See Appendix B for these values.)
				Staging Indicator and Pacing Count for BIND sender to BIND receiver normal flow:
12	0	B'0' B'1'	B'0' B'1'	Two-stage pacing One-stage pacing
	1			Reserved
	2-7	B'c'	B'≤c'	BIND-sender CPMGR send pacing count
13	0-1			Reserved
	2-7	B'd'	B'd'	BIND-sender CPMGR receive pacing count (Byte 8 must equal byte 13 for single stage pacing.)
14	0	B'0'	B'0'	PS usage field format is the basic format
	1-7	B'0000100'	B'0000100'	LU-LU session type 4

BIND Sender to BIND Receiver Flow

15				Printer Data Stream Profiles (DSPs) (B'0' = No support, B'1' = Supported)
	0	B'0',B'1'	B'0',B'1'	Base DSP
	1	B'0',B'1'	B'0',B'1'	General data stream
	2	B'0',B'1'	B'0',B'1'	Job data stream
	3	B'0'	B'0'	Reserved
	4	B'0',B'1'	B'0',B'1'	WP raw form data stream
	5-7	B'000'	B'000'	Reserved
				Note: Base DSP with NL and FF is part of all other printer DSPs. If bit 0 = 0, no printer is supported on BIND receiver. If bit 1, 2, or 4 is set to 1, bit 0 must be 1.
16				Additional data stream profiles (X'00' = No additional media) (B'0' = No support, B'1' = Supported)

Byte	Bits	Request settings	Response settings	Discussion
	0	B'0'	B'0'	Reserved
	1	B'0',B'1'	B'0',B'1'	Card
	2	B'0'	B'0'	Reserved
	3	B'0',B'1'	B'0',B'1'	Basic exchange
	4	B'0'	B'0'	Reserved
	5	B'0',B'1'	B'0',B'1'	Word processing exchange diskette
	6	B'0'	B'0'	Reserved
	7	B'0'	B'0'	Reserved
17				Console data stream profiles (B'0' = No support, B'1' = Supported)
	0	B'0',B'1'	B'0',B'1'	Base DSP (see Note)
	1	B'0',B'1'	B'0',B'1'	General data stream
	2	B'0',B'1'	B'0',B'1'	Job data stream
	3-7	B'00000'	B'00000'	Reserved
				Note: Base DSP with NL and FF is part of all other console DSPs. If bit 0 = 0, no console is supported on BIND receiver. If bit 1 or 2 is set to 1, bit 0 must be 1.
18				FM and FMH usage (B'0' = No support, B'1' = Supported)
	0	B'0',B'1'	B'0',B'1'	Supervisory services FM data
	1-2	B'nn'	B'nn'	Destination-selection suspension stack (nn = 00 or 01 or 11; nn = 10 is reserved)
	3	B'0',B'1'	B'0',B'1'	Compaction
	4	B'0',B'1'	B'0',B'1'	PDIR for all media
	5	B'0'	B'0'	Reserved
	6	B'0',B'1'	B'0',B'1'	QUERY FOR DATA SET FMH-2
	7	B'0',B'1'	B'0'	BIND receiver need not receive CD on every EDS (end of destination selection)
		B'0',B'1'	B'1'	BIND receiver must receive CD on every EDS

BIND Receiver to BIND Sender Flow

19				Printer data stream profiles (DSPs) (B'0' = No support, B'1' = Supported)
	0	B'0',B'1'	B'0',B'1'	Base DSP
	1	B'0',B'1'	B'0',B'1'	General data stream
	2	B'0',B'1'	B'0',B'1'	Job data stream
	3	B'0'	B'0'	Reserved
	4	B'0',B'1'	B'0',B'1'	WP raw form data stream
	5-7	B'000'	B'000'	Reserved
				Note: Base DSP with NL and FF is part of all other printer DSPs. If bit 0 = 0, no printer is supported on BIND sender. If bit 1, 2, or 4 is set to 1, bit 0 must be 1.
20				Additional data stream profiles (X'00' = no additional media) (B'0' = No support, B'1' = Supported)
	0	B'0'	B'0'	Reserved
	1	B'0',B'1'	B'0',B'1'	Card
	2	B'0'	B'0'	Reserved
	3	B'0',B'1'	B'0',B'1'	Basic exchange
	4	B'0'	B'0'	Reserved
	5	B'0',B'1'	B'0',B'1'	Word processing exchange diskette
	6	B'0'	B'0'	Reserved
	7	B'0'	B'0'	Reserved

Byte	Bits	Request settings	Response settings	Discussion
21				Console data stream profiles (B'0' = No support, B'1' = Supported)
	0	B'0',B'1'	B'0',B'1'	Base DSP (see Note)
	1	B'0',B'1'	B'0',B'1'	General data stream
	2	B'0',B'1'	B'0',B'1'	Job data stream
	3-7	B'00000'	B'00000'	Reserved
				Note: Base DSP with NL and FF is part of all other console DSPs. If bit 0 = 0, no console is supported on BIND sender. If bit 1 or 2 is set to 1, bit 0 must be 1.
22				FM and FMH usage (B'0' = No support, B'1' = Supported)
	0	B'0',B'1'	B'0',B'1'	Supervisory services FM data
	1-2	B'nn'	B'nn'	Destination-selection suspension stack (nn = 00 or 01 or 11; nn = 10 is reserved)
	3	B'0',B'1'	B'0',B'1'	Compaction
	4	B'0',B'1'	B'0',B'1'	PDIR for all media
	5	B'0'	B'0'	Reserved
	6	B'0',B'1'	B'0',B'1'	QUERY FOR DATA SET FMH-2
	7	B'0',B'1'	B'0'	B'0' = BIND sender need not receive CD on every EDS (end of destination selection)
		B'0',B'1'	B'1'	B'1' = BIND sender must receive CD on every EDS
23				Code selection (B'0' = No support, B'1' = Supported)
	0-3			Repertoire
	0	B'0',B'1'	B'0',B'1'	EBCDIC
	1	B'0',B'1'	B'0',B'1'	ASCII or ISCII or ITA 5
	2-3	B'0'	B'0'	Reserved
	4-5	B'nn'	B'nn'	Code 0 (main code) selection. (nn = 00 (EBCDIC) or 01 (ASCII); other values reserved)
	6-7	B'nn'	B'nn'	Code 1 (alternate code) selection. (nn = 00 (EBCDIC) or 01 (ASCII); other values reserved)
24				General characteristics
	0-1	B'00'	B'00'	Reserved
	2	B'0'	B'0',B'1'	B'0' = BIND sender may send data first
		B'1'	B'0',B'1'	B'1' = BIND receiver must send data first
	3	B'0'	B'0'	Reserved
	4	B'0',B'1'	B'0'	B'0' = BIND sender/receiver will initiate attended mode
		B'0',B'1'	B'1'	B'1' = BIND sender/receiver will initiate unattended mode
	5	B'0',B'1'	B'0'	B'0' = BIND sender/receiver will not alternate between attended and unattended mode
		B'0',B'1'	B'1'	B'1' = BIND sender/receiver may alternate between attended and unattended mode
	6-7	B'00'	B'00'	Reserved
25				Reserved

Cryptography Options

Byte	Bits	Request settings	Response settings	Opt	Discussion
					An X in OPT column indicates value is optional.
26	0-1				Private cryptography options:
		B'00'	B'00'	X	No private cryptography supported
		B'01'	B'00',B'01'	X	Private cryptography supported
		B'10'	B'00',B'10'		Reserved
		B'11'	B'00'		Reserved
					Note: BIND receiver must accept sender's choice of cryptography or return B'00'.
	2-3				Session-level cryptography options:
		B'00'	B'00'	X	No session-level cryptography supported
		B'01'	B'00',B'01'	X	Session-level selective cryptography supported
		B'10'	B'00'		Reserved
		B'11'	B'00',B'01'	X	Session-level mandatory cryptography supported
	4-7				Session-level cryptography options field length:
		X'0'	X'0'	X	No session-level cryptography specified
		X'9'	X'9'	X	Session-level cryptography specified
27	0-1				Session cryptography key encipherment mode:
		B'00'	B'00'	X	Key enciphered under BIND receiver key
	2-4	B'000'	B'000'		Reserved
	5-7				Cryptography cipher method:
		B'000'	B'000'	X	Block chaining
28-k				X	Session cryptography key enciphered under BIND receiver master key

Names and End User Data

Byte	Bits	Request settings	Response settings	Discussion
k+1				Length of BIND sender's name in field k+2-m
		X'00'	X'00'	Field k+2-m does not exist
		X'nn'	X'nn'	Field k+2-m contains X'nn' bytes (≤X'08')
k+2-m				BIND sender's LU name
m+1				Length of user data in field m+2-n
		X'00'	X'00'	Field m+2-n does not exist
		X'nn'	X'nn'	Field m+2-n contains number of bytes specified
m+2-n				User data (response same format as request)
n+1				Length of user request correlation field
		X'00'	X'00'	Field n+2-p does not exist
		X'nn'	X'mm',X'00'	Field n+2-p contains number of bytes specified
n+2-p				User request correlation field
p+1				Length of BIND receiver's name in field p+2-r
		X'00'	X'00'	Field p+2-r does not exist
		X'nn'	X'nn'	Field p+2-r contains X'nn' bytes (≤X'08')
p+2-r				BIND receiver's LU name

Definitions for Negotiable BIND

The following definitions apply to bytes 14 to 25 of the negotiable BIND request:

1. LU-LU session type (byte 14): This field specifies LU session type 4 and thus identifies the format and field definitions for bytes 15 through 25.

2. BIND sender to BIND receiver flow (bytes 15 to 18):

   BIND sender's send capabilities (negotiable BIND request). This field provides information that assists the session partner in identifying the range of send capabilities that may be desired. The session partner should use this information when deciding how to adapt to these capabilities; the information may not be used to reject session activation.

   BIND receiver's receive capabilities (negotiable BIND response). This field defines the session parameters that are acceptable for this direction of flow. Acceptance of the session activation means that the BIND sender will comply with the BIND parameters for this direction of flow.

3. BIND receiver to BIND sender flow (bytes 19 to 22):

   BIND sender's receive capability (negotiable BIND request). This field defines the session parameters that are proposed for this direction of flow. The session partner may use this information in preparing the BIND response; the information may not be used to reject session activation.

   BIND receiver's send capability (negotiable BIND response). This field defines the range of session parameters that may be used in this direction of flow. Acceptance of session activation means that the BIND sender will comply with these parameters for this direction of flow.

4. Code selection (request and response) (byte 23).
   In the BIND request, the BIND sender specifies both the main (code 0) and alternate (code 1) codes proposed for use in the session. If the use of alternate code is not proposed (that is, if bit 4 of byte 6 of the BIND request is set to B'0'), then the code 1 selection field is meaningless. The BIND sender also indicates his code repertoire to assist the BIND receiver in identifying a mutually compatible code if multiple-code capability exists.

5. The capability to send, to receive, or to prohibit FM data in the supervisory services session is supported. In the case of a terminal-to-terminal or cluster-to-cluster session, these supervisory services are provided by the LUs. In a system managed by an SSCP, the SSCP provides these services to the LUs.

6. LU_T4 uses FMH-1 to control the beginning and ending of destination selection. Optionally, LU_T4 uses FMH-2 to process data at the active destination, and FMH-3 to send a compaction table for use by all LU components. Complete definitions of these headers and subsets supported are found in Part 2 Chapter 4 and under "FM Header Processing" in this chapter.

7. Support of compaction is optional. This indicator applies only to SCB compaction.

8. Support of the PDIR FMH-2 is optional. When supported, it applies to all media.

9. QUERY FOR DATA SET FMH-2 allows one LU to solicit input from the other's various sources: card, disk, and so on.

10. General characteristics (request and response) (byte 24).
    This field specifies additional session parameters that may be required by the session partners. If a conflict arises between the BIND sender and BIND receiver, the BIND response will resolve the conflict.

11. A console is optional for LU_T4.

The BIND receiver must adapt to the code selection in the BIND request as follows (applicable to products with multiple-code capability):

1. If the main code proposed in the BIND request exists in the repertoire of the BIND receiver, then the proposed main code must be accepted and such acceptance is indicated in the BIND response by the appropriate setting of the code 0 selection field.

2. If the main code proposed in the BIND request does not exist in the repertoire of the BIND receiver, a comparison is made with the code repertoire indicated in the BIND request. If any (one or more) mutually compatible code(s) exist, the BIND receiver specifies the main code to be used for the session by an appropriate setting of the code 0 selection field in the BIND response. The order of preference of this code selection is, first, that code specified in bit 0 of the code repertoire, followed in order by those codes specified by bits 1, 2, and 3.

3. If the BIND sender proposes that alternate code not be used, the BIND receiver must set bit 4 of byte 6 to B'0' in the BIND response. This makes the code 1 selection field meaningless.

4. If the BIND sender proposes that alternate code be used, and if that proposed alternate code exists in the repertoire of the BIND receiver, then the BIND receiver accepts the proposed alternate code by the appropriate setting of the code 1 selection field of the BIND response. Otherwise, the BIND receiver specifies that alternate code is not to be used by setting bit 4 of byte 6 to B'0' in the BIND response.

   Thus, the code selection to be used in the session is defined as being specified in the BIND response.

Type 6 LU-LU sessions are used for data communication between transaction processing systems (TPSs). The TPSs then provided functions that permit application programmers to develop transaction processing applications in which different parts of the application are performed at different TPSs.

The TPSs are interconnected via type 6 LU-LU sessions, and they provide two major services:

- They allow resources, such as files and queues, to be distributed throughout the network, and provide a means for easily accessing such resources
- They allow application programs, called transaction processing programs, to be distributed throughout the network, and provide a means for these transaction processing programs to communicate and cooperate with each other in processing transactions.

These two services are discussed in greater detail in SNA Concepts and Products as the remote resource access capability and the TPP (transaction processing program) conversational capability.

An example of an LU_T6 session is where one CICS/VS and one IMS/VS system are in session; a CICS/VS application program may communicate with an IMS/VS program. LU_T6 implementations use the term transaction to describe this type of communication between programs. Transactions are usually of short duration; the longer batch and interactive sessions are not precluded, however.

Within type 6 sessions, many transactions may occur on the session between TPSs. In this sense, a transaction is a unit of work initiated by a single request. Where the work is accomplished is determined by where the end user placed the transaction processing programs within the network.

In some cases, the location of the data (local or remote) becomes transparent to the programs. The programs can obtain access to remote data without issuing a unique transaction to the target system to access the data. The TPSs convert these requests to a standardized format and forward them to the appropriate TPS following the session's protocol rules.

LU-T6 sessions operate in a network controlled by one or more SSCPs. The half-sessions operate in a symmetric environment; that is, the partners are equals and neither controls the session. The half-session that sends the data assumes error recovery responsibility for the data.

The session begins when one TPS sends a BIND SESSION request to the desired session partner (another TPS). The BIND SESSION request identifies those functions the BIND sender wishes to use. The request may be a negotiable BIND. The BIND receiver can agree to the session parameters and return a positive response to BIND that is identical to the one received. If the BIND receiver wishes to negotiate, however, it returns a positive response to BIND that shows the changed parameters.

LU_T6 implementations need not contain all functions of the LU_T6 protocol. By using BIND SESSION, implementations can determine whether there is a subset of functions that allows meaningful communication between session partners.

HALF-SESSION CHARACTERISTICS

FM, TS, AND PS PROFILES

LU_T6 implementations use FM profile 18 and TS profile 4. The SNA requests that are allowed with each profile are found in SNA Reference Summary.

LU_T6 implementations use a PS profile that allows the following function management header and data stream options:

Function Management Headers (FMHs). LU_T6 implementations may use these FM headers:

- FMH-4 - This header is used to carry a logical block command and associated information for logical message services (LMS), an end-user service that can be implemented by TPSs. Logical message services uses LU-T6 sessions to transmit messages between end users.
- FMH-5 - This header is used to select a named transaction program within the receiving LU. FMH-5 flows from the transaction processing program using the sending half-session to the LU services manager for the receiving half-session. The transaction processing program must use FMH-5 to identify a named transaction program if the default destination at the receiving half-session is inappropriate or not provided.
- FMH-6 - This header is used to carry a command or data, or both, from an active transaction program attached to the sending half-session to an active transaction program attached to the receiving half-session.
- FMH-7 - This header is used to send further information about an error after negative response X'0846' has been received.
- FMH-10 - This header is used to start sync point processing. The LU services manager for the sending half-session sends the FMH-10 to the LU services manager for the receiving half-session.

Function management headers are discussed further in "Part 2 Chapter 4: Function Management Headers."

Data Streams. The session partner that sends the ATTACH FMH-5 selects the data stream. The data streams used by LU_T6 sessions are defined by the user.

TC (TRANSMISSION CONTROL) CONSIDERATIONS

Transmission Control services, such as pacing and cryptography, are summarized in SNA Concepts and Products and defined in SNA Format and Protocol Reference Manual: Architecture Logic.

DFC (DATA FLOW CONTROL) CONSIDERATIONS

Data flow control services, such as chaining, brackets, and send/receive modes, are summarized in SNA Concepts and Products and defined in SNA Format and Protocol Reference Manual: Architecture Logic. The following paragraphs give additional considerations.

SEND-RECEIVE MODES

LU_T6 implementations use half-duplex flip-flop. A half-session's reset state is either send (¬BETB.SEND.BETC with attributes S,¬R) or receive (¬BETB.RCV.BETC with attributes ¬S,R) when the bracket reset state is INB. A half-session's reset state is BETB.RCV.BETC with attributes ¬S,R when bracket reset state is BETB.

BRACKET PROTOCOL

LU_T6 implementations use the bracket protocol defined in SNA Format and Protocol Reference Manual: Architecture Logic. The BIND SESSION option that

allows neither half-session to end a bracket is invalid; that is byte 4 bit 7 and byte 5 bit 7 cannot both be B'0'. The BIND receiver can reject the BIND if optional bracket settings are not supported; it must support the required setting.

Transactions cannot span brackets.

CHANGE-DIRECTION MANAGEMENT

Use of CD Indicator (CDI): The change direction indicator (CDI) in the request header (RH) is used to change the direction of data flow when session partners are using half-duplex flip-flop mode and are in brackets (after BB and before EB). The session partner that has the authority to send at a given moment changes the direction of flow by turning on the CDI in the last RU it chooses to send.

The sending of the CDI should occur under the following circumstances:

- A transaction program has reached an end-of-send data-transition condition.
- A transaction program has received SIG X'0001' (request change direction). The procedures that must be followed when returning the CDI are discussed below under "Signal Codes."
- A transaction program has been session-activated in the send state, but has no data traffic to send.
- When the transaction program receives CD and it doesn't have any data to send.

A transaction program should send available FM data, if any, when the CDI is received.

Use of SIG (Signal): An LU that is capable of sending FM data may choose to send SIG X'0001' (request change direction) when in receive state with FM data to send.

The sender of SIG X'0001' is not required to return the CDI when it has no more data to send.

The receiver of SIG X'0001' should not send SIG X'0001' after sending the CDI until the end-chain indicator (ECI) has been returned. This prevents a signal ping/pong condition from occurring with no data flowing.

Data Direction Resolution: If brackets are used, the LU designated as the contention loser (BIND byte 7, bit 3) normally sends a BID request if it wants to enter send state, but it may enter send state by sending data to the session partner if that data is sent requiring a definite response (BB,RQD). The LU designated at the contention winner (first speaker) enters send state by sending data to the PLU.

PS (PRESENTATION SERVICES) CONSIDERATIONS

FM HEADER PROCESSING

After a session has been activated between two LUs, transaction processing programs can begin using the session. An ATTACH FMH-5 is sent between LUs before the sending transaction processing program sends data to the receiving transaction processing program. The ATTACH FMH-5 contains the name of the receiving program and other information needed to exchange data. LU_T6 implementations may choose to have a default destination if data is received without an ATTACH FMH-5.

Once the receiving program has been attached, other FM headers are valid if appropriate for the application or TPS.

DATA STREAM PROFILES

LU_T6 implementations specify the user-defined data stream profile in the ATTACH FMH-5.

SIGNAL CODES

SIG is an expedited request that may be sent between half-sessions regardless of the normal flows. It carries a 4-byte signal code set by the sending LU.

One signal code is valid for LU_T6:

X'0001' + 'rrrr' Request to send (0001) and reserved field (rrrr).

Signal codes that are not recognized or cannot be handled may be rejected with a negative response (sense code X'1003' function not supported).

LU STATUS (LUSTAT) CODES

Two LUSTAT codes must be received by LU_T6 half-sessions:

0006	Request header bit carrier
0007	Sender currently has no FM data requests to transmit

The format of LUSTAT is:

Byte	Value	Definition
0	X'04'	Request code
1-4		Status value and status extension field (two bytes each)
	X'0006'+'rrrr'	No-op except to allow an RH to be sent when no other request is available or allowed (0006) and reserved field (rrrr)
	X'0007'+'rrrr'	Sender currently has no FM data to transmit (0007) and reserved field (rrrr)

LU_T6 allows BB (begin bracket) on LUSTAT.

SYNC POINTS

LU_T6 implementations provide a protocol for coordinating events between two cooperating transaction processing programs. Under this protocol, the programs establish periodic synchronization points. When a synchronization point is reached, both TPSs are notified that all work since the last synchronization point has been successfully completed. If the conversation between the two transaction processing programs is disrupted before a synchronization point is reached, both TPSs cancel data base changes made since the previous synchronization point, so that files used by both transaction processing programs will be at the same level.

ERRORS

All errors are turned into exception requests (EXRs) by the element of the network that detects the error. The exception request is sent forward (except for loss of path errors) to the destination identified by the DSAF and DAF (destination subarea field and destination address field of the tran-

smission header). The destination then processes the EXR and returns a negative response if the request was marked definite or exception response requested.

ERP TRANSITION

An error may be detected while a half-session is sender or receiver and while multiple requests are outstanding. To ensure an orderly transition to ERP state by both half-sessions, the following rules are used.

Note: There are several levels of synchronization on an LU_T6 session. The level discussed here is the synchronizing of RU chains. Examples of other levels are brackets synchronization and transaction program unit-of-work synchronization.

For sender detected errors:

1. The session partners may already be synchronized. This condition occurs if no RU chains have been sent since the receipt of CD or if a positive response has been received to the last chain sent. If the last chain was sent RQD, then synchronization is achieved by waiting for the response.
2. If the last chain was sent RQE, then CHASE is sent to achieve synchronization.
3. If the current chain is incomplete, then CANCEL is sent to achieve synchronization.

For receiver detected errors:

1. If there is an outstanding RU chain, then the receiver sends negative response X'0846' unless there is an inescapable transition to BETB (BB,EB or EB,RQE chain), in which case a more informative code may be used. (See "Sense Codes and Sense Data" on page 118).
2. If there is no outstanding chain (error detected subsequent to response), then the receiver sends negative response X'0846' to the next chain. This action avoids compounding the error situation by attempting to use Signal (RCD).
3. Once a negative response has been sent, RQE chains are purged to achieve synchronization. Each nonsynchronizing chain is discarded without response. Responses to the possible synchronizing events are:

Synchronizing event	Response
RQE,CD	-RSP.0867
RQD FM data	-RSP.0867
DFC(LUSTAT),RQD	-RSP.0867
DFC(¬LUSTAT),RQD	+RSP

Negative response X'0867' is used to reduce the possibility of a transition to BETB with resulting loss of ERP coordination.

LUSTAT is the only DFC request that allows a negative response. The positive response to DFC(¬LUSTAT) signifies to both half-sessions that purge state has been exited and that the ERP message can flow.

Note: The possibility of asynchronously detected errors leads to the possibility of contention for ERP.SEND. This condition is manifested by the receipt of negative response X'0846' while purging, and this response overrides the current purging operation.

SENDER ERP BUT WITH SELECTIVE RECEIVER ERP

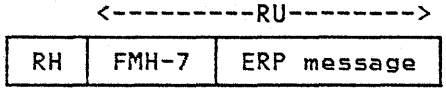
LU_T6 sessions require that the session partners use sender ERP; that is, the sender of data is responsible for error recovery of that data. Under this protocol, the receiver remains in receive state after it sends a negative response unless it entered BETB state. The sender of data then attempts to correct the error and synchronize the session.

There are times when the receiver of data can provide valuable information about the error, but it cannot contain the information within the sense data field of the negative response. When this condition occurs, sense code X'0846' is returned with the negative response. This alternative allows the data receiver (the sender of the negative response) to choose when to change the normal send-receive (S,R) transitions.

If the data receiver returns negative response X'0846', it must follow the response with an ERP message.

ERP Messages

ERP messages are requests used by the session partners to define the cause of errors and to provide corrective information. The request unit that carries the ERP message contains an FH header type 7 (FMH-7) and the ERP message, which can be any information understood by the session partners:



The traditional SNA method of notifying the session partner of an error is the negative response returned by the receiver when an error is detected. Within LU_T6 sessions, the ERP message provides session partners with an alternative protocol.

SENSE CODES AND SENSE DATA

Sense code values for LU_T6 sessions are:

CHAIN ERRORS:

RH usage errors: X'40'

4003	BB not allowed
4009	CD not allowed

Request errors: X'10'

10030000	Function not supported
10036002	Invalid destination program name (DPN)
10036003	Invalid primary resource name (PRN)
10080000	Invalid FM header
1008200E	Invalid concatenation
10086001	Invalid deblocking algorithm (DBA)
10086004	Invalid queue name length
10086006	Invalid data stream profile (DSP)
10086007	FMH-7 not preceded by negative response X'0846'
10086008	Invalid attach access code
10086009	FMH-5 fixed length parameter count not equal to 2
1008600A	Not first FMH-5 and IUT ¬= IUT in register and IUE ¬= on
1008600B	FMH-5 command invalid
1008600C	Null sequence field required
1008600D	User to user program transition not allowed
1008600E	User to architected program transition not allowed
1008600F	RAP FMH-5 not sent properly
10086010	RAP FMH-5 sent with inactive attach register

Request Reject: X'08'

080F	End user not authorized
0812	Insufficient resource
0813	Bracket BID reject -- no RTR forthcoming
0814	Bracket BID reject - RTR forthcoming
0819	RTR not required

081C	Request not executable
0826	FM function not supported
0829	Change direction required
084B0000	Requested resource not available
084B6002	Requested DPN resource unavailable
084B6003	Requested PRN resource unavailable

UNIT OF WORK ERRORS: Unit of work errors are request reject errors (category code X'08'):

0824	Unit of work aborted
0846	ERP message forthcoming
0864	Function abort -- loop will occur upon reexecution
0865	Function abort -- sender responsible to detect loop
0866	Function abort -- receiver responsible to detect loop

X'0846' is used to cause a transition from sender ERP to receiver ERP.

BIND SESSION FORMAT FOR LU-LU SESSION TYPE 6

This section defines valid bit settings for the BIND request. The BIND for LU_T6 can be sent as a negotiable or nonnegotiable request. In a negotiable BIND, the BIND sender and BIND receiver can negotiate to establish session protocols. In the nonnegotiable BIND, the BIND sender specifies the session parameters.

The formats of both the nonnegotiable BIND and the negotiable BIND request and response are identical. Differences in protocols are noted in the field definitions.

BIND SESSION REQUEST

This section defines valid bit settings for the BIND request. All values listed must be supported, unless the value is optional (identified by P in the Opt column for the BIND sender and by S in the Opt column for the BIND receiver) or not valid.

Byte	Bits	Value	Opt	Discussion
0	0-7	X'31'		Identifies this RU as a BIND request.
1	0-3	X'0'		Specifies the format of the BIND RU. Only one format has been defined: format 0 (zero).
(1)	4-7			Type - denotes the type of BIND to be performed:
		X'0'		Negotiable BIND (cold start)
		X'1'		Nonnegotiable BIND (cold start)

FM (Function Management) Profile

2	0-7	X'12'		FM profile 18
				The FM profile defines data flow control (DFC) protocols. (See SNA Reference Summary for highlights of the FM profile.)

TS (Transmission Subsystem) Profile

3	0-7	X'04'		TS profile 4
				The TS profile defines transmission control (TC) protocols. (See SNA Reference Summary for highlights of the TS profile.)

FM Usage

(4-7)				See bytes 4 through 7 below.

BIND Sender's Send Protocols

				Chaining Use:
4	0	B'0'		Not valid
		B'1'		BIND sender can send single or multiple element chains.

Byte	Bits	Value	Opt	Discussion
				Request Mode Selection:
(4)	1	B'0'		Immediate request mode is used. Only one definite response can be outstanding at a time. That response must be received before the BIND sender can send another RU.
		B'1'	P	Delayed request mode is used. Multiple definite responses can be outstanding at one time.
				Note: If delayed request mode is used, the RU sender must be responsible for recovery, and EC (end chain) may not be sent on the current chain when a definite response is outstanding for the previous chain.
				Chaining Responses:
(4)	2-3	B'00'		Not valid
		B'01'		Not valid
		B'10'	P	BIND sender can request only definite responses.
		B'11'		BIND sender can request definite or exception responses.
				Two-phase Commit Indicator:
(4)	4	B'0'		Two-phase commit not supported (cannot send PREPARE FMH-10).
		B'1'	P	Two-phase commit supported (can send PREPARE FMH-10).
(4)	5			Reserved
				Compression Indicator:
(4)	6	B'0'		BIND sender cannot send compressed data.
		B'1'		Not valid.
				Send End Bracket Indicator:
(4)	7	B'0'		BIND sender will not send EB (see byte 6 bit 2).
		B'1'		BIND sender can send EB.
				Note: First speaker must be able to send EB (see byte 7 bit 3). If BIND sender is first speaker, bit 7 must be 1.

BIND Receiver's Send Protocols

				Chaining Use:
5	0	B'0'		Not valid
		B'1'		BIND receiver can send single or multiple element chains.
				Request Mode Selection:
(5)	1	B'0'		Immediate request mode is used. BIND receiver can issue a request for a single definite response. No further transmission is sent until the BIND receiver receives the requested response.
		B'1'	S	Delayed request mode is used. BIND receiver allows several definite responses to be outstanding at one time.
				Note: If delayed request mode is used, the RU sender must be responsible for recovery.
				Chaining Responses:
(5)	2-3	B'00'		Not valid
		B'01'		Not valid
		B'10'	S	BIND receiver can request only definite responses.
		B'11'		BIND receiver can request either definite or exception responses.

Byte	Bits	Value	Opt	Discussion
				Two-phase Commit Indicator:
(5)	4	B'0'		Two-phase commit not supported.
		B'1'	S	Two-phase commit supported.
(5)	5			Reserved
				Compression Indicator:
(5)	6	B'0'		BIND receiver cannot send compressed data.
		B'1'		Not valid.
				Send End Bracket Indicator:
(5)	7	B'0'	S	BIND receiver will not send EB (see byte 6 bit 2).
		B'1'		BIND receiver can send EB.
				Note: First speaker must be able to send EB (see byte 7 bit 3). If BIND receiver is first speaker, bit 7 must be 1.

Common Protocols for FM Data

6	0		Reserved
(6)	1	B'0'	Not valid.
		B'1'	BIND sender and BIND receiver can exchange FM headers.
			Brackets Usage and Reset State:
			Note: Brackets are required in LU_T6 sessions.
(6)	2	B'0'	The bracket state manager's reset state is INB (in brackets).
		B'1'	The bracket state manager's reset state is BETB (between brackets).
			Bracket Termination Rule Selection:
(6)	3	B'0'	Not valid
		B'1'	Bracket termination rule 1 is used.
			Note: Bit 3 is reserved if brackets are not used by the session.
			Alternate Code Selection:
(6)	4	B'0'	Alternate code set may not be used.
		B'1'	Not valid.
			Sequence Number Usage for Resynchronization:
(6)	5	B'0'	Sequence numbers not available.
		B'1'	Sequence numbers available (from the previous activation of the session with the same session name) for sync point resynchronization.
			Bracket State Status for Resynchronization:
(6)	6	B'0'	BIS request not sent prior to session termination.
		B'1'	BIS request sent prior to session termination (from the previous activation of the session with the same session name) so sequence numbers are not required for resynchronization.
(6)	7		Reserved

Byte	Bits	Value	Opt	Discussion
				Normal-Flow Send and Receive Mode Selection:
7	0-1	B'00'		Not valid
		B'01'		Not valid
		B'10'		Half-duplex flip-flop
		B'11'		Reserved
				Recovery Responsibility:
(7)	2	B'0'		Not valid.
		B'1'		Symmetric responsibility for recovery. The sender of data is responsible for recovery.
				Contention Winner/Loser:
(7)	3	B'0'		BIND receiver is contention winner (first speaker), and BIND sender is contention loser.
		B'1'	PS	BIND sender is contention winner (first speaker), and BIND receiver is contention loser.
(7)	4-6			Reserved
				Half-Duplex Flip-Flop (HDX-FF) Reset States:
				Note: Byte 7 bit 7 is reserved if bracket state manager reset is BETB; it is used only when reset is INB (see byte 6 bit 2).
(7)	7	B'0'		HDX-FF reset state is RECEIVE for the BIND sender and SEND for the BIND receiver (that is, the BIND receiver sends normal-flow requests first after session activation or reset).
		B'1'		HDX-FF reset state is SEND for the BIND sender and RECEIVE for the BIND receiver (that is, the BIND sender sends normal-flow requests first after session activation or reset).

TS Usage

(8-13)				See bytes 8 through 13 below.
				Staging Indicator and Pacing Count for BIND-receiver CPMGR (connection point manager) to BIND-sender CPMGR normal flow:
8	0	B'0'		Pacing in this direction occurs in one stage.
		B'1'		Pacing in this direction occurs in two stages.
				Note 1: When the session involves a boundary function, pacing may occur in two stages. (For the definition of a boundary function, see SNA Format and Protocol Reference Manual: Architecture Logic.)
				Note 2: The meanings of 0 and 1 are reversed from the staging indicator for BIND-sender CPMGR to BIND-receiver CPMGR (see byte 12).
(8)	1			Reserved
(8)	2-7	B'nnnnnn'		BIND-receiver CPMGR's send pacing count. A value of 000000 means no pacing of requests flowing from the BIND receiver.
9	0-1			Reserved
(9)	2-7	B'nnnnnn'		BIND-receiver CPMGR's receive pacing count. A value of 000000 causes the boundary function to substitute the value set by a system definition pacing parameter (if the system definition includes such a parameter) before it sends the BIND RU on to the BIND receiver. A value of 000000 received at the BIND receiver is interpreted to mean no pacing of requests flowing to the BIND receiver.

Byte	Bits	Value	Opt	Discussion
				Maximum RU Size:
10		X'85'		Maximum RU size of 256 bytes sent on the normal flow by the BIND receiver.
		X'ab'	P	Maximum RU size greater than or less than 256 bytes.
				This value represents the largest RU that can be sent. It is expressed as a mantissa and an exponent value of 2 by which the mantissa is multiplied. For example, when the mantissa is 8 and the exponent is 5 (X'85'), the RU size is 256 bytes (8×2^5).
				When bit 0 is set to zero, no maximum is specified, and the remaining bits 1-7 are ignored. When bit 0 is set to one, the byte is interpreted as X'ab'. (See Appendix B for these values.)
11		X'85'		Maximum RU size of 256 bytes sent on the normal flow by the BIND sender.
		X'ab'	S	Maximum RU size greater than or less than 256 bytes.
				This value represents the largest RU that can be sent by the BIND sender and is specified in the same format as the BIND receiver (byte 10).
				Staging Indicator and Pacing Count for BIND-sender CPMGR to BIND-receiver CPMGR normal flow:
12	0	B'0'		Pacing in this direction occurs in two stages.
		B'1'		Pacing in this direction occurs in one stage.
				Note: The meanings of 0 and 1 are reversed from the staging indicator for BIND-receiver CPMGR to BIND-sender CPMGR (see byte 8, notes 1 and 2).
(12)	1			Reserved
(12)	2-7	B'nnnnnn'		BIND-sender CPMGR's (connection point manager's) send pacing count. A value of 000000 means no pacing of requests flowing from the BIND sender.
				For single-stage pacing in the BIND sender to BIND receiver direction, this field is redundant with, and will indicate the same value as, the BIND-receiver CPMGR's receive pacing count (see byte 9, bits 2-7 above).
13	0-1			Reserved
(13)	2-7	B'nnnnnn'		BIND-sender CPMGR's receive pacing count. A value of 000000 means no pacing of requests that flow to the BIND sender. For single-stage pacing in the BIND receiver to BIND sender direction, this field is redundant with, and will indicate the same value as, the BIND-receiver CPMGR's send pacing count (see byte 8, bits 2-7 above).

PS Profile

(14-25)				See bytes 14 through 25 below.
14	0	B'0'		PS usage field format is the basic format.
		B'1'		Reserved
(14)	1-7	B'0000110'		LU-LU session type 6

Byte	Bits	Value	Opt	Discussion
PS Usage				
15				Reserved
16-19				BIND Sender Flags:
16	0-1			Reserved
	2	B'0'		System message transaction program not supported
		B'1'	P	System message transaction program supported
	3	B'0'		Scheduler transaction program not supported
		B'1'	P	Scheduler transaction program supported
	4	B'0'		Queue transaction program not supported
		B'1'	P	Queue transaction program supported
	5			Reserved
	6	B'0'		DL/1 transaction program not supported
		B'1'	P	DL/1 transaction program supported
	7			Reserved
17-19				Reserved
20-23				BIND Receiver Flags:
20	0-1			Reserved
	2	B'0'		System message transaction program not supported
		B'1'	S	System message transaction program supported
	3	B'0'		Scheduler transaction program not supported
		B'1'	S	Scheduler transaction program supported
	4	B'0'		Queue transaction program not supported
		B'1'	S	Queue transaction program supported
	5			Reserved
	6	B'0'		DL/1 transaction program not supported
		B'1'	S	DL/1 transaction program supported
	7			Reserved
21-23				Reserved
24-25				Reserved

Byte	Bits	Value	Opt	Discussion
Cryptography Options				
26	0-1			Private cryptography options:
		B'00'	PS	No private cryptography supported.
		B'01'	PS	Private cryptography supported. The session cryptography key and cryptography protocols are privately supplied by the end user.
		B'10'		Reserved
		B'11'		Reserved
(26)	2-3			Session-level cryptography options:
		B'00'	PS	No session-level cryptography supported.
		B'01'	PS	Session-level selective cryptography supported. All cryptography key management is supported by the SSCP and LU; exchange (using +RSP to BIND) and verification (using CRV) of the cryptography session-seed value are supported by the LUs for the session. All FM data requests with EDI (enciphered data indicator) on are enciphered/deciphered.
		B'10'		Reserved
		B'11'	PS	Session-level mandatory cryptography supported; same as session-level selective cryptography except <u>all</u> FM data requests are enciphered/deciphered.
(26)	4-7			Session-level cryptography options field length:
		X'0'	PS	No session-level cryptography specified. The cryptography option fields (bytes 27-k) are omitted.

Byte	Bits	Value	Opt	Discussion
		X'9'	PS	Session-level cryptography specified. Additional options follow in the next nine bytes.
27	0-1			Session cryptography key encipherment mode:
		B'00'	PS	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key using a seed value of zero (only value defined).
(27)	2-4			Reserved
(27)	5-7			Cryptography cipher method:
		B'000'	PS	Block chaining with seed and cipher text feedback, using the Data Encryption Standard (DES) algorithm (only value defined).
28-k			PS	Session cryptography key enciphered under BIND receiver (or SLU) master cryptography key; an eight-byte value that, when deciphered, yields the session cryptography key used for enciphering and deciphering FM data requests.

Names and End User Data

k+1			Length of BIND sender's name in field k+2-m.
	X'00'		Field k+2-m does not exist.
	X'nn'		Field k+2-m contains the number of bytes specified (cannot exceed X'08').
k+2-m			BIND sender's LU name.
m+1			Length of user data in field m+2-n.
	X'00'		Field m+2-n does not exist.
	X'nn'		Field m+2-n contains the number of bytes specified.
m+2-n		PS	User data
m+2		PS	User data key.
	X'00'	PS	Structured subfields follow.
	¬X'00'	PS	First byte of unstructured user data.
			Note: The unstructured user data continues in m+3 and goes to byte n. (For information on structured subfields, see the BIND information in SNA Format and Protocol Reference Manual: Architecture Logic.)
n+1			Length of user request correlation field.
	X'00'		Field n+2-p does not exist.
	X'nn'		Field n+2-p contains the number of bytes specified.
n+2-p			User request correlation field.
p+1			Length of BIND receiver's name in field p+2-r.
	X'00'		Field p+2-r does not exist.
	X'nn'		Field p+2-r contains the number of bytes specified (cannot exceed X'08').
p+2-r			BIND receiver's LU name.

Notes

The receiver of a nonnegotiable BIND request cannot return a negative response because a mandatory value is specified, but may return a negative response when an optional value is specified and the BIND receiver doesn't support it.

The receiver of a negotiable BIND request may negotiate only the changing of an optional value to a mandatory value. It may unbind the session when an optional value is specified and the BIND receiver does not support it.

When preparing a negotiable BIND response, the BIND receiver should indicate the PS profile options it actually supports (in bytes 20-23) and the options that the BIND sender should support (in bytes 16-19).

This chapter defines the controls of the SNA character string used by LU-LU sessions.

SNA CHARACTER STRING CONTROLS

SNA character string (SCS) controls are EBCDIC control codes that may be used to define a data stream. Their primary function is to format a visual presentation medium, such as a printed page or a display screen. They also set modes of device operation, define data to be used in a unique fashion, or are used for communication between a device operator and an application program (where the specific function associated with the code is defined in a protocol established between a program and an operator).

Data streams based on SCS controls consists of a sequential string of SCS controls and data characters. These data streams are defined by the LU-LU session types that use SCS controls. This chapter defines the SCS controls that may be used to build these data streams, but no sets of controls are identified.

Controls may be intermixed with graphic data characters. SCS control codes are in the range X'00' through X'3F' plus X'FF'. Graphic codes are in the range X'40' through X'FE'. Other data types (such as binary and packed decimal) are permitted, but only in conjunction with the Transparent (TRN), Presentation Position (PP), Program Operator Communication (POC), and Control Sequence Prefix (CSP) controls, discussed later. One-byte parameters that specify functions or binary values are permitted with some codes.

SCS control codes and data appear within the RU portion of the basic information unit (BIU). They may be preceded or separated by other control information in the RU, such as function management headers (FMHs) and string control bytes (SCBs) for functions such as destination selection, data management, and compression or compaction.

SCS functions do not include data flow control functions, even though both may be available to a keyboard operator through keys on the keyboard. CANCEL, for example, is a data flow control request which may be initiated by a key on the keyboard.

SCS functions include certain unique graphic characters that contain control functions. (See "Graphic Codes that Possess Control Characteristics," "Adjust Text Mode," and "Text Justify Mode" later in this chapter.)

An SCS control and parameter sequence may span RUs; an SCS control code and parameter sequence may not span RU chains. A receiver of these sequences must terminate any SCS parameter updating in process upon receiving a first-in-chain or only-in-chain RU. Expected parameters that are not received are to be interpreted as invalid parameters within the chain containing the start of the SCS control code.

Parameters in SCS controls are of two types: function and value. Function parameters are used to extend the function defined by the SCS control. For example, the PP control has a function parameter to define explicitly the positioning function to be performed. Value parameters are used to specify a numeric value associated with the SCS control. The PP control also has a value parameter associated with it. If the move is relative to the current position, the value parameter specifies the number of columns or lines the presentation position is to be moved from its current position. If the move is absolute, the value parameter specifies the absolute column or line number to which the presentation position is to move.

Function parameters are EBCDIC characters; value parameters are binary numbers or graphic symbols.

SCS control functions are assigned EBCDIC codes as shown below. The control function ID is a one-byte or multiple-byte field (for example, 0D for carriage return and 2BC6 for set line density).

Code (EBCDIC)	SCS Control Function	Abbreviation
16	Backspace	BS
2F	Bell (Stop)	BEL (STP)
0D	Carriage Return	CR
1B	Customer Use 1	CU1
3B	Customer Use 3	CU3
11	Device Control 1	DC1
12	Device Control 2	DC2
13	Device Control 3	DC3
3C	Device Control 4	DC4
14	Enable Presentation	ENP
36	Expanded Backspace (Numeric Backspace)	EBS (NBS)
E1	Expanded Space (Numeric Space)	ESP (NSP) *
0C	Form Feed (Page End)	FF (PE)
08...	Graphic Escape	GE
05	Horizontal Tab	HT
39	Indent Tab	IT
33	Index Return	IR
24	Inhibit Presentation	INP
1C	Interchange File Separator	IFS
1D	Interchange Group Separator	IGS
1E	Interchange Record Separator	IRS
1F	Interchange Unit Separator	IUS
25	Line Feed (Index)	LF (INX)
15	New Line (Carrier Return)	NL (CRE)
00	Null	NUL
34...	Presentation Position	PP
17...	Program Operator Communication	POC
0A	Repeat	RPT
3A	Required Form Feed (Required Page End)	RFF (RPE)
06	Required New Line (Required Carrier Return)	RNL (RCR)
41	Required Space	RSP *
0450	Secure String ID Reader	SSR
04C1	Select Left Platen	SLP
046n	Select Magnetic Encoder	SME
04C2	Select Right Platen	SRP
28...	Set Attribute	SA
2BD1 Cn	Set Chain Image	SCI
2BC8	Set Graphic Error Action	SGEA
2BC1	Set Horizontal Format	SHF
2BC6	Set Line Density	SLD
2BD2 29	Set Print Density	SPD
2BD1 Cn	Set Translation Table	STT
2BC2	Set Vertical Format	SVF
0F	Shift In	SI
0E	Shift Out	SO
2BC3	Start of Format	SOF
38	Subscript	SBS
3F	Substitute	SUB
09	Superscript	SPS
2A	Switch	SW
CA	Syllable Hyphen	SHY *
35...	Transparent	TRN
1A	Unit Backspace	UBS
04...	Vertical Channel Select	VCS
0B	Vertical Tab	VT
23	Word Underscore	WUS

Note: Functions with ellipses extending their 1-byte code (those that have the expression "..." following them) have one or more parameters and are multiple-character code points.

* This function uses a graphic code. (See "Graphic Codes that Possess Control Characteristics" on page 150.)

CONTROL FUNCTION DESCRIPTIONS

This section describes, in alphabetical order, the SCS controls that may be used to transmit a data stream between points in a network.

The formatting controls are based on a presentation surface, which is mapped onto a device. The presentation surface consists of a matrix of character positions with an origin of column 1 and line 1. The maximum presentation surface width and depth may be the fixed parameters of the device; the top margin, bottom margin, left margin, and right margin are variable parameters (Figure 1-1). The variable parameters have default values specified later.

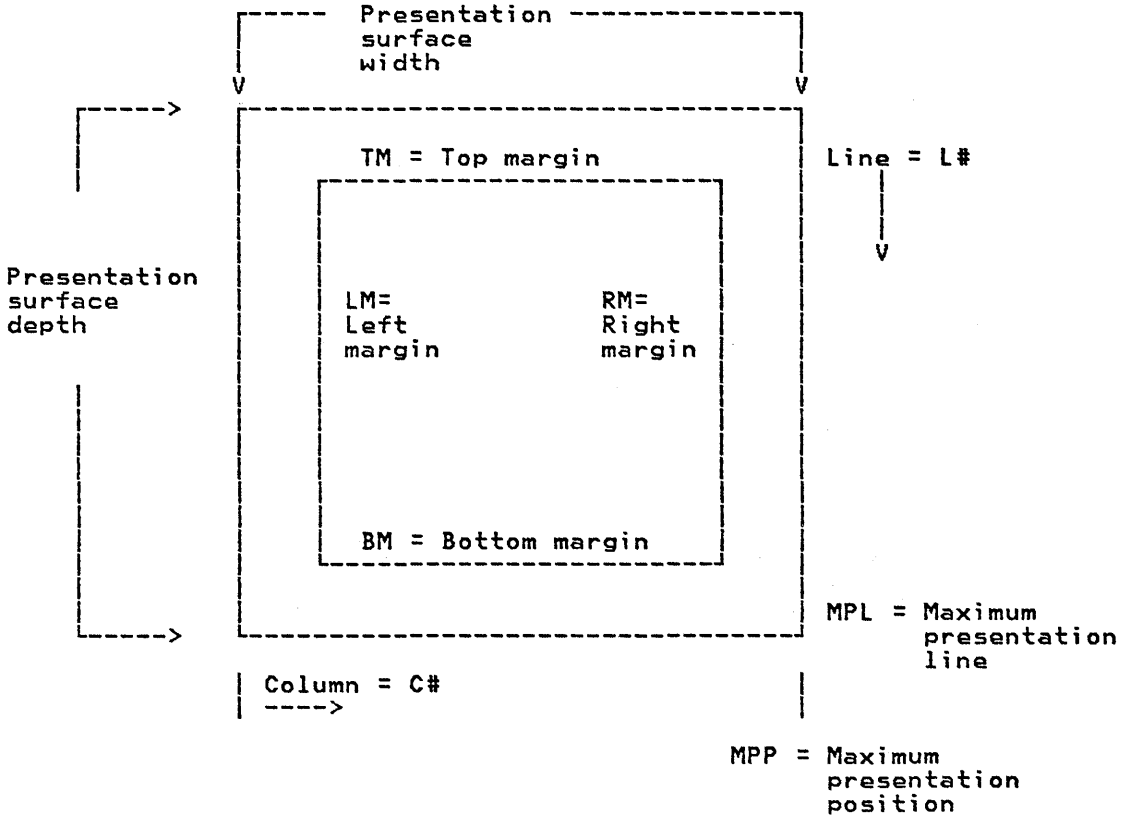


Figure 1-1. The presentation surface. TM, BM, and MPL are expressed by a number whose origin is line number one (the topmost line of the presentation surface). LM, RM, and MPP are expressed by a number whose origin is column number one (the leftmost column of the presentation surface).

The presentation position is defined by a pair of coordinates (L#,C#) that specify the line number and column number where the next graphic character will be displayed. The presentation position is the print head position on a serial printer, the next print position on a line printer, and so on. All characters are of a uniform size and all lines are uniformly spaced.

A console consists of a presentation surface and a keyboard that are tied together by sharing a common buffer. When the keyboard enters characters into the buffer, they are also printed or displayed. Exceptions to this rule exist as noted under those SCS controls that are affected.

The presentation surface is mapped onto a printer by making each page equal to a presentation surface, where the maximum presentation line (MPL), or page depth, is set to accommodate the printer form size.

The physical positioning by the device is the same, whether the source of the control function is the received data stream or key action. When the source

is the keyboard, the control code is also inserted in the data stream being generated.

The control definitions are presented in two stages. The first stage consists of the essential features within each control function; these are given below. The second stage augments the individual control definitions with generally applicable definitions for automatic line, column, and presentation surface advancement.

BACKSPACE (BS, X'16')

BS is a formatting control that moves the presentation position horizontally one position to the left. If the presentation position is at column one, the function becomes a no-op. Left margin settings are ignored.

Using the (L#,C#) pair previously defined:

```
C#(new)<----maximum(1,C#(old)-1)
L#(new)<----L#(old)
```

This function is for overstrike purposes and not for error correction. Backspace for error correction is a local device function, and requires a separate key that does not generate a code for transmission from the device.

Differences can occur in the output if the presentation position is at the column one position when the backspace code is received. (See "Unexpected Output with Contiguous Backspaces" on page 154.)

BELL (BEL, X'2F')

BEL is a program-to-operator communication control used in the same fashion as POC. The difference is that BEL has no parameter associated with the control. For program-to-operator communication, BEL may perform a stop function. This requires the device to stop processing the data stream until the operator action associated with BEL is performed.

CARRIAGE RETURN (CR, X'0D')

CR is a formatting control that moves the presentation position horizontally to the left margin on the same line. If the current position already equals the left margin, the function becomes a no-op.

```
L#(new)<----L#(old)
C#(new)<----LM
```

CUSTOMER USE (CU)

CUSTOMER USE 1 (CU1, X'1B')
CUSTOMER USE 3 (CU3, X'3B')

CU1 and CU3 are program-to-operator communication controls used in the same fashion as POC. The difference is that there are no parameters associated with the CU1 and CU3 controls.

DEVICE CONTROL (DC)

```
DEVICE CONTROL 1 (DC1, X'11')
DEVICE CONTROL 2 (DC2, X'12')
DEVICE CONTROL 3 (DC3, X'13')
DEVICE CONTROL 4 (DC4, X'3C')
```

DC1, DC2, DC3, and DC4 are device controls for device-specific control functions. For example, when data is directed to a magnetic stripe writer, DC1 identifies a test record to be written on the magnetic stripe.

ENABLE PRESENTATION (ENP, X'14')

ENP is a device mode control that enables the presentation of device-entered data at the entering device. All data entered at the device after receipt of the ENP control function is presented in the normal fashion. The ENP control function is used in conjunction with the INP control function to control presentation during the entry of sensitive data.

EXPANDED BACKSPACE (EBS, X'36')

EBS is a formatting control that provides a large, fixed-size backspace used primarily for tabular and columnar formatting on proportional-space devices. The normal backspace on proportional-space devices does not provide this function because it is a variable-sized backspace based on the previous character. On nonproportional devices, EBS may be folded into BS.

EXPANDED SPACE

See "Graphic Codes that Possess Control Characteristics" on page 150.

FORM FEED (FF, X'0C')

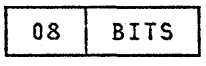
FF is a formatting control that moves the presentation position to the top and left margins of the next presentation surface. If the MPL parameter has not been set (and no device-specific default exists), then it defaults to one; positioning is to the left margin of the next line.

```
L#(new)<----TM(next presentation surface)
C#(new)<----LM(next presentation surface)
```

GRAPHIC ESCAPE (GE, X'08...')

GE is a control that indicates that the bit-pattern (graphic) that follows shall be interpreted according to the most recently designated graphic escape page. GE appears as:

GE,GRAPHIC



where:

BITS Is a bit pattern constructed within one or more bytes.

HORIZONTAL TAB (HT, X'05')

HT is a formatting control that moves the presentation position horizontally to the right to the next tab stop setting. Horizontal tab stop values may be set using SHF (set horizontal format). If no horizontal tab stops are set to the right of the current presentation position, HT results in a space.

```
L#(new)<----L#(old)
If HTV(any)> C#(old)
Then C#(new)<-----HTV(next)
Else C#(new)<-----C#(old)+1
```

where:

```
HTV = Horizontal tab stop value expressed as a column number.
HTV(next) = Smallest HTV>C#(old)
```

INDENT TAB (IT, X'39')

IT is a mode-setting control character that requires a device to generate one or more automatic horizontal tab operations after each new line is keyed or NL (new line) is seen on the outbound character string. IT has the effect of temporarily indenting the left margin. The number of automatic tabs performed is equal to the number of ITs inserted from the keyboard or seen in the outbound character string. The RNL (required new line) or IR (index return) control cancels the IT setting, or additional IT codes cause the tab setting to be moved farther to the right on the device. The number of ITs allowed is equal to the number of tab settings allowed.

In indent tab mode, the device automatically tabs after each NL control that is present between IT codes and the RNL control. This tabbing is accomplished without the presence of tab codes within the outbound character string. Indent tab mode does not preclude the use of embedded HT codes for additional tabbing if required. IT is treated as a no-op by half-sessions that do not support the adjust text function.

Note: The text adjusting process automatically ends one line at approximately the right margin and starts the next line at the left margin. When operating in the middle of an indented paragraph, IT starts the new line at the proper tab location rather than at the left margin.

INDEX RETURN (IR, X'33')

IR is a line-ending control character that is both a formatting control and a grouping control.

As a formatting control, IR causes a mandatory movement of the print position to the left margin of the next line and resets indent tab mode. IR is similar to the RNL control, except that it may be used with magnetic media in such a way that the next line of display data may appear immediately following the IR code in the same magnetic card track.

As a grouping control, IR is used to delimit the end of a field of a record and the end of a line on a magnetic disk medium.

INHIBIT PRESENTATION (INP, X'24')

INP is a device mode control that inhibits the presentation of keyboard-entered data at the entering device, while allowing the entered data to be transmitted in the normal fashion. Presentation of all data entered at the device after receipt of INP is inhibited. Data received by the device is presented, not inhibited. In either case, the presentation position is moved in the normal fashion. INP is used in conjunction with ENP (described above) to control presentation during the entry of sensitive data.

INTERCHANGE SEPARATORS (IUS, X'1F'; IRS, X'1E'; IGS, X'1D'; IFS, X'1C')

The interchange separators consist of IUS (interchange unit separator), IRS (interchange record separator), IGS (interchange group separator), and IFS (interchange file separator); they are used as logical separators. When used in hierarchical order, the ascending order is: IUS, IRS, IGS, IFS. That is, IUS codes separate the smallest units of data, while IFS codes separate the largest.

An information block must not be split by a higher order separator; for example, a record may contain a whole number of units, but may not contain a part of a unit.

Within SCS, IRS has additional uses as follows:

- Terminates a secure data string headed by an SSR control
- Terminates a start-of-format data string headed by an SOF control
- Indicates the end of a card in a data stream sent to a card punch or received from a card reader

LINE FEED (LF, X'25')

LF is a formatting control that moves the presentation position vertically down to the next line. The column position remains unchanged.

```
L#(new)<---L#(old)+1
C#(new)<---C#(old)
```

NEW LINE (NL, X'15')

NL is a formatting control that moves the presentation position horizontally to the left margin, and vertically down to the next line.

NL is functionally equivalent to CR followed by LF.

```
L#(new)<----L#(old)+1
C#(new)<----LM
```

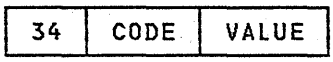
NULL (NUL, X'00')

NUL is a control character that may be used as a time or media fill. On the presentation surface, NUL is treated as a no-op.

PRESENTATION POSITION (PP, X'34...')

PP is a formatting control that moves the presentation position according to two parameter types: function and value. PP appears as:

PP,FUNCTION-CODE,VALUE



where:

The PP code (X'34') is followed first by the function parameter. The value parameter follows the function parameter; it is a 1-byte binary number and denotes either an absolute or relative column or line number.

There are four operations defined by the function parameter:

Absolute move or relative move
Horizontal move or vertical move
Move and erase or move without erasing
Erase to new position, then reset to old position

Note: To erase a character at a position means to replace it with a device dependent character, generally a null or a space.

Codes are assigned to allow combinations of these functions (see table, following). Erase functions are intended for display devices. Value parameters for absolute moves can be equal to or less than the MPL or MPP, as appropriate and, for relative moves, not past the end of the line or page. A zero value is a no-op. An error is indicated if any parameter is not valid.

When an absolute move is specified, the value parameter denotes a specific line or column number, and the presentation position is moved to the line or column specified. For a printer, an absolute move to a line number less than the current line is equal to a move to that line on the following page.

When a relative move is specified, the value parameter denotes a positive incremental value; the presentation position is moved the number of line or column increments specified.

Function codes not shown in the following table are reserved.

Function Parameter		Code
Absolute, Horizontal	Move PP and do not erase	X'C0'
Absolute, Horizontal	Erase line to new PP, then reset to old PP	X'C1'
Absolute, Horizontal	Erase line to and move to new PP	X'C2'
Absolute, Vertical	Move PP and do not erase	X'C4'
Absolute, Vertical	Erase column to new PP, then reset to old PP	X'C5'
Absolute, Vertical	Erase column to new PP and move to new PP	X'C6'
Relative, Horizontal	Move PP and do not erase	X'C8'
Relative, Horizontal	Erase line through new PP, then reset to old PP	X'C9'
Relative, Horizontal	Erase line through, and move to, new PP	X'4A'
Relative, Vertical	Move PP and do not erase	X'4C'
Relative, Vertical	Erase column through new PP, then reset to old PP	X'4D'
Relative, Vertical	Erase column through new PP and move to new PP	X'4E'

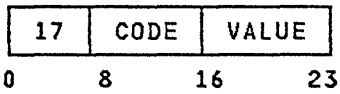
An absolute move with erase does not erase the new PP.

PROGRAM OPERATOR COMMUNICATION (POC, X'17...')

POC is a program-to-operator communication control that provides communication between end users, where at least one of the end users is a terminal operator. The communication may be from a program to an operator, from an operator to a program, or from an operator to an operator.

Typically, the program receives a POC sequence identifying one of several program function keys that an operator has activated; or the program sends a POC sequence that turns on one or more indicator lights visible to the operator. The meaning associated with POC character sequences is defined by the end user. POC appears as:

POC,FUNCTION-CODE,VALUE



Valid parameters are defined next; the handling of invalid parameters is discussed under "Invalid Parameter," on page 153. Values not defined are reserved.

Function parameter		Code
Indicator lights	Function keys	
No-op	No-op	X'C0'
Set individual indicators on	Identify function key	X'C1'
Set individual indicators off		X'C2'
Set indicators by mask		X'C3'
Set all indicators off		X'C4'

Value Parameters: The value parameter is required on all POC sequences to provide a consistent length, but it has no meaning with the X'C0' and X'C4' function parameters. When the function parameter is X'C1' or X'C2', the value parameter is a binary number defining a unique indicator or function key. Valid values are from 1 to 255, depending on the number of indicators or function keys implemented. A zero value is a no-op. When the function parameter is X'C3', the value parameter is treated as an 8-bit mask, where each bit represents an indicator. A zero value in the mask causes the corresponding indicator to be set off, and a one value in the mask causes the corresponding indicator to be set on. All bit-setting combinations of the value parameter following the X'C3' function parameter are valid.

REPEAT (RPT, X'0A')

RPT is a device control used to manage a print buffer. It allows a device to print repeatedly the character string contained within a buffer. The repeat function is not performed at the time the code is placed in the buffer as the result of keyboard action. Instead, it is decoded during the process of printing the contents of a buffer. The function resets the buffer location pointer to the beginning of the character string to be repeated. (See "Auto-Letter Mode" on page 152 for the use of this function.)

REQUIRED FORM FEED (RFF, X'3A')

RFF is a formatting control that stops printing of the page and forces a form feed to the top and left margins of the next page on those devices that have this capability. RFF contrasts with the Form Feed (FF) code, which may be replaced or eliminated on those machines that have the capability to adjust text. Machines that do not have or are not using this capability may treat the RFF code as an FF code, and not cause an error condition.

REQUIRED NEW LINE (RNL, X'06')

RNL is a formatting control that causes a mandatory move of the print position to the left margin on the next printing line. This character is distinguished from the regular NL (new line) control in that NL is ignored when the machine is in an adjust-text mode. On those machines that do not have or are not using the adjust-text capability, RNL may be treated as NL, and no error condition created.

RNL may also be used to terminate a start-of-format data string headed by an SOF control. (See "Start of Format" on page 146.)

REQUIRED SPACE

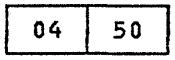
See "Graphic Codes that Possess Control Characteristics" on page 150.

SECURE STRING ID READER (SSR, X'0450')

SSR is used to distinguish between an operator identification number entered into the system from a magnetic stripe reader and one entered from a keyboard. This is accomplished by delimiting a magnetic stripe number with the SSR and IRS controls. (See "Interchange Separator" on page 135.)

Because magnetic stripe readers may be used to read both credit cards and operator identification cards, a device attaching a magnetic stripe reader must distinguish between the two types of cards and apply the SSR control to the data from the magnetic stripe on the operator identification card only. These magnetic stripes are distinguished by the use of an operator identification code as the first data character on the stripe containing an operator identification number. Before the data read from the stripe is forwarded, the code is replaced by the SSR control and the end of data code from the stripe is replaced by the IRS control. Data designated as secure through use of the delimiting SSR and IRS codes is not printed or displayed, nor can it be edited in the buffer. SSR appears as:

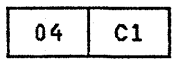
SEL,READER



SELECT LEFT PLATEN (SLP, X "04C1")

SLP, or select primary platen, is a device control that causes the data following it to be printed on the platen, starting from the left margin of the printer. When SLP is in effect, all the formatting control such as left margin, right margin, top margin, bottom margin, MPP, vertical and horizontal tab stops, and so on, are associated with the left platen only. Once SLP is issued, it remains in effect until another selection is made. When neither SLP nor SRP (select right platen) are in effect, the left platen is selected by default. SLP appears as:

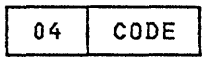
SEL,LEFT-PLATEN



SELECT MAGNETIC ENCODER (SME, X'04...')

SME is used to denote the start and encoding options of an EBCDIC data stream to be encoded on a magnetic stripe. The end of the data stream is denoted by the IRS control. Invalid graphic characters cause a negative response indicating an RU data error. This error response is also sent if the magnetic stripe message length is greater than the device capacity. SME appears as:

SEL,FUNCTION-CODE



The character following X'04' is used to define the magnetic stripe code set to be used in the operation and whether the data stream is secure or nonsecure.

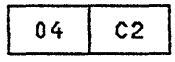
Function	Code
Encode non-secure data, numeric set	60
Encode secure data, numeric set	61
Encode non-secure data, alphanumeric set	62
Encode secure data, alphanumeric set	63

If multiple SME data streams are received by the encoding device, only the last one received is retained for encoding.

SELECT RIGHT PLATEN (SRP, X'04C2')

SRP, or select secondary platen, is a device control that causes the data following it to be printed on the right platen, starting from the left margin defined for right platen. When SRP is in effect, all the formatting controls, such as left margin, right margin, top margin, bottom margin, MPP, vertical and horizontal tab stops, and so on, are associated with right platen only. Once SRP is issued, it remains in effect until another selection is made. When neither SRP nor SLP are in effect, the left platen is selected by default. SRP appears as:

SEL,RIGHT-PLATEN



SET ATTRIBUTE (SA, X'28...')

SA is a control that associates attribute values to the graphic characters that follow in the data stream to define:

- Highlighting
- Color
- Character set

SA appears as:

SA,TYPE,VALUE

where:

Type = Characteristic to be defined
X'41' - Highlighting
X'42' - Color
X'43' - Character set
X'00' - Reset characteristics

Value = Attribute value to be associated to the following SCS graphic characters according to the type code.

Values are shown below by type:

Highlighting

X'00' - Default
Other - All other values are assigned to highlighting identification by the Query Reply (Highlight) structured field.

The value X'00' selects the device default highlighting indicated in the Query Reply (Highlight) structured field.

Color

X'00' - Default
X'F7' - Multicolor
Other - All other values are assigned to color identification by the Query Reply (Color) structured field.

Values X'00' and X'F7' have unique data stream meaning. The value X'00' selects the device default color indicated in the Query Reply (Color) structured field. The value X'F7' indicates the color is defined using a triple-plane programmed symbol set. If a single-plane character set is referenced, the color defaults to the single color specified for the X'F7' data stream value by the Query Reply (Color) structured field.

Character Set

X'00' - Default - base character set
Other - Defined by user
X'FF' - Reserved

The default values are:

Highlighting - No highlighting, normal presentation of graphics as defined by the implementor.

Color - Monochrome color as defined by the implementor

Character set - Base character set as defined by the implementor

SA values are associated with their respective type upon setting and until receipt of another SA control of the same type. Therefore the set of attribute values is a composite by attribute type of the attribute specified in any previously encountered SA control. For example:

Setting	Highlight	Color	Character set
Initial Setting	Default	Default	Default
SA color red	Default	Red	Default
SA character set X'F3'	Default	Red	X'F3'
SA color blue	Default	Blue	X'F3'
SA highlight underline	Underline	Blue	X'F3'

Unrecognized attribute values are handled in the following ways:

Highlighting and Color: Unsupported valid values are folded into implementation defined supported values for presentation.

Whenever a hyphen is substituted for an undefined graphic or for an unsupported/invalid control code, the color and highlighting data stream attributes associated with the substituted code point still apply.

Invalid values are rejected with a negative response.

Character Set: Unsupported values are rejected with a negative response.

SET CHAIN IMAGE (SCI, X'2B...')

SCI specifies a character set image. It is used to load a character set image into a chain printer. SCI appears as:

CSP,CLASS,COUNT,CHAIN,IMAGE

2B	D1	CNT	CX	IMAGE

0 8 16 24 32 n

where:

2B D1 Specifies a multiple-character-set-and-font class of control functions.

CNT One-byte count field indicating the length of the sequence including the CNT byte. A count byte of 0 or 1 is invalid. If the count byte is 2, the default of the device is to be used.

CX SCI control function and chaining indicator:

X'C6' - First segment of the chain image
X'C4' - Middle segment of the chain image
X'C5' - Last segment of the chain image
X'C7' - Only segment of the chain image

IMAGE The chain image

Chained SCI controls must follow in sequence, and no intervening SCS controls should appear between them. The first character after the chaining byte in the first or only segment in the chain represents the first character past home on the chain. If the length of the SCI image and the hardware on the device are not equal, the chain image is truncated or repeated as required.

SET GRAPHIC ERROR ACTION (SGEA, X'2B...')

SGEA specifies the action to be taken when an unsupported or unprintable graphic character is received, either online or from a spooling file. SGEA specifies if a sense code is to be issued, or if printing is to continue, with a substitute graphic to represent the unprintable graphic.

The SGEA control contains a 1-byte count field (always X'03'), the substitute graphic, and a 1-byte binary field to indicate whether to continue or to send a negative response. SGEA appears as:

CSP,CLASS,COUNT,GRAPHIC,ACTION

2B	C8	03	GR	AC

0 8 16 24 32 47

where ACTION is defined as:

X'01' Print substitute graphic and continue
X'03' Issue a negative response

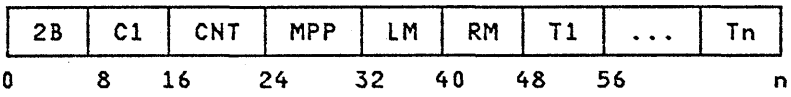
The default action taken when an unsupported graphic is found is discussed under "Unsupported Graphics" on page 153. If SGEA is issued, it overrides that action.

SET HORIZONTAL FORMAT (SHF, X'2B...')

SHF sets horizontal formatting controls, including maximum presentation position, left and right margins, and horizontal tab stops. A 1-byte binary count follows the SHF control; it indicates the number of bytes to the end of the SHF sequence, including the count byte. The first three parameters following the count define the maximum presentation position, the left margin, and the right margin, respectively. Tab stop parameters follow the right-margin (RM) parameter position. Parameters of SHF are 1 byte each, and contain binary numbers in the range 0-255.

The receipt of the SHF control sets all horizontal formatting controls to their default values. The minimum valid sequence which may be sent is SHF with a count of one, in which case all horizontal formatting controls have default values. A zero value for any other SHF parameter is a no-op and results in the function retaining its default value. The default values to be assumed are those described below under the definition of each parameter, unless default values are established by local action at the device, in which case the locally established values are assumed. SHF appears as:

CSP,CLASS,COUNT,MPP,LM,RM,T1,...Tn



where:

MPP Maximum presentation position: Specifies the horizontal extent of the presentation surface (for example, print-line length, in number of characters, with column one the origin). Valid MPP values are less than or equal to the device physical maximum presentation position. The MPP default value is the physical device line length.

LM Left margin value: Specifies the column value of the leftmost presentation position. LM also serves as the first horizontal tab stop. Valid LM values are values less than or equal to MPP. The LM default value is one.

RM Right margin: Used to assist an operator in formatting keyboard-generated data streams. One of its functions is to warn the operator that the end of the line is approaching; the warning is issued only once, when the presentation position column number equals RM minus 10. If the distance between LM and RM is ten or less, the warning is not issued. Valid RM values are values greater than or equal to LM and less than or equal to MPP. The default value for RM is MPP.

T1..Tn Horizontal tab stop parameters: Set column values for use with the Horizontal Tab (HT) function. The tab string does not have to be in order. Valid tab stop values other than zero are values equal to or greater than LM and less than or equal to RM.

Note: The LM parameter is the first horizontal tab stop; repeating the LM value in the horizontal tab stop parameter sequence is redundant. When no tab stop values are set to the right of the current presentation position column value, the HT function acts as a space function.

The following algorithm is executed by a half-session when SHF is received:

1. The SHF parameters are scanned, checked for validity, and saved in an SCS control vector.

2. An appropriate negative response is sent if an error is detected; the effect of the saved parameters is undefined. (For predictable results, any recovery sequence must include a valid SHF.)

To avoid unpredictable results, an SHF must be followed by an FF, RFF, NL, or RNL control. Although an FF, RFF, NL, or RNL control does not have to immediately follow the SHF, there must be no intervening data or control code that causes movement of the active presentation position. Where one of these

controls does follow a sequence of formatting controls (for example, SHF, SPD), the ordering of the sequence has no effect on the execution of the following controls. LM positioning is in terms of the new SHF parameters.

Since default values for MPP, LM, RM, and horizontal tabs may be established by local action at a device, it is necessary, when using these parameters to format the presentation surface, to send SHF or to rely on end-user defined procedures to establish the values.

SET LINE DENSITY (SLD, X'2B...')

SLD specifies the distance to be moved for a single-line vertical space as in LF or NL. It contains a 1-byte count field and a 1-byte binary parameter. The binary parameter expresses the vertical distance in standard typographic points (one point = 1/72 inch). Thus, a value of 12 points would indicate six lines to the inch; 9 points would indicate eight lines to the inch; and 72 points would indicate one line to the inch.

When the SLD control is received, the device sets line density to its default value before analyzing the count and density fields. This default value is retained unless valid count and density values are received. The following parameter values are accepted without rejection with default values retained:

Count byte = X'02' and points byte = X'00'
Count byte = X'01' (indicates no points byte)

Count values other than X'01' and X'02' and points values other than X'00' and those implemented by the device must be rejected with an error response indicating an invalid parameter. SLD appears as:

CSP,CLASS,COUNT,POINTS

2B	C6	CNT	PNTS

0 8 16 24 31

The implementing device must specify the points values it implements and the default value. The device default value may be established by local action. If so, the locally established values are assumed as defaults in the cases mentioned above. To avoid unpredictable results, such as loss of forms integrity, an SLD must be followed by an FF or RFF control. Although the FF or RFF control does not have to immediately follow the SLD, there must be no intervening data or control code that causes movement of the active presentation position. When one of these codes is used following a sequence of formatting controls (for example, SLD, SHF), the ordering of the sequence has no effect on the execution of the formatting controls.

The FF or RFF control causes exiting of the current presentation surface in terms of the old SLD parameter, but positioning in the next presentation surface is done in terms of the new SLD parameter.

The line density specified by SLD becomes effective immediately on receipt of SLD.

Note: Failure to observe the following rules results in misalignment of the printed output relative to the form:

- When the using application always specifies TM = 1, the form must be positioned to line 1 (for example, by using FF) prior to the receipt of SLD.
- When a TM greater than 1 is being used, the form must be manually aligned to the applicable TM prior to receipt of SLD (the alignment must be in terms of the line density specified in the SLD).
- A change of line density requires a corresponding change of MPL or physical forms, or both.

SET PRINT DENSITY (SPD, X'2B...')

SPD is a formatting control; it specifies the number of characters to be presented per inch in the horizontal direction. SPD appears as:

CSP,CLASS,COUNT,TYPE,CHARACTER-DENSITY

2B	D2	CNT	29	CD

0 8 16 24 32 47

where:

CD is character density: A 2-byte binary number from 1 to 255, specifying the number of characters to be printed per inch. Numbers 256 to 65535 are reserved. If CNT = X'04' and CD = X'0000' or CNT = X'02', the default value of the device is used.

The character density specified in SPD becomes effective immediately on receipt of SPD. When the SPD code is received, the device sets the character density to its default value. The default value is retained unless valid count and character density values are received.

Note: If a change in character density is not compensated by changes in LM, RM, and MPP using SHF (set horizontal format), the left and right margins may be misaligned.

SET TRANSLATION TABLE (STT, X'2B...')

STT is a device control used to load the character set translation table in a chain printer. STT appears as:

CSP,CLASS,COUNT,SEGMENT,RESERVED,START,IMAGE

2B	D1	CNT	CX	RESERVED	START	IMAGE

0 8 16 24 32 64 80 n

where:

2B D1 Indicates multiple-character-set-and-font class of control.

CNT Is a 1-byte count field indicating the length of the sequence including the count byte.

CX The STT control function and the segment of the translation table as follows:

X'C2' First segment of the translation table
X'C0' A middle segment of the translation table
X'C1' Last segment of the translation table
X'C3' Only segment of the translation table

START Is a 2-byte field indicating at what starting position in the translation table the new image is loaded. The default is X'FFFF'.

IMAGE The variable-byte translation table.

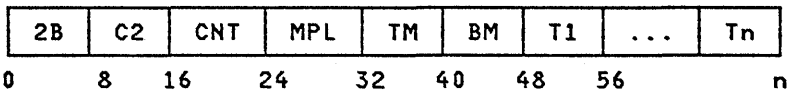
SET VERTICAL FORMAT (SVF, X'2B...')

SVF sets vertical formatting controls, including maximum presentation line (page size), top margin, bottom margin, and vertical tab stops. A one-byte binary count follows the SVF code; it indicates the number of bytes to the end of the SVF string, including the count byte.

The receipt of the SVF code sets all vertical formatting controls to their default values. The minimum sequence that can be sent is with a count of one, in which case all vertical format controls are set to their default values.

A zero value for MPL, TM, or BM is a no-op and results in the function retaining its default value. The default values to be assumed are those described below under the definition of each parameter, unless default values are established by local action at the device, in which case the locally established values are assumed. Vertical tab stop parameters in the SVF sequence start after the bottom-margin (BM) parameter position. Parameters of SVF are one byte each, and contain binary numbers in the range 0-255. SVF appears as:

CSP,CLASS,COUNT,MPL,TM,BM,T1,...Tn



where:

MPL Maximum presentation line: Defines the vertical extent of the presentation surface with origin of line one. This is the page depth for a printer or the number of lines in a display. All values from 1 to 255 are valid. A page size defined by SVF takes precedence over a fixed device value such as display screen line capacity. The default value is the device-fixed line capacity, for those devices that have a fixed-line capacity, and the locally-established MPL for those devices capable of establishing values locally, and is one for devices with neither fixed-line capacity nor locally-setable values.

TM Top margin: Specifies the line value to be used as the top presentation line of the page. The TM is also the first vertical tab stop. Valid TM values are equal to or less than MPL. The default value for TM is one.

BM Bottom margin: Specifies the line value that, if exceeded by L#(new), causes an automatic skip to TM of the next presentation surface. BM must be greater than or equal to TM, and less than or equal to MPL. The BM default value is the MPL value.

T1...Tn Vertical tab stops, selected by VT, or vertical channels, selected by VCS.

Each vertical tab stop parameter specifies a single-line value for use with the vertical tab (VT) function. Valid vertical tab values are equal to or greater than TM (the first vertical tab stop which must not be specified explicitly) and equal to or less than BM or X'00'. While X'00' is a permissible tab stop value, only nonzero values are processed. MPL must be specified as a value greater than one if useful vertical tab stops are to be available. Vertical tabs must be listed in increasing order if predictable results are to be achieved with the VT function. Vertical tab stops may be set at any line, except TM, down to, and including, bottom margin.

Tab stop parameters 1 through 11 are also used to set line values for corresponding control channels 2 through 12; channel 1 is set by the TM. This allows vertical formatting to be accomplished with the vertical channel select (1-12) function in addition to the VT function. The value zero may appear in any of tab stops 1 through 11. A zero line value specified to correspond to any of channels 2 through 12 will result in a default to the LF function when a select channel is given for that channel number. A select channel 1 causes line positioning to move to the TM of the next presentation surface with no change in column position. Vertical tab line values must be in increasing order when used to set channel stops, except that the value zero may appear out of sequence.

The following algorithm is executed by a half-session when SVF is received:

1. The SVF parameters are scanned and checked for validity and saved.

2. An appropriate negative response is sent if an error is detected. The effect of the saved parameters is undefined. (For predictable results, any recovery sequence must include a valid SVF.)

The SVF parameters become effective immediately on receipt of SVF; the device assumes the form is aligned to the line specified by the TM parameter.

Note: Unless the using application always specifies a TM = 1, the form must be manually aligned to the line specified by TM prior to the receipt of SVF. Failure to observe the preceding rule results in misalignment of the printed output relative to the form.

Since default values for MPL, TM, BM, and vertical tabs may be established by local action at a device, it is necessary, when using these parameters to format the presentation surface, to send SVF or to rely on end-user defined procedures to establish the values.

SHIFT IN (SI, X'0F')

SI is a mode control that indicates the bit patterns which follow shall be interpreted according to the most recently designated Shift In graphic code page.

SHIFT OUT (SO, X'0E')

SO is a mode control that indicates the bit patterns which follow shall be interpreted according to the most recently designated Shift Out graphic code page.

START OF FORMAT (SOF, X'2B...')

SOF is a grouping control that is used as a delimiter to separate normal data from unique header information. SOF appears as:

CSP,SOF-CODE

2B	C3

0 8 15

The SOF control functions are:

Single index
Double index
Adjust mode on
Adjust mode off
Tab
Right margin setting

The format of this information must be such that a simple machine can print out these parameters so that an operator can use this information to manually set up the machine, or a more intelligent machine can operate on this same parameter format to automatically set up itself. This special mode in the character string is terminated by the RNL (required new line) or IRS (interchange record separator) control. An error condition can be generated when a device does not see an SOF-RNL or SOF-IRS pair in the proper sequence in the outbound character string.

SUBSCRIPT (SBS, X'38')

SBS is a formatting control that causes a fractional line feed operation in the forward direction. The value of the fraction is less than one line and is equal to the equivalent superscript movement. This is a latching type control, which requires an SPS (superscript) control from the sending half-session to return the presentation position toward the base line.

Multiple SBS characters may be used when multiple levels of subscript are required, and each must be canceled with a corresponding SPS code to return the presentation position to the base line.

A subscript is logically a part of the base-line word or character and, thus, is not split between lines when automatic new line is performed.

Any line advance before returning to the base line gives unpredictable results.

See Figure 1-2 for a flowchart of SBS interpretation.

SUBSTITUTE (SUB, X'3F')

SUB is an error control that replaces a character that is determined to be invalid or in error. For example, during translation from EBCDIC to ASCII, the SUB control may be used in the ASCII code set to represent codes that have a definition in the EBCDIC code set but have no definition in the ASCII code set.

SUPERSCRIPT (SPS, X'09')

SPS is a formatting control that causes a fractional line feed in the reverse direction. The value of the fraction is less than one line, and is equal to the equivalent subscript movement. This is a latching type control, which requires a subscript (SBS) code from the sending half-session to return the presentation position toward the base line.

Multiple SPS codes may be used when multiple levels of superscript are required, and each must be canceled with a corresponding SBS code to return the presentation position to the base line.

A superscript is logically a part of the base-line word or character and, thus, is not split between lines when automatic new line is performed.

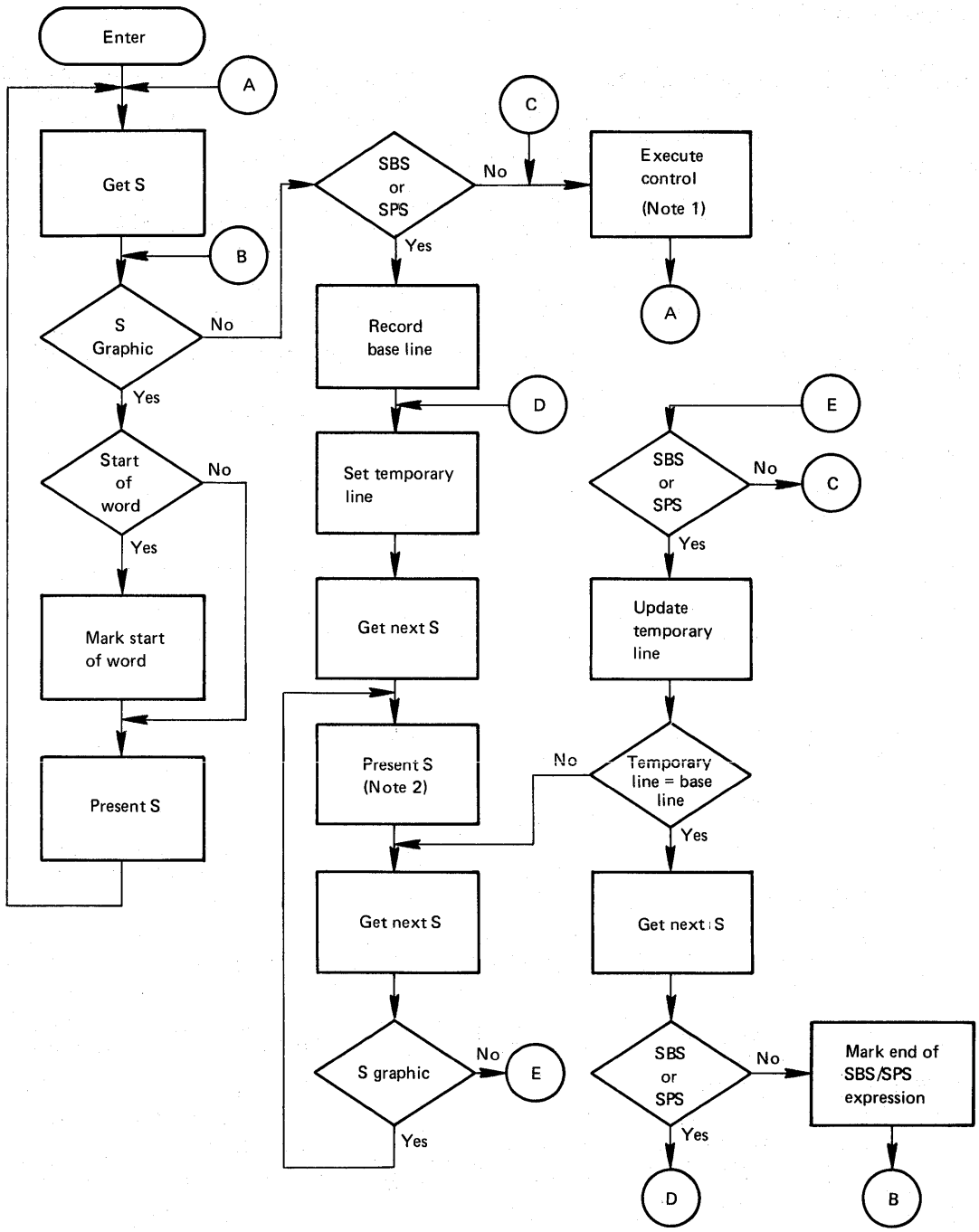
Any line advance before returning to the base line provides unpredictable results.

See Figure 1-2 for a flowchart of SPS interpretation.

SWITCH (SW, X'2A')

SW is a device control used to manage print buffers. Its purpose is to provide a mechanism for switching dynamically between two print buffers during the printing process. Inherent with the switch function is the requirement that each buffer maintain a buffer location pointer. This pointer indicates the buffer location that supplies the next data or control characters.

The switch function is not performed at the time the code is placed in the buffer as a result of keyboard action. (See "Auto-Letter Mode" on page 152 for the use of this function.)



Note 1: Any line-ending code within an SBS/SPS expression can cause unpredictable results.

Note 2: Flowchart logic assumes that there is always a printable character following SBS or SPS.

Figure 1-2. Example of subscript and superscript scan algorithm (S = next character in string)

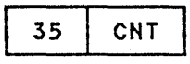
SYLLABLE HYPHEN (SHY, X'CA')

See "Graphic Codes that Possess Control Characteristics," on page 150.

TRANSPARENT (TRN, X'35...')

TRN is a data-defining control used to denote the start of a transparent data stream. The data delimited by TRN is end-user defined and is not scanned for SCS control codes. TRN appears as:

TRN,COUNT



where:

CNT Indicates the number of bytes of transparent data (not including the count byte).

After processing the TRN data, the presentation position must be moved CNT positions.

UNIT BACKSPACE (UBS, X'1A')

UBS provides the small (less than character width) incremental capability needed to get absolute vertical alignment of line endings on proportional space devices.

Nonproportional-space machines may ignore this code.

VERTICAL CHANNEL SELECT (VCS, X'04...')

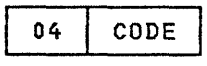
VCS is a formatting control used to select one of twelve vertical channels for the purpose of controlling the vertical format of a presentation surface.

Line stop values for each channel used must be designated before using these select functions if default values are not to be used. (See "Set Vertical Format," on page 144.)

When the current line number is greater than or equal to the largest line value specified for the channel selected, positioning is to the next page, and the line value specified for the selected channel is used. When there are no line stop values specified for a channel, a select channel function for that channel defaults to LF (line feed).

VCS appears as:

SEL,CHANNEL-CODE



where:

Select function	Code
Select channel one	X'81'
Select channel two	X'82'
Select channel three	X'83'
Select channel four	X'84'
Select channel five	X'85'
Select channel six	X'86'
Select channel seven	X'87'
Select channel eight	X'88'
Select channel nine	X'89'
Select channel ten	X'7A'
Select channel eleven	X'7B'
Select channel twelve	X'7C'

VERTICAL TAB (VT, X'0B')

VT is a formatting control that moves the presentation position vertically down to the next tab stop setting. Vertical-tab stop values may be set using SVF (set vertical format). If there are no vertical tab stops set below the current line number, VT defaults to LF (line feed). While zero is a permissible tab stop value, only nonzero values are processed.

```
C#(new)<-----C#(old)
IF: VTV(any) > L#(old)
    Then: L#(new)<---VTV(next)
    Else: L#(new)<---L#(old)+1
```

where:

VTV = Vertical tab stop value expressed as a line number.
VTV(next) = Smallest VTV>L#(old).

WORD UNDERSCORE (WUS, X'23')

WUS is a device control which, when entered from the keyboard or executed during printing, causes the entire word immediately preceding the WUS control to be underscored. The word to be underscored is defined as the string of characters before the WUS code back to one of the following: space, underscore, or any of the SCS controls except SPS and SBS (superscript and subscript).

GRAPHIC CODES THAT POSSESS CONTROL CHARACTERISTICS

This section lists graphic codes that have unique control characteristics in certain applications.

EXPANDED SPACE (ESP, X'E1')

ESP is a large, fixed-size space used in justification of text and in columnar and forms applications for proportional-space devices. The normal space frequently is not large enough to provide the columnar formatting function. Text can be justified more easily when two different space widths are used. Nonproportional-space devices may recognize this control as a normal space.

REQUIRED SPACE (RSP, X'41')

RSP is distinguished from the normal space in that it is treated as any graphic character and therefore is not considered an interword space on those machines that have adjust-text capability. Implementations that do not have this capability may default RSP to a normal space without causing an error.

SYLLABLE HYPHEN (SHY, X'CA')

SHY is a graphic that prints as a normal hyphen (X'60'). Its use is limited to a syllable boundary at a line ending to indicate the continuation of a word on the next line. Use of a unique control for this function permits the syllable hyphen to be discarded in adjust-text mode when the adjusting process moves the word away from the line straddling position. The normal hyphen is treated as any other graphic in adjust-text mode and is never discarded. In devices without adjust-text capability, the syllable hyphen is treated as a normal hyphen.

MEDIA FORMATTING

The formatting controls described earlier format the output medium at a device. In addition to these controls, a device implementing the SNA character string (SCS) can format a received character string to fit the line length of the device automatically. This capability of automatic new-line generation eliminates device line-length dependencies from those applications where a specific output format is not required. Thus, the same character string can be sent to devices with varying line lengths without having to reformat the character string.

The automatic new line function cannot be executed until receipt of the first graphic character, including a nonprinting space character, after the MPP has been exceeded. This implies a character look-ahead and allows execution of control codes such as BS (backspace) before the automatic new line is performed. This function is mandatory for the output string only. When a character string is generated from the keyboard, the device implementations may optionally perform an automatic new line or reject additional keyboard input when MPP is exceeded.

A device that provides left-adjusted text may discard space characters that would occur at the beginning of a line, and may adjust the text to avoid improper placement of punctuation marks at the beginning of a line.

When specific line and page formats are required, formatting controls are used. The automatic new line feature is always active; thus, a character string formatted for a given line length can be presented on a device with a shorter line length without loss of data, but the format is changed. When reversed (where a character string is constructed for an MPP that is less than the line length capability of the device it is directed to), use of the smaller MPP in an explicit SHF allows the string to be presented without loss of data or change of format.

BM (bottom margin) specifies a line value that, when exceeded by L#(new), causes an automatic skip to TM (top margin) of the next page. The function that causes the line positioning to exceed the BM value determines the horizontal positioning on the new line. For example, if NL (new line) causes the automatic skip, the horizontal positioning is to the left margin. If LF (line feed) causes the automatic skip, the horizontal positioning is unchanged. Note that automatic new line may also cause an automatic skip if the new line created exceeds BM.

SCS formatting controls assume that the output medium consists of a matrix of columns and lines, where column one, line one, is the top left position on a page or display screen. FF (form feed) positions the print element to a predefined column (LM) and line (TM) on a new page or screen.

Valid column values for a device are from one to the device maximum. In addition to column positioning by defined controls, each graphic data character causes the column position to be incremented by one after printing.

Line positioning is accomplished through the use of control functions that can alter line values. Use of functions that specify a line position greater than the next sequential line requires the definition of a page size in terms of the total number of lines on the page. The page size may be defined as a fixed value for a given device (such as a display), or it may be specified via a parameter on SVF (set vertical format). The page size given with SVF takes precedence over a device-defined page size value.

RELATIONSHIP OF INPUT AND OUTPUT

Implementation of SCS by an input/output device may include the capability to generate, as well as receive, SCS controls. An input string can contain all the control functions defined for SCS. SCS controls that are initiated from a device keyboard cause the insertion of the appropriate SCS code in the data stream that is transmitted from the device.

ADJUST-TEXT MODE

Adjust-text mode allows implementations to reformat text to accommodate varying line lengths and page sizes, and to provide assistance to the operator in adjusting line endings. When a device is in adjust-text mode, certain formatting controls may be eliminated in the adjusting process, and others may not be eliminated. Required controls are RSP (required space), RNL (required new line), and RFF (required form feed). SHY (syllable hyphen) and IT (indent tab) are also required.

A device is placed in adjust-text mode by SCS data stream control or an operator key on the keyboard.

TEXT-JUSTIFY MODE

Text-justify mode allows implementations to reformat text to align the line endings exactly on right margin. This is accomplished by changing the effective width of the interword space throughout the printed line. SCS controls for proportional spacing, such as EPS (Expanded Space), EBS (Expanded Backspace), and UBS (Unit Backspace) are used separately or in combination to create the desired space width. The device is placed in text-justify mode by an SNA character string control or an operator key on the keyboard.

AUTO-LETTER MODE

Auto-letter mode allows implementations to create a set of finished documents (letters) from two different text sources. These sources can be any combination of memory buffers, magnetic media, operator keyboards, or even the communication line. By using the SW (switch) and RPT (repeat) controls, the two sources of text can be serially combined in any desired fashion to produce completed documents. The normal use of this mode is to prepare completed letters from one source containing the body of the letter and a second source containing names and addresses.

FORMAT CONTROL PARAMETER DEFAULT

Active format control parameters are to be determined in the following priority:

1. Received via the data stream as a valid parameter (that is, within limits defined for the specific parameter) following a valid format control code.

2. Locally defined. A locally defined parameter may be of types:

   a. Hardware physical limits, for example, the MPP limit of 132 on a printer.

   b. Retainable (static) operator-established values for example, MPL set in rotary or decade switches.

   An example of nonretainable operator established values that would not be available for a default return would be dynamic horizontal tab values entered by an operator from a keyboard. If these values were dynamically entered into an active table, they may be over written by a valid SCS parameter and not available for default later.

3. Architecture defined. An architectural default is defined for each parameter. If a parameter for type 1 or 2 above is not available, this value is made active.

Note: For predictable results, any recovery must include a valid format control code and parameters.

ERROR HANDLING

SCS errors are considered errors encountered in creating the character string. They are either program errors or operator errors, depending on the source of the character string.

INVALID CONTROL CODE

This error occurs when a control code or control code sequence is detected by a receiving device that has not implemented the received code.

INVALID PARAMETER

This error is indicated when a function or value parameter is detected that:

- Contains an invalid code for that parameter
- Contains a value that is outside the range of the device. This includes the case of vertical positioning past the next sequential line, when there has been no presentation surface size specified for the device. In the absence of a defined presentation surface size, either inherent in the device or specified by SVF (set vertical format), a device is considered to have a presentation surface size of one line.

Note: For sessions where it is important to display as much of the data as possible, even if an occasional error occurs, or for spooled, offline, printing jobs where an error response would be meaningless, a default action may be substituted for an invalid control code or an invalid parameter.

UNSUPPORTED GRAPHICS

If characters from a large character set are transmitted to a device with a smaller character set:

- Lowercase alphabetics, including national use alphabetic substitutions, are folded into the corresponding uppercase alphabetic.

- All other unsupported graphics are folded to the graphic specified by SGEA (set graphic error action). Hyphen (-) is the substitute graphic if SGEA is not used.

No error codes are sent if an unsupported graphic is received.

UNEXPECTED OUTPUT WITH CONTIGUOUS BACKSPACES

The automatic new line function can cause a BS control to occur at an unpredictable position relative to the physical end of the line when contiguous BS controls are found in the data stream. Since the BS control is a no-op if the presentation position is at left margin, contiguous BS controls should be avoided.

There are no error responses as a result of the misuse of the backspace function. If an error occurs, however, it may be detected by examination of the printed output.

SCS DEFAULTS

An SCS control may be treated in one of three ways:

1. The code is recognized, and the function is performed.
2. The code is recognized, but a default is taken, and the specified function is not performed.
3. The code is not supported and a negative response is returned.

A particular code is supported in the case of either (1) or (2) above; both require hardware or software support to recognize the code. The difference is in whether an implementation can perform the function or must execute a default.

Figure 1-3 shows the defaults.

Taking default action for SCS controls specified by a single code point is straightforward. Controls with multiple control points are more complicated. Some controls, such as the controls based on SEL (X'04'), are 2-byte sequences in which the second byte is a valid graphic character. Taking a default for such a code could cause an extraneous graphic character to be printed. Other SCS controls include binary value parameters. If they are not recognized as value parameters, unpredictable results could occur (they could be misinterpreted as valid SCS controls).

Controls that contain binary parameters include PP (presentation position), POC (program operator communication), TRN (transparent), and the controls based on X'2B' (CSP) as the first byte of the control.

Note: If a control is not included in Figure 1-3, refer to the control sequence prefix entry in the figure.

SECOND-LEVEL DEFAULTS

A default for a particular SCS control may, in turn, take a second default. For example, IR defaults to RNL, which may further default to NL.

Control	Abbr	Code	Default
Backspace	BS	16	No-op
Bell	BEL	2F	No-op
Carriage return	CR	0D	New line
Control sequence prefix	CSP	2B	No default
Customer use 1	CU1	1B	No-op
Customer use 3	CU3	3B	No-op
Device control 1	DC1	11	No-op
Device control 2	DC2	12	No-op
Device control 3	DC3	13	No-op
Device control 4	DC4	3C	No-op
Enable presentation	ENP	14	No default
Expanded backspace	EBS	36	Backspace
Expanded space	ESP	E1	Space
Form feed	FF	0C	New line
Graphic escape	GE	08	No default
Horizontal tab	HT	05	Space
Indent tab	IT	39	Tab
Index return	IR	33	Required new line
Inhibit presentation	INP	24	No default
Interchange file separator	IFS	1C	New line
Interchange group separator	IGS	1D	New line
Interchange record separator	IRS	1E	New line
Interchange unit separator	IUS	1F	Space
Line feed	LF	25	New line
New line	NL	15	No default
Null	NUL	00	No-op
Presentation position	PP	34	No-op
Program operator communication	POC	17	No-op
Repeat	RPT	0A	Bell
Required form feed	RFF	3A	Form feed
Required new line	RNL	06	New line
Required space	RSP	41	Space
Secure string ID reader	SSR	0450	No default
Select left platen	SLP	04C1	Left platen
Select magnetic encoder	SME	046n	No-op
Select right platen	SRP	04C2	Left platen
Set chain image	SCI	2BD1	No-op
Set graphic error action	SGEA	2BC8	No-op
Set horizontal format	SHF	2BC1	No-op
Set line density	SLD	2BC6	No-op
Set translation table	STT	2BD1	No-op
Set vertical format	SVF	2BC2	No-op
Subscript	SBS	38	No-op
Substitute	SUB	3F	Space
Superscript	SPS	09	No-op
Switch	SW	2A	Bell
Syllable hyphen	SHY	CA	Hyphen
Transparent	TRN	35	No-op
Unit backspace	UBS	1A	No-op
Vertical channel select	VCS	04nn	Line feed
Vertical tab	VT	0B	Line feed
Word underscore	WUS	23	No-op

Figure 1-3. Defaults of SCS controls.

MULTIPLE-CHARACTER-SEQUENCE DEFAULTS

SCS controls containing more than one 1-byte character, and SCS controls with parameters except for SSR (secure string ID reader), default to no-op. (SSR does not have a default.) Implementations that choose to take these defaults must decode and ignore the entire sequence. Otherwise, a negative response may be returned.

SCS CODE POINTS

SCS code points are shown in Figure 1-4.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL															
1		DC1			RSP										ESP	
2		DC2														
3		DC3	WUS	IR												
4	SEL	ENP	INP	PP												
5	HT	NL	LF	TRN												
6	RNL	BS		EBS												
7		POC														
8	GE		SA	SBS												
9	SPS			IT												
A	RPT	UBS	SW	RFF									SHY			
B	VT	CU1	CSP	CU3												
C	FF	IFS		DC4												
D	CR	IGS														
E	SO	IRS														
F	SI	IUS	BEL	SUB												

Figure 1-4. SNA character string (SCS) code points.

This chapter contains criteria used to select functions of the SNA 3270 data stream. It shows how those functions are grouped to obtain compatible sets of function within IBM products. It does not define the formats and protocols of the SNA 3270 data stream; these are defined in 3270 Data Stream Programmer's Reference.

This chapter contains detailed information. You may need to read 3270 Data Stream Programmer's Reference before reading this chapter.

The 3270 data stream consists of user-provided data and commands which are transmitted between the primary logical unit (PLU) and the secondary logical unit (SLU) of an LU-LU session. Control information, which governs the way data is handled and formatted, is also transmitted.

The SNA 3270 data stream is the only data stream used on LU-LU session types 2 and 3. It is an optional data stream on LU-LU session types 0 and 6. The data stream supports both display and printer applications.

An application program communicates with a display operator using one of two methods. In one method, the display surface is left unformatted by the application program, and the operator uses it in a free-form manner. In the second, the display surface is completely or partially formatted (organized or arranged into fields), and the operator enters data in the fields.

The data stream allows the application programmer to divide the display surface into one active area and, optionally, one or more reference areas. Each area is called a partition. The partition that is active contains a cursor, and it is the only partition in which the operator can enter data or requests.

Each function of the SNA 3270 data stream can be viewed as being contained in a function set. In order to implement a function in any given function set, you must implement all of the operations defined by that set, and all operations defined by previous sets in the hierarchy.

FUNCTION SETS OF THE DATA STREAM

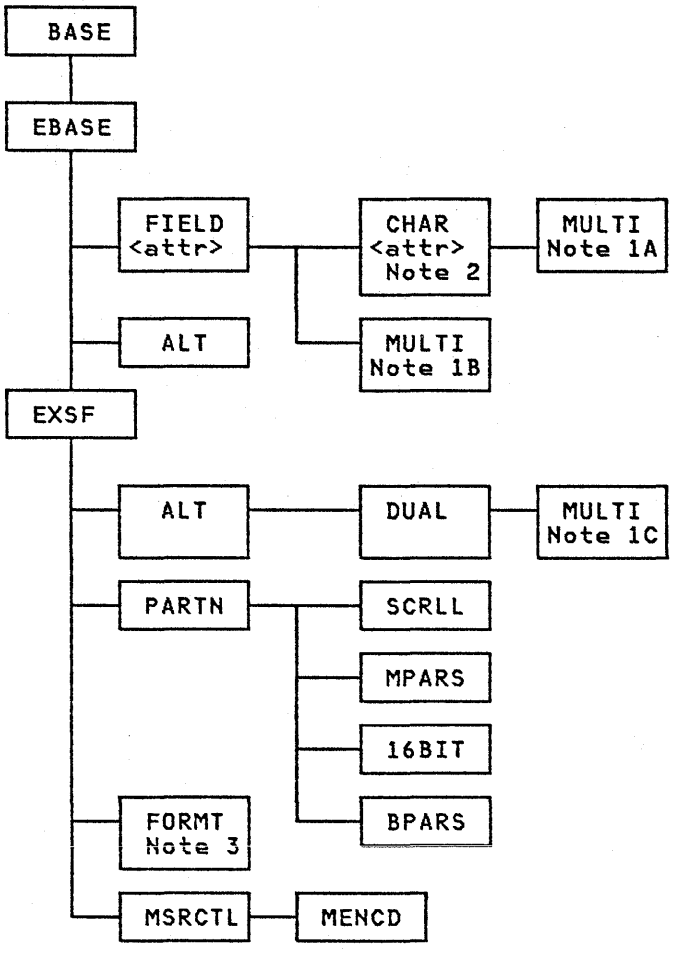
The rules for defining function sets used in IBM products are based on the structure shown in Figure 2-1.

In order to implement a function in any given set in this structure, you must implement all of the operations defined by that set and all operations in previous sets. You may implement one or more sets for each device.

Thus, an implementor selects function sets for a product as follows:

1. Select the functions that you want in your product.
2. Select those function sets that support the chosen functions, and trace upward through the set structure. The operations defined by that set, and all the operations in previous sets, form a consistent group of functions.
3. Repeat step 2 as required to define all sets supported by the product.

The functions obtained by each function set are summarized in Figure 2-1. Items not applicable or partially applicable to printers are noted. More detailed descriptions are included below. When a detailed description contains a note, such as Note 2, the note is explained in Figure 2-1 (Part 2 of 2).



Note 1: If the MULTI set is chosen, the attribute types selected for (attr) must include character sets. Character sets can be selected using:

1A. Extended field attributes or character attributes (Note 1A above)
1B. Extended field attributes (Note 1B above)
1C. Graphic Escape characters (Note 1C above)

Note 2: Validation cannot be specified as a character attribute; it must be specified as a type within the extended field attribute.

Note 3: If the MPARS set is not supported, PID must equal 0.

Figure 2-1 (Part 1 of 2). Function set structure of the SNA 3270 data stream.

SET NAME	QUERY REPLIES	FUNCTION
BASE		All SNA 3270 commands except WSF and all orders except SA, SFE, and MF
EBASE	Usable Area	WSF (Write Structured Field) command Read Partition (Query operation only) WCC using Reset bit
EXSF	Usable Area (EXSF flag = 0)	Outbound 3270DS Erase/Reset
FIELD	<attr> Reply Modes = EXTENDED FIELD (Note 1)	SFE <attr> MF <attr> Set Reply Mode = EXTENDED FIELD (Note 1)
CHAR	<attr> Reply Modes = CHAR (Note 1)	SA <attr> Set Reply Mode = CHAR <attr> (Note 1)
ALT	Character Sets (ALT flag on)	Graphic Escape Character (X'08')
DUAL	Character Sets (DUAL flag on)	Define Alternate Character Set
MULTI	Character Sets (MULTI flag on)	Load Programmed Symbols
PARTN	Partitions (n=1) (Note 1)	Create Partition Read Partition (RM, RMA, RB operations) (Note 1) Reset Partition Inbound 3270DS
SCRLL (Note 2)	Partitions (Scroll flag(s) on)	Create Partition with presentation space > window Set Window Origin
MPARS (Note 2)	Partitions (n>1)	Activate Partition Destroy Partition
16BIT	Usable Area (16-bit flag on)	SBA, RA, EUA - 16 bit Create Partition - 16 bit
BPARS (Note 1)	Partitions (n>1)	Restrictions on certain commands, orders, and structured fields
FORMT (Note 3)	Format Presentation	Select Format Group Present Absolute Format Present Relative Format
MSRCTL (Note 2)	MS Device Control (TYPE = X'01')	MS Device Control (TYPE = X'01')
MENCD (Note 2)	MS Device Control (TYPE = X'02')	MS Device Control (TYPE = X'02')
Note 1: Limited application to LU_T3. Note 2: Not applicable to LU_T3. Note 3: If MPARS is not supported, PID must equal 0.		

Figure 2-1 (Part 2 of 2). Function set structure of the SNA 3270 data stream.

BASE (BASE)

This set must be implemented and the device must support:

- The following 3270 commands:
  - Read Buffer (RB)
  - Read Modified (RM)
  - Read Modified All (RMA)
  - Write (W)
  - Erase/Write (EW)
  - Erase/Write Alternate (EWA)
  - Erase All Unprotected (EAU)
- The following 3270 orders:
  - Start Field (SF)
  - Set Buffer Address (SBA)
  - Insert Cursor (IC)
  - Program Tab (PT)
  - Repeat to Address (RA)
  - Erase Unprotected to Address (EUA)
- 12-bit addressing
- Write control character (WCC) with reset bit always 1 and start printer bit supported

EXTENDED BASE (EBASE)

If this set is implemented, the device must support:

- All 3270 commands, including Write Structured Field (WSF), and all orders except SA, SFE, and MF
- Read Partition structured field (query operation)
- Query Reply (Usable Area) structured field with 12/14-bit addressing
  - 14-bit addressing is optional
- WCC with reset bit set to 0 or 1
- SBA, RA, EUA, with 12/14-bit addressing (14-bit addressing optional)

EXTENDED STRUCTURED FIELD SUPPORT (EXSF)

If this set is implemented, the device must support:

- Query Reply (Usable Area) with the EXSF flag set to B'0'
- Outbound 3270DS structured field
- Erase/Reset structured field

EXTENDED FIELD ATTRIBUTES (FIELD)

If this set is implemented, the device can respond to a query with the structured fields that describe the attribute properties supported, from the following:

- Query Reply (Color)
- Query Reply (Highlight)
- Query Reply (Character Sets)
- Query Reply (Field Validation) (Note 2)

In addition, the device must support:

- Query Reply (Reply Modes) structured field (Note 1)
  - field and extended-field modes
- Set Reply Mode structured field (Note 1)
  - field and extended-field modes
- Start Field Extended (SFE) order - 3270 and <attr>
- Modify Field (MF) order - 3270 and <attr>

CHARACTER ATTRIBUTES (CHAR)

If this set is implemented, the device supports the attribute types indicated by the query replies supported by its previous sets. In addition, the device must support:

- Query Reply (Reply Modes) structured field (Note 1)
  - character mode
- Set Reply Mode structured field (Note 1)
  - character <attr>
- Set Attribute (SA) order - <attr>

MULTIPLE CHARACTER SET CAPABILITY (MULTI)

If this set is implemented, the device must support the functions defined in the appropriate previous sets (FIELD, CHAR or DUAL). In addition, it must support the following:

- Query Reply (Character Sets) structured field
  - with MULTI flag set on
- Load Programmed Symbols structured field

ALTERNATE CHARACTER SET (ALT)

If this set is implemented, the device must support:

- Query Reply (Character Sets) structured field
  - with ALT flag set on
- Graphic Escape Character (X'08')

DUAL LANGUAGE CAPABILITY (DUAL)

If this set is implemented, the device must support:

- Query Reply (Character Sets) structured field
  - with DUAL flag set
- Define Alternate Character Set structured field

PARTITIONS (PARTN)

If this set is implemented, the device must support:

- Query Reply (Partitions) structured field
  - with n=1
- Create Partition structured field
- Inbound 3270DS structured field (Note 1)
- Read Partition structured field (Note 1)

SCROLLING (SCRLL) (NOTE 2)

In addition to the partitions functions found in previous sets, the device must support:

- Query Reply (Partitions) structured field
  - with scroll flag(s) set
- Set Window Origin structured field

MULTIPLE PARTITIONS (MPARS) (NOTE 2)

In addition to the partitions functions found in previous sets, the device must support:

- Query Reply (Partitions) structured field
  - with n>1
- Activate Partition structured field
- Destroy Partition structured field

16-BIT ADDRESSING (16BIT)

In addition to the partitions functions found in previous sets, the device must support these structured fields and orders (with 16-bit addressing):

- Query Reply (Usable Area) structured field
- SBA, RA, and EUA orders
- Create Partition structured field

BASIC PARTITIONS (BPARS)

If this set is implemented, the device must support:

- Query Reply (Partitions)
  - with the unbuffered and protected partition flags set on
- Create Partition structured field
  - parameters for unbuffered and protected partitions.
- The following:
  - Read Buffer, Read Modified, Read Modified All commands
  - Erase All Unprotected command
  - Insert Cursor order
  - Program Tab order
  - Erase Unprotected to Address order
  - Modify Field order

FORMAT PRESENTATION (FORMT)

If this set is implemented, the device must support:

- Query Reply (Format Presentation) structured field
- Select Format Group structured field
- Present Absolute Format structured field
- Present Relative Format structured field

MSR CONTROL (MSRCTL)

If this set is implemented, the device must support:

- Query Reply (MS Device Control) structured field (with TYPE = X'01')
- MS Device Control structured field (with TYPE = X'01')

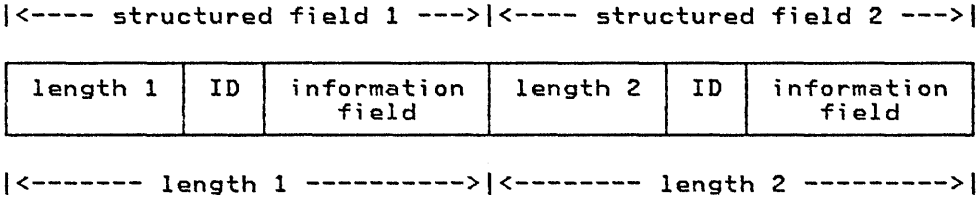
MAGNETIC ENCODER (MENCD)

If this set is implemented, the device must support:

- Query Reply (MS Device Control) structured field (with TYPE = X'02')
- MS Device Control structured field (with TYPE = X'02')

This chapter contains structured fields used by the SNA components described in this book, except for those structured fields used exclusively by the SNA 3270 data stream. Those readers who are using the 3270 data stream should see 3270 Data Stream Programmer's Reference for all structured field formats for that data stream.

The syntax for structured fields permits variable-length data and controls to be encoded in such a way that the receiver of a data stream can decompose a sequence of fields into its component fields without having to scan every byte. Variable length structured fields are achieved by providing a length as the first parameter of the structured field, thus:



Note: Some structured fields may specify a value of zero in the length field. A length of zero indicates that:

- The length of the structure was not determined prior to transmission.
- This is the last or only structured field in the request unit (RU).
- The length of the structured field should be determined using a communication-control function, for example, the RU's EC (end chain) indicator.

A length field containing zero is restricted to those structured fields that have "XX" as the second byte of the ID in following table.

Structured fields do not span chains; that is, it is an error condition (length error) if the chain ends before a structured field's length count is satisfied. Also, control sequences (for example, set attribute) do not span structured fields; it is an error condition if the entire control sequence does not appear in the same structured field.

The structured fields in this chapter are described in alphabetical order. The following list shows them in hexadecimal order by their identification (ID) field.

ID	NAME
01 XX	Read Partition - Query
06 XX	Load Programmed Symbols
10 30	Request Recovery Data
10 31	Recovery Data
10 32	Set Checkpoint Interval
10 33	Restart
41 XX	SCS Data
81 81	Query Reply (Usable Area)
81 85	Query Reply (Character Sets)
81 86	Query Reply (Color)
81 87	Query Reply (Highlight)
81 A0	Query Reply (Device Characteristics)

Each of these structured fields is defined below.

Structured field type codes not defined are reserved and are rejected (SNA sense code X'1003'). Unless specifically stated to the contrary, any

bits/fields classified as reserved must be checked for zero value; nonzero values are rejected (sense code X'1003').

SNA sense codes associated with structured fields are listed on page 184.

LOAD PROGRAMMED SYMBOLS (LOAD PS)

FUNCTION

Defines additional coded graphic character sets, and then loads that symbol- and character-definition data into a specified storage area in the device.

This structured field causes one or more characters to be loaded into contiguously-addressable slots in the specified storage area. Each storage area contains 191 contiguously-addressable slots, which are associated with data stream code points X'40' through X'FE'. The slot associated with X'40' cannot be loaded, and contains a blank.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2		X'06'	Load Programmed Symbols (Load PS)
3		FLAGS	
	0	BASIC/EXT. B'0' B'1'	Basic or extended form - Basic form (only bytes 0 through 6 present) - Extended form (length determined by byte 7)
	1	CLEAR B'0' B'1'	Clear PS-set storage area - Do not clear PS-set storage area - Clear the specified PS-set storage area prior to loading the character data in this structured field
	2	SKIP B'0' B'1'	Skip suppress (See Note 2) - Suppression not required - Suppression required
	(3-7) 3 4-7	TYPE X'5' X'6' others	PS data format type: Reserved Format: - Column loading (from left to right) - Type X'5' compressed - Reserved
4		LCID	Local character set ID - Use X'40' to X'EF'. - Use X'FF' to indicate that RWS (read-write storage) associated with this LCID is free (not assigned). - Other values are reserved.
5		CHAR	Beginning code point - X'41 through X'FE'
6		RWS	PS-set storage number (X'02' through X'07') in RWS (read-write storage)

Bytes 7 to 12 are parameters for the extended form of Load PS.

BYTE	BIT	CONTENT	MEANING
7		P.LENGTH	Length of parameters for extended form, including the length parameter. The parameters defined below are progressively included by specifying the appropriate length. Omitted parameters are assumed to have the architecturally defined default indicated by the zero value for that parameter.
8	0	APA B'0' B'1'	All points available - All points available - Not all points available
	1	CB B'0' B'1'	LCID compare - LCID compare - No LCID compare
	2	OB B'0' B'1'	Operator selectable by PS key - Operator selectable - Not operator selectable
	3-7	Reserved	Must be zero
9		LW	Number of X-units in character cell
10		LH	Number of Y-units in character cell
11			Reserved
12	0-4	Reserved	Must be zero
	5-7	COLOR B'000' B'001' B'010' B'100' other	Color planes - All planes - Blue plane - Red plane - Green plane - Reserved

Note 1: Values for parameters other than those shown above are rejected; invalid parameter values are also rejected. (See "Sense Code Table" on page 184.)

Note 2: The lost space resulting from setting of the skip suppress to B'1' is made up following the last display/print line. That is, the last display/print line moves up from the bottom of the physical presentation space. For a printer, setting of skip suppress to B'1' must not result in loss of forms sync following a form feed (FF) or forms overflow.

The PS data follows in the format described in bits 3-7 of byte 3.

BYTE	BIT	CONTENT	MEANING
m-n		DATA	PS data (character-definition data)

OPERATION

1. The extended-form bit in byte 3 indicates whether this structured field is of the extended form, that is, contains bytes 7 through 12. The extended form contains additional information associated with copy operations, character cell size, and color. If the half-session does not support the extended form and this bit is set to 1, the data stream is rejected (invalid parameter, sense code X'1005').

2. If skip suppression is on, any row on the usable area containing characters from this character set will have the skip suppressed that would normally follow that row. The characters on the following row will then be vertically adjacent to the characters on the current row. If the skip suppress flag is not set, then no skip suppression is required for characters from this character set.

3. The RWS number indicates the physical RWS to be loaded. There is a fixed relationship between the physical RWS number and the attribute selection keys defined for PS. The value in CHAR indicates the first slot to be loaded, and must be in the range X'41' through X'FE'. If CHAR < X'41' or CHAR > X'FE', the data stream is rejected (invalid parameter, sense code X'1005').

   If the CLEAR flag is set on, all slots in the specified RWS are cleared, then the character definitions in the data portion of the structured field, interpreted according to the data format type (byte 3), are loaded into contiguously-addressable slots in the PS RWS, starting at the position defined by CHAR.

   If the data type specifies compressed data (type 6), the data is first decompressed. (The compression-decompression function is discussed in the literature published by products that have implemented Load PS.)

   For data of type 5 or type 6, each set of LH contiguous bits defines one column of a character cell. LW such contiguous columns define a character as a LW by LH matrix of dots, where LW is the width of the character cell and LH is the height of the character cell. If LW and LH are not supplied in the extended form, or if they are supplied and set to zero, the values of LW and LH are determined by the device.

   Loading of character definitions continues until either:

   - The data is exhausted. In this case, the last complete cell definition in the data is loaded. If there are any excess bits, they are ignored and a negative response (sense code X'1005') is returned.
   - The slot corresponding to X'FE' is loaded. Excess data is ignored and a negative response (sense code X'1005') is returned.

4. The LCID is a one-byte code identical to the value byte definition for a character set identified by a set attribute control in an SNA character string. The LCID specified in this structured field is released from any previously associated PS RWS number. The PS RWS number released is assigned LCID = X'FF'. This is not done until the LCID specified in this structured field is assigned to the associated RAM.

5. The RWS (read-write storage) number specified in this structured field is released from any previously assigned LCID. Any subsequent reference to this released LCID will be an error condition.

6. An LCID of X'FF' indicates that this PS RWS is free (not assigned). In a printer, a PS RWS with LCID = X'FF' may be overlaid when PS data is copied.

   On a Load PS, where an error condition is detected before the contents of the associated RWS is altered, a negative response is sent but the LCID is not changed. Where the error condition occurs during the update of the RAM, the negative response is sent and the LCID is updated to that specified in Load PS. The above applies to a sequence of Load PS structured fields sent after an FMH-1 that specifies that structured fields follow. Also, any sequence of Load PS structured fields following the failing Load PS is not executed.

7. The APA bit, when set to 1 (not APA), implies that fewer than all points may be displayed or printed to allow a performance gain for specific devices.

8. When the CB bit is set to 0, and a local copy is initiated, the LCID of this character set is compared with character set LCIDs in the printer to determine whether or not there is a match. If the LCIDs match and the CB bits are both zero, the copy operation is performed using the correspond-

ing LCID in the printer. If not, characters from the base character set of the printer are used.

When the CB bit is set to 1, the LCID of this character set is not compared with LCIDs of character sets in the printer. The copy operation is performed using the base character set of the printer.

When the printer can accept character sets from the display during a copy operation, and there is no corresponding LCID in the printer, the character set may be copied independent of the setting of the CB bit.

9. When the OB bit is set to 1, it signifies that this character set is intended for output only. Thus, the PS selection key normally associated with the RWS containing this character set cannot be enabled by the terminal operator while the key is connected to this LCID. Selection of individual character sets can thus be disabled, even though the Set Reply Mode structured field allows character set selection. When the Set Reply Mode structured field disables character set selection, selection is disabled for all character sets, independent of the setting of the OB bit for each character set.

10. If LW and LH for the character cell dimensions are specified and are nonzero, and LW is not more than the maximum character cell width supported by the device, and LH is not more than the maximum character cell height supported by the device, then the character cells defined by LW and LH are loaded into the specified PS RWS so that the first bit addresses the upper left-hand corner of the character slot.

    If either LW or LH exceeds the maximum slot size supported, the structured field is rejected (invalid parameter, sense code X'1005').

    If LW and LH are not specified or are set to zeros, a device default character cell size is used to determine the character cell size in the PS data.

11. A character set with triple-plane capability has three color planes into which characters may be loaded. For any code point (X'41' to X'FE') within the character set, each plane may be loaded independently; that is, a different character definition may be loaded into each of the color planes in the character set for that code point.

    For a triple-plane character set, if the COLOR field is B'001', B'010', or B'100', the character set data is loaded only into the specified character slots in that plane. Other COLOR values are reserved and rejected.

    For a triple-plane character set, if the COLOR field is B'000', the character set data is loaded into the specified character slots in all three planes.

    For a single-plane character set, if the COLOR field is B'001', B'010', or B'100', the data stream is rejected.

QUERY REPLY

FUNCTION

Transmits a response to a query request. The query request is sent in the Read Partition structured field. (See "Read Partition" on page 177.)

FORMAT

BYTE	CONTENT	MEANING
0-1	L	Length of this structure
2	X'81'	Query Reply
3	REPLY CODE	Reply identifying code: X'81' = Usable area X'85' = Character set X'86' = Color X'87' = Highlight X'A0' = Device characteristics
4-n	PLIST	Parameter list

OPERATION

In reply to a query, the device transmits inbound a set of structured fields that describe the device features.

Each structured field has an ID of X'81nn', where nn is the reply code identifying the feature.

PLIST is a variable-length parameter list. For the parameter list for each reply code, see the specific query reply in the definitions that follow.

QUERY REPLY (CHARACTER SETS)

FUNCTION

Transmits information about (1) each character set supported, and (2) the ability of the device to support:

- The Load PS structured field
- A graphic escape character

The SLU sends this form of query reply when responding to a Read Partition structured field indicating query.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'85'	Character sets
4		FLAGS	
	0	ALT	Graphic escape (GE):
		B'0'	GE not supported
		B'1'	GE supported
	1	RES	Reserved
	2	MULTI	Multiple character set capability:
		B'0'	Load PS not supported
		B'1'	Load PS supported
	3	EXT	Extended form of Load PS:
		B'0'	Load PS extended form not supported
		B'1'	Load PS extended form supported
	4	MS	Matrix size
		B'0'	Fixed-size matrix
		B'1'	Variable-size matrix
	5-7	RES	Reserved
5		RES	Reserved
6		SDW	Default matrix width for Load PS
7		SDH	Default matrix height for Load PS
8-11		FORM	Supported Load PS format types; bit encoded (If bit i = 1, then type i supported)
12		DL	Length of each descriptor

The base part is followed by one or more character set descriptors, each defining the characteristics of one character set located in either read-only storage or read-write storage. The length of each descriptor is contained in DL. Its format is:

1		SET	Device-specific character set ID (number of the read-only or read-write storage)
2		FLAGS	
	0	LOAD B'0' B'1'	Loadable: Nonloadable character set Loadable character set
	1	TRIPLE B'0' B'1'	Triple plane: Single-plane character set Triple-plane character set
	2	CHAR B'0' B'1'	Character representation: One-byte coded character set Two-byte coded character set
	3	CB B'0' B'1'	Compare bit: LCID compare No LCID compare
	4-7	RES	Reserved
3		LCID	Local character set ID
4-15		RES	Reserved

OPERATION

The parameters in the base describe the device's capability to support character sets, as follows:

- The following description of the FLAGS byte in the base (byte 4) refers to the function sets supported by the device.

  ALT indicates that the device subset includes the ALT node.

  MULTI indicates that the device subset includes the MULTI node.

  EXT indicates that the device supports the extended form of the Load PS structured field. Thus, EXT = B'1' only if MULTI = B'1'.

  MS indicates that the device supports a matrix size that is related to the character set size, that is, the parameters SW and SH are present in each descriptor. If the device supports nonmatrix characters (as indicated in byte 5 bit 1 of Query Reply (Usable Area)), bytes 6 and 7 of the base (SDW and SDH) are not applicable and must be set to zero. As with EXT, MS = B'1' only if MULTI = B'1'.

- SDW and SDH define the default matrix size. (See also SW and SH, below.)

- FORM defines the Load PS format types supported by the device. This field is a 32-bit field. Each bit corresponds to a format type that can be specified in Load PS (See TYPE field of Load PS on page 164.) Thus, if bit i = 1 in FORM, the device supports format type i in Load PS.

  Currently, Load PS format types 5 and 6 are defined for LU_T1. Bit values B'11111..1' of the first byte of the field are reserved, as are bytes 2 through 4.

- DL defines the length of each descriptor (in bytes).

The parameters in each descriptor define the characteristics of a character set, as follows:

- SET defines the device specific ID. For a loadable set, this is the value specified in byte 6 of Load PS. SET also defines the key that can be used by the operator to select this character set.

- LOAD indicates that this character set is loadable.

- TRIPLE indicates that this character set has one or three planes.
- CHAR indicates that this is a 1- or 2-byte coded character set.
- CB indicates whether this character set can be compared for copy. (See also the CB field and subsequent discussions in Load PS on page 164.)
- LCID identifies the local character set ID currently connected to this character set. The ID may not identify uniquely the contents. A value of X'FF' indicates that this PS set is free (not assigned). LOAD PS allows an LCID to be connected to a loadable character set. The LCID is then the data stream value used in SA SNA character string controls to identify characters from this set. Nonloadable character sets have a default LCID assigned to them by the device in the range X'F0' through X'FE'.

QUERY REPLY (COLOR)

FUNCTION

Transmits the color properties of the device if the device accepts or supports some set of color attribute values.

The device sends this form of query reply when responding to a Read Partition structured field indicating query.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'86'	Color
4		FLAGS	
	0		Reserved
	1		
		B'0'	Printer only - A one-color ribbon is not loaded
		B'1'	Printer only - A one-color ribbon is loaded
	2-7	Reserved	
5		N	Length of color attribute list (N = number of CAV/COLOR pairs)
m		CAV(n)	Color attribute value accepted by the device
m+1		CID(n)	Color identifier of the color displayed/printed for CAV(n). Color identifiers are:
		X'00'	Default color
		X'F1'	Blue
		X'F2'	Red
		X'F3'	Pink
		X'F4'	Green
		X'F5'	Turquoise
		X'F6'	Yellow
		X'F7'	Neutral
		X'F8'	Orange
		X'F9'	Black
		X'FA'	White

OPERATION

The parameters CAV(n) are all those color attribute values that are accepted by the device without causing a negative response.

The parameters CID(n) identify the colors that are displayed or printed by the device for each of the accepted color attribute values. The device must either display the color whose architected color identifier is the same as the color attribute value or display the device default color. Except for CAV(n) = X'00', translation by the device of a color attribute value to a color different from that architected for the specific color attribute value is not allowed, that is, the value of CID(n) can only be equal to the value of CAV(n) or to X'00'. No other values are allowed.

The CAV/CID pairs are repeated for each of the data stream values accepted by the device.

The color associated with the CAV(n) value of X'F7' defines the default color that is displayed/printed when a single plane character set is referenced; the associated CID(n) value may be any of the values in CAV(n) including X'00'. The value X'F7' for CID(n) indicates the default color is neutral. Neutral means black or white.

The CAV(n) value of X'00' may have an associated CID(n) value of any of the defined values except X'00'.

All devices that send Query Reply (Color) are required to have the values CAV1 = X'00', CID1 = value associated with the the device default color, as the first entry in the CAV/CID pairs list.

Query Reply (Color) Example:

CAV(n) ATTR VALUE	CID(n) COLOR IDENTIFIER
X'00'	X'F7'
X'F1'	X'F1'
X'F2'	X'F2'
X'F3'	X'00'
X'F4'	X'F4'
X'F5'	X'00'
X'F6'	X'00'
X'F7'	X'00'

QUERY REPLY (DEVICE CHARACTERISTICS)

FUNCTION

Transmits the device's ability to support SNA Character String (SCS) functions.

The device sends this form of query reply when responding to a Read Partition structured field indicating query.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'A0'	Device Characteristics
4-n			Function descriptor structured fields

The base part is followed by a descriptor:

Set Print Density (SPD) descriptor

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2-3		X'1104'	SCS X'2B' identifier
4-5		X'D229'	Set Print Density descriptor
6		X'00'	SPD is supported, default value is used
7-8		X'00nn'	Default value for print density (for example, X'000A' is 10 characters per inch)
9		X'02'	CD (character density) parameter length
10		X'60'	The CD parameter is present sometimes, it only takes discrete values, and it appears only one time
11		X'nn'	Number of discrete values supported by the device
12-n			The discrete CD values supported

OPERATION

The function descriptor is included in Query Reply (Device Characteristic) to indicate how an SCS function is supported.

QUERY REPLY (HIGHLIGHT)

FUNCTION

Transmits the types of highlighting supported by the device.

The device sends this form of Query Reply when responding to a Read Partition structured field indicating query.

FORMAT

BYTE	CONTENT	MEANING
0-1	L	Length of this structure
2	X'81'	Query reply
3	X'87'	Highlight
4	N	Number of attribute-value/action pairs
n	Vi	Data stream attribute value accepted
n+1	Ai	Data stream action

OPERATION

If a device accepts the highlight attribute type, then it must accept attribute value X'00' (default specification). It may optionally accept other attribute values. All accepted values are listed in the query reply. For each accepted value, the query reply lists the device action in replies of 1-byte action codes. The action codes (Ai) are:

X'F0' Normal (no highlight)
X'F1' Blink
X'F2' Reverse video
X'F4' Underscore

In addition to the above, the action code may specify X'00'. The code X'00' indicates that the device action for the corresponding attribute value is the same as the action for the attribute value X'00', that is, the default action of the device.

An example of Query Reply (Highlight) is:

BYTE	CONTENT	MEANING
0-1	X'000C'	Length
2	X'81'	Query reply
3	X'87'	Highlight
4	X'04'	Number of pairs
5	X'00'	Attribute value (default)
6	X'F0'	Normal (no highlighting)
7	X'F1'	Attribute value
8	X'00'	Action - default
9	X'F2'	Attribute value
10	X'F2'	Action - reverse video
11	X'F4'	Attribute value
12	X'F4'	Action - underscore

QUERY REPLY (USABLE AREA)

FUNCTION

Transmits the size and characteristics of the screen or page.

The SLU sends this form of query reply when responding to a Read Partition structured field indicating query.

The screen or page is comprised of a specific number of cells. The number of these cells and their size is specified in this query reply.

A character may be placed in each cell. This character may be taken from any of the character sets defined in Query Reply (Character Sets).

Note: A device may be implemented to allow the cell size to be controlled by the host (the host specifies the cell size in a Create Partition structured field). In this case, the host-specified value overrides the default described above.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2		X'81'	Query reply
3		X'81'	Usable Area
4		FLAGS	
	0-1	RES	Reserved.
	2	EXSF	Structured field usage:
		B'0'	SCS Data supported
		B'1'	SCS Data not supported
	3	HC	Hard copy characteristics:
		B'0'	Not a hard copy device
		B'1'	A hard copy device
	4-7	ADDR	Addressing mode
		X'F'	Unmapped (no explicit addressing)
		Other	Other values reserved.
5		FLAGS	
	0	RES	Reserved
	1	CHAR	Characters
		B'0'	Matrix characters
		B'1'	Nonmatrix characters (for example, a belt printer)
	2-7	RES	Reserved
6-7		W	Width of usable area in cells
8-9		H	Height of usable area in cells
10		UNITS	Unit of measure:
		X'00'	Inches
		X'01'	Millimeters
11-14		Xr	Distance between points in x direction as a fraction, measured in units: 2-byte numerator 2-byte denominator
15-18		Yr	Distance between points in y direction as a fraction, measured in units: 2-byte numerator 2-byte denominator
19		AW	No of x-units in default cell
20		AH	No of y-units in default cell

OPERATION

EXSF indicates whether structured field support has been configured and is executable.

HC indicates that this is a hard-copy device, that is, a printer.

CHAR indicates matrix or nonmatrix characters. If CHAR = B'1' (nonmatrix characters), bytes 10-20 are not applicable and must be set to zeros.

W and H define the size of the usable area in cells of size AW x AH (see also AW, AH). Thus, the total size of the usable area in points is W x AW by H x AH.

The primary unit of measure is always character cells. Thus, W x H defines the maximum number of characters that can be presented in the usable area, when characters from a nonloadable set are being use.

For a printer, W and H are the maximum print position (MPP) and the maximum print line (MPL) supported by the hardware. These values correspond to the MPP and MPL values in the SNA character string (SCS). On some devices, MPP and MPL may be set by the operator. The values returned in this query reply refer to the hardware's capability, not to the current setting.

UNITS, Xr and Yr define the spacing granularity. For example, if the device has 72.5 points/inch horizontally and 69 points/inch vertically, the the values would be:

```
UNITS - X'00'
Xr    - X'00020091' (that is, 2/145 decimal)
Yr    - X'00010045' (that is, 1/69 decimal)
```

In combination with Load PS, the application can use these parameters to present data where precise dimensions are critical. Alternatively, if the relative dimension (x/y aspect ratio) is critical, the ratio Xr/Yr allows this relationship to be controlled.

AW and AH define the default cell size for the device. They specify the smallest cell size available for the presentation of characters from any nonloadable character set.

Generally, the values for Xr, Yr, AW, and AH are fixed for a given implementation. If the values can be changed, such as by an operator-actuated switch operation, the values defined are those in effect when a Read Partition (Query) is received.

READ PARTITION

FUNCTION

Queries the device as to its capabilities.

FORMAT

BYTE	CONTENT	MEANING
0-1	L	Length of this structure
2	X'01'	Read Partition
3-4	X'FF02'	This read is a query operation

OPERATION

A set of codes is transmitted to the sender of the Read Partition structured field. These codes describe the features on the device. (See "Sense Code Table" on page 184 for error conditions.)

RECOVERY DATA

FUNCTION

Sent from SLU to PLU in response to a Request Recovery Data structured field. It contains the recovery data needed by the PLU to recover from the error.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2-3		X'1031'	Recovery Data
4			Reserved
5		Flags	
	0		0 - Vertical format not needed on Restart. 1 - Vertical format needed on Restart.
	1		0 - Horizontal format not needed on Restart. 1 - Horizontal needed on Restart.
	2-7		Reserved
6		SLD	SLD (set line density) parameter in effect at the checkpoint. The default value (X'00') indicates the parameter is not to be reinitialized at restart.
7		Char.Set	Character set parameter of SA (set attribute) control in effect at the checkpoint. The default value (X'00') indicates the parameter is not to be reinitialized at restart.
8-11		Vertical	Byte offset from Set Checkpoint Interval structured field to the SVF (set vertical format) control in effect for the checkpoint. If the SVF control is within a parameterized string the counter points to the control code of the parameter. It is set to zero (0) if no SVF control code is received.
12-13		V-offset	Byte offset within the SCB (string control byte) string or structured field of the SVF character. The offset is to the uncompressed and uncompacted character. It is set to zero (0) if the SVF character is not within an SCB string or structured field.
14-15		V-seq'ce	RU sequence number of the RU containing the SVF character.
16-17		V-length	Length of the SVF character string required for restart.

BYTE	BIT	CONTENT	MEANING
18-19		SPD	SPD (set print density) parameter in effect at the checkpoint. The default value (X'0000') indicates the parameter is not to be reinitialized at restart.
20-23		Horizon.	Same as vertical for SHF (set horizontal format).
24-25		H-offset	Same as V-offset for SHF.
26-27		H-seq'ce	Same as V-sequence for SHF.
28-29		H-length	Same as V-length for SHF.
30		Color	Color parameter of SA (set attribute) in effect at the checkpoint. The default value (X'00') indicates the parameter is not to be reinitialized at restart.
31		Hilite	Highlight parameter of SA in effect at the checkpoint. The default value (X'00') indicates the parameter is not to be reinitialized at restart.
32-33		Pages	Number of pages printed since the checkpoint.
34-35		Lines	Number of lines printed on the page with the error.
36-39		Chkpoint	Byte offset from Set Checkpoint Interval structured field to the first character after the (code point or character) that caused an eject to the checkpointed page. If the character is within a parameterized string the counter points to the control code of the parameters.
40-41		C-offset	Byte offset within the SCB string or structured field of the checkpointed character. The offset is to the uncompressed and uncompacted character. It is set to zero (0) if the checkpointed character is not within an SCB string or structured field. If the character is within a parameterized SCS control code this is an offset to the SCS control code.
42-43		C-seq	RU sequence number of the RU containing the checkpointed character.
44-45		C-Seqoff	Byte offset within the RU of the checkpointed character.
46-47		C-SCSoff	Byte offset within the parameterized SCS control code (for example, TRN) of the checkpointed character. It is set to zero (0) if the checkpointed character is not within a parameterized SCS control code.
48		Prime	Prime compression character in effect at time of checkpoint.

OPERATION

- Pages

The pages indicate the number of pages that the SLU has printed since the checkpoint was taken that is being sent to the PLU.

- Lines

The lines indicate the number of lines that the SLU has printed since the beginning of the page on which the error had been detected.

- Checkpoint (Chkpoint)

The checkpoint field is a 4-byte binary counter that indicates the number of FM data bytes from the Set Checkpoint Interval structured field to the first code point after the code point that caused a page eject to the page for which a checkpoint is to be taken. That is, it counts all FM data bytes in the RUs following the latest Set Checkpoint Interval structured field. It then points to the first code point processed after the printer ejected to the top of the page for which the interval count requires a checkpoint. If the eject was caused by a parameterized string, the counter points to the control code of the parameters (for example, the immediately preceding SCB control code).

- C-offset

C-offset is a 2-byte binary counter that indicates the position within a compression/compaction SCB string or structured field of the actual checkpoint character. It points to the uncompacted/uncompressed position of the character, not the compressed/compacted position.

This field is zero (0) if the checkpoint field points to the exact checkpointed character.

- C-SCS-offset (C-SCSoff)

C-SCS-offset is a 2-byte binary counter that indicates the position within a parameterized SCS code of the checkpointed character.

This field is set to zero (0) if the checkpointed character is not within a parameterized SCS code.

- Vertical

The vertical field is a 4-byte binary counter that indicates the byte offset from the Set Checkpoint Interval structured field to the SVF code point in effect for this checkpoint. If no SVF code has been received this field is set to zero (0). If the SVF is inside a parameterized string, the counter points to the control code of the parameters (for example, the immediately preceding SCB control code.)

- V-offset

V-offset is a 2-byte binary counter that indicates the position within an SCB string or structured field of the SVF code. It points to the uncompacted/uncompressed position of the SVF.

This field is set to zero (0) if the vertical field points to the actual SVF code.

- Horizontal (Horizon.)

The horizontal field is a 4-byte binary counter that indicates the byte offset from the Set Checkpoint Interval structured field to the SHF code point in effect for this checkpoint. If no SHF has been received, this field is set to zero (0). If the SHF is inside a parameterized string, the counter points to the control code of the parameter (for example, the immediately preceding SCB control code).

- H-offset

H-offset is a 2-byte binary counter that indicates the position within an SCB string or structured field of the SHF code. It points to the uncompacted/uncompressed position of the SHF.

This field is set to zero (0) if the Horizontal Field points to the actual SHF code.

• Flags

Bit 0 of the flag field indicates whether or not the printer requires the SVF format to be present in the Restart structured field. If bit 0 is one (1), the primary must use the vertical field and V-offset to locate the SVF format and resend it in the Restart structured field chain. If bit 0 is zero (0), the primary does not have to send the SVF format as the printer can assure that the formats are still set properly.

Bit 1 of the flags field is the same as bit 0, but for SHF.

• Set Line Density (SLD)

This field is the 1-byte parameter field of the SLD in effect at the checkpoint. The default value (X'00') is used if SLD is not supported or has not changed since the checkpoint.

• C-seq

This field contains the RU sequence number of the RU containing the checkpointed character.

• C-seq-offset (C-seqoff)

This field contains the byte offset within the RU of the checkpointed character.

• V-sequence (V-seq'ce)

This field contains the RU sequence number of the RU that contains the SVF control code in effect at the checkpoint.

• H-sequence (H-seq'ce)

This field contains the RU sequence number of the RU that contains the SVF control code in effect at the checkpoint.

• V-length

This field contains the length of the SVF parameter string that must be returned in the Restart structured field. It includes the SVF control code and all parameters following.

• H-length

This field contains the length of the SHF parameter string that must be returned in the Restart structured field. It includes the SHF control code and all parameters following.

• Set Print Density (SPD)

This field is the 2-byte parameter field of the SPD in effect at the checkpoint. The default value (X'0000') is used if SPD is not supported or has not changed since the checkpoint.

• Color

This field is the 1-byte parameter field of the SA (set attribute) value for color in effect at the checkpoint. The default value (X'00') is used if SA for color is not supported or has not changed since the checkpoint.

• Character Set (Char.Set)

This field is the 1-byte parameter field of the SA (set attribute) value for character set in effect at the checkpoint. The default value (X'00') is used if SA for character set is not supported or has not changed since the checkpoint.

• Hilite

This field is the 1-byte parameter field of the SA value for highlighting in effect at the checkpoint. The default value (X'00') is used if SA for highlighting is not supported or has not changed since the checkpoint.

REQUEST RECOVERY DATA

FUNCTION

Sent from PLU to SLU to request recovery data for Print Job Restart.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2-3		X'1030'	Request Recovery Data
4			Reserved

OPERATION

This structured field must flow to enable the SLU to send the recovery data to the PLU.

RESTART

FUNCTION

Sent from PLU to SLU to indicate that the restart is in progress and that a certain number of pages and lines should be bypassed before printing starts again using the data that follows.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2-3		X'1033'	Restart
4			Reserved
5-6		Start Pg	Number of pages to skip on restart
7-8		Start Ln	Number of lines to skip on page for restart
9-n		SCS Data	SCS data (noncompressed and noncompacted) to set up for restart

OPERATION

The SCS data field must include the required SHF and/or SVF and other SCS data for restart. The first byte of FM data resumes at the checkpoint spot or at the start of the SCB string or structured field containing the checkpoint spot in the following RU chain.

- Start Page (Start Pg)

The start page is the indication sent by the PLU to the SLU of the number of pages that are to be bypassed prior to printing during a restart operation.

- Start Line (Start Ln)

The start line is the indication sent by the PLU to the SLU of the number of lines on the starting page that are to be bypassed prior to printing during a restart operation.

- SCS Data

SCS data is sent in the Restart structured field to reestablish various parameters to the state at the time of the checkpoint. For example, the SVF and SHF codes indicated by the vertical and horizontal offsets in the Recovery Data structured field. Any SCS codes (control or graphic) may be included. Counts are reset to those at the time of the checkpoint after processing the SCS codes within the structured field.

SCS DATA

FUNCTION

Allows SNA character string (SCS) controls and data to be intermixed with structured fields.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2		X'41'	SCS Data
3		X'00'	Reserved
4-n		Data	SCS controls and data

OPERATION

The SCS data portion of this structured field is treated the same as SCS data not delimited by an SCS Data structured field. (For details on SCS controls, see Part 2 Chapter 1.)

SET CHECKPOINT INTERVAL

FUNCTION

Passes from PLU to SLU the number of pages that are to be in the interval between checkpoints. It contains the recovery data needed by the PLU to recover from the error.

FORMAT

BYTE	BIT	CONTENT	MEANING
0-1		L	Length of this structure
2-3		X'1032'	Set Checkpoint Interval
4			Reserved
5-6		Interval	Checkpoint interval: Number of pages to be printed between checkpoints. A zero value indicates that checkpoints are not to be taken.

OPERATION

Interval specifies the number of pages in the interval between SLU checkpoints. This number is set in the SLU by the PLU's use of the Set Checkpoint Interval structured field; it is a 2-byte value to enable checkpointing of jobs that use short forms without taking an excessive number of checkpoints.

This structured field resets all previous checkpointed information. The checkpoint counters begin with the first FM data byte following the structured field.

SENSE CODE TABLE

	SNA SENSE CODES (HEX)			
	1003	1005	0863	OTHER
STRUCTURED FIELDS (all struc. flds.)				
Invalid structured field type	X			
Missing structured field parameter		X		
Reserved field is not zero	X			
LOAD PROGRAMMED SYMBOLS				
Incorrect length		X		
Invalid data type	X			
Invalid LCID (Values X'40' thru X'EF' are not considered invalid)	X			
CHAR < X'41' or > X'FE'		X		
RWS not in range X'02'-X'07'				084C
RWS is valid but not installed				084C
EXTN not supported		X		
P-length incorrect		X		
Byte 8, bits 3-7 not zero	X			
Invalid LH or LW		X		
Byte 12, bits 5-7 invalid				084C
Excess bits in data		X		
Compressed data terminator incorrectly specified		X		
PS resource not available				084C
READ PARTITION				
Incorrect length		X		
Invalid read operation code	X			
Read Partition not last structured field in chain		X		
Chain containing Read Partition does not specify CD				0829
Chain containing Read Partition does specify EB				0829
SCS DATA				
Incorrect length		X		

This chapter explains how function management headers (FMHs) are used to exchange information between session partners and how that information controls the activities performed by the LU (logical unit). It also defines the formats of the FM headers and shows how those FM headers are exchanged with the RU (request/response unit).

This chapter need not be read by persons implementing LU-LU session types 2 and 3. Those persons implementing LU-LU session types 1 and 4 should read the sections covering FMH-1 through FMH-3. Those persons implementing LU-LU session type 6 should read the sections covering FMH-4 through FMH-10.

The RU is that part of the PIU (path information unit) that carries control information and end-user data between half-sessions. A control RU contains a request or an acknowledgment. A data RU may contain function management headers (FMHs), string control bytes (SCBs), data, or any combination of the three. The data can be any data stream allowed by the LU-LU session type. Examples of RU formats are shown in Figure 4-1.

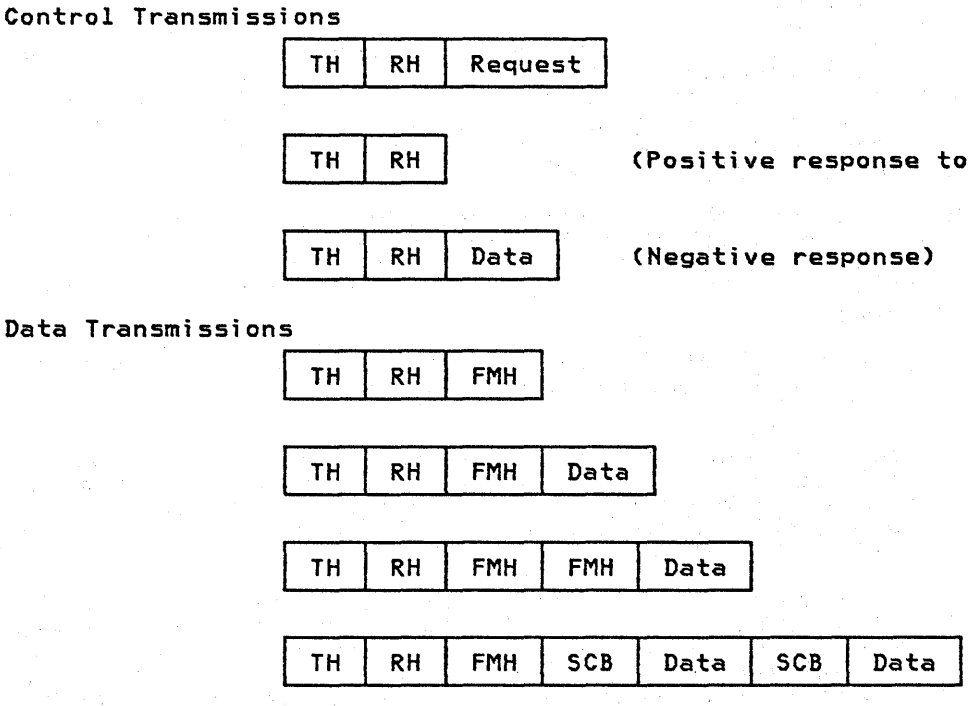


Figure 4-1. Examples of RU formats. The PIU is composed of the TH (transmission header), RH (request/response header), and RU (request/response unit). Although the use is not exclusive, path control (PC) uses the TH, transmission control (TC) and data flow control (DFC) use the RH, and session presentation services (SPS) and end users use the RU. The data formats are representative; not all formats are shown.

TYPES OF FUNCTION MANAGEMENT HEADERS (FMH)

Function management headers (FM headers or FMHs) enable an LU to send a data stream to a specific destination and control the way the data is presented at the destination. FM headers are the mechanism that one LU uses to select some of the functions it wants the presentation-services components of its session partner to perform.

FM headers are used in LU-LU session types 1, 4, and 6; they are not used in the SSCP-LU session or by LU session types 2 and 3.

The BIND request's presentation services usage parameters indicate whether the LU is allowed to send FM headers in the RU. These parameters also define which optional FMH functions are permitted in the LU-LU session. Since FMH functions differ between session types, you should review the "FM Header Processing" section and the BIND request description in the chapters that describe LU-LU session types 1, 4, and 6.

The request header (RH) contains a format indicator (FI) that, when on, indicates that an FMH is at the beginning of the RU. The format indicator is set on in the RH of the first RU in the chain. If more than one FM header is present in the RU, a concatenation flag is set on in each FMH that has another FMH following it. The rules that must be followed are described under "Rules for FMH-1, FMH-2, and FMH-3" below.

The FM headers that can be used by LU-LU sessions are:

FMH-1: FMH-1 is used to select a destination within an LU. A destination is the medium on which data is presented; it may be a device, a data set residing on a device, or a data stream. The medium select and destination name fields within the header identify the destination. The fields can be used alone or in combination as follows:

- Medium select field (byte 2). Selects a medium (a device type and a particular logical component) within the LU. The LU may or may not have a physical component that matches the medium selected. The absence of a console, for example, does not preclude the sending of an FMH-1 selecting a console. In this case, the receiving LU may elect to direct the data to a printer or disk. The receiving LU is responsible for taking the appropriate action.

  Each medium has a default data stream. Unless you specify a different data stream in another field of the FMH-1, the medium select field defines the data stream to be used. (For allowable data streams, see "Function Management Header Type 1 (FMH-1)" below, then see Part 1 Chapters 2 and 5 for limitations on their use.)

- Destination name field (bytes 9-n). Selects a logical component (either alone or in combination with the medium select field), which implies a data stream.

The physical device that is selected by the receiving LU is transparent to the session partner, but the session partner can influence the selection by using the demand select indicator in byte 3 of the FMH-1. For example, an LU may choose to use the indicator to prevent the session partner from spooling the data sent after the FMH-1.

The FMH-1 is used to initiate and conclude data traffic to a destination. It is also used to interrupt previously started data traffic (for example, to send a message to the console operator) and to resume it at a later time. Also, it can be used to indicate that the characteristics of the data stream are changing, but that the destination remains the same. Finally, the FMH-1 identifies whether an SCB follows.

FMH-2: FMH-2 specifies the data management activities to be performed at the destination selected with the FMH-1. Typical data management activities are adding and replacing records, creating and deleting data sets, and providing status information. An FMH-2 may be concatenated to an FMH-1. When it is, the FMH concatenation (FMHC) field of the FMH-1 is set to one to indicate that another FMH follows.

FMH-3: FMH-3 carries information that relates to all destinations of both session partners. For example, a compaction table to compact and decompact data can be sent as an FMH-3 if the table pertains to all destinations within the session.

FMH-4: FMH-4 carries information for a transaction program called logical message services. The FMH-4 tells the receiver what to do with a block of data within a message, as defined by logical message services.

FMH-5: FMH-5 identifies the transaction program that is to be attached to the receiving half-session. For example, a transaction program at the sending half-session sends an FMH-5 to tell the receiving half-session the name of the transaction program it wants to talk to.

FMH-6: FMH-6 carries information that relates to work being done by the transaction programs at each end of the session. For example, a transaction program sends an FMH-6 to identify the queue on which it wants the subsequent data to be placed.

FMH-7: FMH-7 carries information that relates to a previous error on the session. For example, an FMH-7 and error information are sent when a session partner detects certain error conditions.

FMH-10: FMH-10 requests that the receiver prepare for a sync point. The receiving session partner, on the next flow, either requests a sync point or aborts the unit of work.

The next section in this chapter discusses the rules for using FM headers. You should use Figure 4-2 to decide which of the FMH descriptions you need to read for the LU-LU session type you are using.

LU-LU session type	Types of FM headers allowed
0	User defined
1	1, 2, 3
2	None
3	None
4	1, 2, 3
6	4, 5, 6, 7, 10

Figure 4-2. FM headers allowed by LU-LU session type

FM HEADERS FOR LU-LU SESSION TYPES 1 AND 4

USING FMH-1 FOR DESTINATION SELECTION

Function management header type 1 (FMH-1) has seven functions that can be used by an LU to control the direction of data movement with another LU. These functions are:

BEGIN	Selects a destination (called the "active" destination) at the receiving partner to which the sending partner will send data
SUSPEND	Interrupts, or suspends temporarily, data traffic to the active destination
RESUME	Causes the destination that was last suspended to again become the active destination
END	Ends data traffic to the active destination
BEGIN/END	Selects a destination, transmits data, and ends traffic in one operation
END-ABORT	Abnormally ends data traffic to the active destination
CONTINUE	Retains the active destination, but changes FMH-1 parameters describing the data to be transmitted

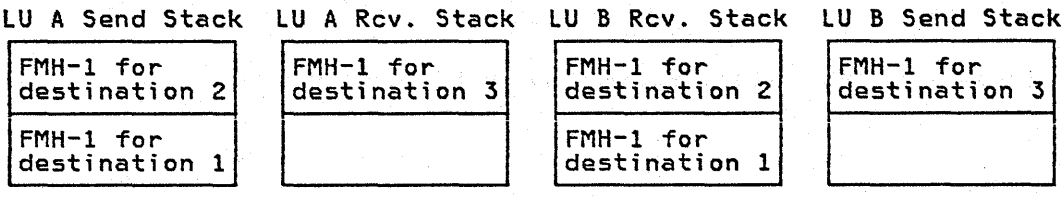
A default destination is assumed when FM headers are used by the session and data is sent or received without an FMH-1 (with no specific destination active). The default destination and data format are:

Medium = Console
Logical subaddress = 0
No destination name (DSNAME)
No compression
No compaction

The process of managing which destination is active is done by the session partners maintaining stacks, or push-down queues. A stack entry contains all the device control information needed to manage the device, or it points to that information.

A pair of stacks may be viewed as containing the destinations begun by one LU and received by the other. If LU A selects destination 1, then suspends it and selects destination 2, the stacks will appear:

A second pair of stacks contains the destination(s) begun by the session partner (LU B). When LU B begins a destination, the stacks contain:



The active destinations appear at the top of the stacks -- one active destination for each direction. All other entries in the stacks are called suspended destinations.

A parameter in BIND defines the depth of the stacks; that is, how many FMH-1s the stack can hold. If a one-level stack is selected, there can be one destination selected. The destination cannot be suspended; it must be ended before another destination is begun in that direction.

If a two-level stack is selected, the active destination can be suspended, and data sent to a second destination (as shown above). Suspending a print job and sending a message to an operator is an example of how a two-level stack can be used.

If a three-level stack is selected, two destinations can be suspended and data sent to a third destination.

Each half-session controls its own send stack and, through the FMH-1s it sends, its active destination. Either half-session can terminate session traffic abnormally. By using the stack reference indicator (SRI) in the FMH-1, an LU can identify either half-session's send stack and send an END-ABORT FMH-1 against the active destination in that stack. By using the end bracket indicator (EBI) in the RH, an LU can reset all stacks for the half-session.

The following destination selection operations are defined:

- The sender and receiver of a BEGIN write an entry in the first element of a stack. All other entries in the stack remain unchanged.

- The receiver of BEGIN activates all information contained in the FMH-1 and FMH-2 except:

  Length of header
  Concatenated or not
  Type of header

- The receiver of BEGIN/END writes an entry in the first element of a stack, activates all information contained in the FMH-1 and FMH-2, and after processing any data, erases all information.

- The sender and receiver of SUSPEND push down all entries in a stack by one element. The first element becomes empty; that is, it has no entry. The destination previously active becomes the second element.

- The receiver of SUSPEND saves all information that was activated by the previous BEGIN.

- The sender and receiver of RESUME pop up all entries in a stack by one element. The second entry, previously suspended, becomes active. The last entry in the stack, if present, is moved up one element, and the last element of the stack becomes empty.

- The receiver of RESUME restores all information that was saved from the previous SUSPEND.

- The receiver of END or END-ABORT erases all active information in the top entry of the stack.

- The sending or receiving of a CONTINUE has no effect on the stack.

- The receiver of CONTINUE overlays the information contained in the CMI, CPI, and ERCL fields of the FMH-1. (These fields are for compression, compaction, and exchange record length, respectively.) All other information remains active from the previous BEGIN.

- A stack is reset when all its elements are empty. In addition, the stacks are reset whenever the half-session enters in-brackets state. The stacks also are reset by ACTPU, DACTPU, ACTLU, DACTLU, BIND, SDT, CLEAR, and UNBIND.

The rules to be used for these operations are defined under "Rules for FMH-1, FMH-2, and FMH-3" on page 191.

USING FMH-2 FOR DATA MANAGEMENT

Frequently, an LU needs to be able to access data located at its session partner. Function management headers type 2 (FMH-2s) enable an LU to perform these data management operations. The FMH-2 is the way one LU tells another LU what it wants done at the active destination it specified in a previous FMH-1.

A variety of data set organizations are permitted:

- Sequential. Sequential access to all records

- Addressed direct. Relative data set using a record number for retrieval

- Keyed direct. Relative data set using a key for retrieval

- Keyed indexed. Indexed data set using a key for retrieval

An LU uses FMH-2s to create and modify these data sets. Changes can be performed on a record or a group of records. In addition, several FMH-2s allow the LU to send and receive status information about the data set.

In some cases, more than one FMH-2 can be used to perform an operation. One header can define the operation to be performed, while another tells where to perform it or provides a password. The FMH-2 that defines the operation is

the root FMH-2. The ones that identify where the operation is to be performed or provide the password are the extension FMH-2s.

Root and extension headers are shown in Figure 4-3.

Operation Type	FMH-1	Root FMH-2	Extension FMH-2
Destination selection	Begin Begin/end Resume		
Volume selection	Begin Begin/end		Volume ID, Password Volume ID, Password
Data set	(Note)	Create data set Scratch data set Scratch all data sets Erase data set records	Password Password Password Password
Record	(Note)	Add Add replicate Replace Replace replicate Erase	Record ID, Password Password Record ID, Password Record ID, Password Record ID, Password
Compression and compaction	(Note)	Compaction table Prime compression character	
Operation information	(Note)	Peripheral data information record	
Status	(Note)	Query for a data set Note Note reply	Password
Scheduling	(Note)	Execute program offline	
Record identification	(Note)		Record ID, Password

Note: A destination must be active, that is, a BEGIN, BEGIN/END, CONTINUE, or RESUME FMH-1 must have been the last FMH-1 issued. If no destination is active, the transmission goes to the console.

Figure 4-3. FMH-1 and FMH-2 relationship. Data management is accomplished using function management headers (FMHs). An FMH-1 is used to select a destination, then FMH-2s are used to process data at that destination.

In summary, this section has defined how an LU is able to access data in specific data sets using FMH-1 and FMH-2. As such, the LU sends and receives information from a single destination -- the active destination.

There is some information, however, that applies to all destinations in the session. To send and receive this common information, an LU uses function management headers type 3 (FMH-3s).

USING FMH-3 FOR DATA MANAGEMENT

Function management headers type 3 (FMH-3s) carry information that pertains to all destinations of the session. The FMH-3 format is identical to the FMH-2 format except that the FMH-3 does not have a stack reference indicator, having no need to identify a specific destination.

An FMH-3 cannot be concatenated to other headers; it must be sent as the first header in a chain. Some FMH-3s permit headers to follow. (See "FMH-3 Concatenation and Chaining Rules" on page 195.)

FMH-3 functions are shown in Figure 4-4 and defined under "Function Management Header Type-3 (FMH-3)" on page 227.

Type of Operation	FMH-3 Function
Compression and compaction	Query for a compaction table
	Send a compaction table
	Send a prime compression character
FMH correlation	Send a series ID
	Send status

Figure 4-4. FMH-3 functions. FMH-3 is used to process information for all destinations in the LU-LU session.

RULES FOR FMH-1, FMH-2, AND FMH-3

Hierarchy Rules

Hierarchy rules describe how and when headers are used. The following rules specify and define FMH-1, FMH-2, and FMH-3 interactions:

1. An FMH-1 must be used to select a destination except for the default destination (use the BEGIN, BEGIN/END, or RESUME form of FMH-1). A destination, once selected, becomes the active destination.
2. An FMH-1 must be used to indicate completion of the selected destination (use SUSPEND or END).
3. All FMH-2s apply to the active destination.
4. An FMH-2 may occur only when a destination is active and may occur:

   Immediately following an FMH-1 or another FMH-2

   As the first FM header in an RU chain
5. An extension FMH-2 applies to the last root FMH-2 sent for the active destination.
6. An FMH-3 applies to all destinations in the LU-LU session.
7. An FMH-3 may be sent at any time; a destination need not be active.

FMH-1 Destination Selection Rules

Rules for selecting destinations are listed below and summarized in Figures 4-5 and 4-6.

1. A BEGIN, BEGIN/END, or RESUME is valid only when the first element of a stack is empty. An error response must be returned if the first element is not empty. (See Figure 4-5.)
2. An END, END-ABORT, CONTINUE, or SUSPEND is valid only when an entry is in the first element of a stack. That is, a destination must be active to suspend or end (normal or abort) it. An error response may be returned if the first element is empty. (See Figure 4-5.)
3. BEGIN, SUSPEND, RESUME, CONTINUE, and END must be sent with the stack reference indicator (SRI) set to zero; only END-ABORT and, for reply-type structured fields only, BEGIN/END may be sent with the SRI set to one.

4. The receiver of a BEGIN retains all the information contained in the FMH-1 except FMHC and SRI information. The receiver also is required to retain related, static FMH-2 information. The sender of BEGIN is not required to resend this information on a subsequent SUSPEND, RESUME, END, CONTINUE, or END-ABORT.

5. Since a stack entry contains all the destination selection parameters contained in the FMH-1 as well as related, static information contained in subsequent FMH-2s (such as VOLUME ID), the receiver of a SUSPEND, RESUME, END, or END-ABORT performs the appropriate destination selection operation. No other parameters in these FMH-1s are examined.

6. After a destination has been selected using BEGIN, the destination remains active, and all FM data is sent to the active destination until another FMH-1 is sent to change the active destination.

7. The use of SUSPEND requires the use of RESUME to continue directing data to the suspended destination.

Input	Stack Condition	Output
BEGIN	P1 = Empty	Write to stack
	P1 ¬= Empty	Error (mandatory -RSP X'10082001')
END or END-ABORT	P1 = Empty	Error (-RSP X'10082002')
	P1 ¬= Empty	Erase active destination entry
SUSPEND	P1 = Empty	Error (-RSP X'10082002')
	P1 ¬= Empty	Push down stack
RESUME	P1 = Empty	Pop up stack
	P1 ¬= Empty	Error (mandatory -RSP X'10082001')
CONTINUE	P1 = Empty	Error (-RSP X'10082002')
	P1 ¬= Empty	Use active destination with any altered CMI, CPI, or ERCL
BEGIN/END	P1 = Empty	Write to stack and erase active destination at completion of operation
	P1 ¬= Empty	Error (mandatory -RSP X'10082001')

Figure 4-5. Summary of destination selection operations. P1 is the top entry in the stack.

SEQUENCE 1:

```
   LU X                                        LU Y
Send stack                                  Receive stack

    A              ---BEGIN A---->              A
Empty, A           --SUSPEND A--->          Empty, A
                   -No FMH, data->          (Data without an FMH-1
                                            is sent to default
                                            destination)
```

SEQUENCE 2:

```
   LU X                                        LU Y
Send stack                                  Receive stack

    A              ---BEGIN A---->              A
Empty, A           --SUSPEND A--->          Empty, A      (Note 1)
   B,A             ---BEGIN B---->              B,A
Empty, A           ----END B----->          Empty, A
    A              --RESUME A---->              A         (Note 2)
                   -No FMH, data->          (Data without an FMH-1
                                            is sent to destination A)
```

Figure 4-6 (Part 1 of 2). Sample stack sequences.

SEQUENCE 3:

LU X Send stack	LU X Rcv stack		LU Y Rcv stack	LU Y Send stack
A	Reset	---BEGIN A--->	A	Reset
		----data-----> (Note 3)		
A	B	<---BEGIN B---	A	B
		<---data------ (Note 4)		
A	B	data	A	B
		-------------> (Note 5)		

SEQUENCE 4:

LU X Send stack	LU X Rcv stack		LU Y Rcv stack	LU Y Send stack
A	Reset	---BEGIN A--->	A	Reset
A	Reset	<--- +RSP----- (Note 6)	A	Reset
A	Reset	----data----->	A	Reset
A	Reset	----data----->	A	Reset
A	Reset	<--- -RSP----- (Note 7)	A	Reset
Empty, A	Reset	---SUSPEND---> ----data-----> (Notes 8,9)	Empty, A	Reset

Notes:

1. The use of SUSPEND requires the use of RESUME to continue directing data to destination A.
2. If RESUME A were not sent, the data following END B would be directed to the default destination.
3. Data directed to destination A. (A stack is reset when all stack elements are empty.)
4. Data directed to destination B.
5. Data directed to destination A.
6. A positive response is returned to a BEGIN FMH-1 sent OIC.
7. Negative response does not put the default destination in LU X send stack or LU Y receive stack.
8. Data directed to default destination.
9. If SUSPEND were omitted from this sequence, the data following the negative response would be directed to destination A.

Figure 4-6 (Part 2 of 2). Sample stack sequences.

FMH-1 Concatenation and Chaining Rules

1. An FMH-1 cannot be concatenated to another FMH-1.
2. An FMH-2 cannot be concatenated to an FMH-1 indicating SUSPEND, CONTINUE, END, or END-ABORT.
3. An FMH-2 can be concatenated to an FMH-1 indicating BEGIN, BEGIN/END, or RESUME.

4. An FMH-3 cannot be concatenated to an FMH-1.
5. An FMH-1 must be contained within a chain.

FMH-1 Data Rules

1. An FMH-1 indicating SUSPEND, CONTINUE, or END-ABORT must be sent without FM data.
2. An FMH-1 indicating BEGIN, BEGIN/END, RESUME, or END may be sent with or without data.
3. An EB resets the send and receive stacks for both directions.
4. CLEAR and SDT reset the send and receive stacks in both directions.

FMH-2 Concatenation and Chaining Rules

1. A root FMH-2 may be sent alone, or it may be concatenated to an FMH-1.
2. A root FMH-2 cannot be concatenated to an FMH-1 indicating SUSPEND, END, or END-ABORT.
3. Only one root FMH-2 is allowed per chain.
4. Extension FMH-2s follow these rules:
   a. A VOLUME-ID FMH-2, when sent, must be concatenated to an FMH-1.
   b. A RECORD-ID FMH-2 may be sent alone in a chain, or it may be concatenated to a root FMH-2.
   c. A PASSWORD FMH-2, when sent, must be concatenated to an FMH-1, a root FMH-2, or an extension FMH-2.
   d. More than one extension FMH-2 is permitted per chain, but there can be only one of each kind per chain.
5. An FMH-2 must be contained within a chain.

FMH-2 Data Rules

1. FM data may follow an FMH-2 or be sent as the next chain.
2. If no root FMH-2 is concatenated to an FMH-1 with FM data, sequential addition of records (the ADD FMH-2) is the default.

FMH-3 Concatenation and Chaining Rules

1. An FMH-3 must be sent first in a chain.
2. The SERIES-ID FMH-3 and the STATUS FMH-3 permit an FMH-1 or FMH-2 to be concatenated to them.
3. An FMH-3 must be contained within a chain.

SENSE CODES FOR FMH-1, FMH-2, AND FMH-3 ERRORS

The logical errors concerned with the hierarchy, format, and protocol of FM headers are reflected in user sense codes appended to the system sense code X'1008' (invalid FM header). FM header session errors and data processing errors also follow this approach. If this level of error detection is not provided, then the sense code X'10080000' is returned.

In addition to FM header errors (sense codes X'10080000' or X'1008xxxx'), other request error as well as request reject errors, transmission control errors, and path control errors may be sent or received to an RU chain containing only FM header(s) or FM header(s) and end user data.

Session Errors (100840xx)

Invalid FMH type (4001)
Invalid FMH code (4002)
Compression not supported (4003)
Compaction not supported (4004)
Basic exchange not supported (4005)
Only basic exchange supported (4006)
Medium not supported (4007)
Code selection compression violation (4008)
FMHC not supported (4009)
Demand select not supported (400A)
DSNAME not supported (400B)
Invalid medium subaddress field (400C)
Insufficient resources to perform FMH function (400D)
DSP select not supported (400E)

FM Header Protocol Errors (100820xx)

Invalid destination -- active (2001)
Invalid destination -- inactive (2002)
Invalid destination -- suspended (2003)
Invalid suspend-resume sequence (2004)
Interruption level violation (2005)
Invalid resume properties (2006)
Destination (MEDIUM.SUBADDRESS.DSNAME) not available (2007)
Invalid end sequence (2008)
Invalid FM header length (2009)
Invalid field setting -- reserved field set to one or setting not defined (200A)
Invalid destination -- destination does not exist (200B)
Invalid ERCL (200C)
Invalid DST (200D)
Invalid concatenation -- header can not be concatenated (200E)
FM data not allowed for header (200F)
Bind FM header set violation (2010)
FM header not sent concatenated (2014)
Stack reference indicator invalidly set to one for BEGIN, SUSPEND, RESUME, or END FMH-1 or for FMH-2 (2019).
Unable to accept CMI modification (201A)
Unable to accept CPI modification (201B)
Unable to accept ERCL modification (201C)

Data Processing Errors (100808xx)

Invalid function code parameters (0801)
Forms function cannot be performed (0803)
Unable to perform copy function (0805)
Compaction table outside supported subset (0806)
Invalid PDIR identifier (0807)
Printer train function cannot be performed (0808)
FCB Load function cannot be performed (0809)
FCB load function not supported (080A)
Invalid compaction table name (080B)
Invalid ACCESS (080C)
Invalid RECLEN (080D)

Invalid NUMRECS (080E)
Data set in use (080F)
Data set not found (0810)
Invalid password (0811)
Function not allowed for destination (0812)
Record too long (0813)
Data set full (0814)
Invalid RECID (0815)
Invalid VOLID format (0817)
Number of logical records per chain exceeded (0818)
Data set exists (0819)
No space available (081A)
Invalid VOLID (081B)
Invalid DSACCESS (081C)
Invalid RECTYPE (081D)
Insufficient resolution space (081E)
Invalid key technique (081F)
Invalid key displacement (0820)
Invalid key (0821)
Invalid N (number of records) (0822)
Invalid KEYIND (0823)
Invalid SERID (0824)
Invalid RECID format (0826)
Password not supplied (0827)
Record ID not supplied (0828)
Volume ID not supplied (0829)
Invalid PGMNAME (082A)

Note: The capitalized words above are parameters within an FMH-1, FMH-2, or FMH-3. (See the header formats later in this chapter for parameter definitions.)

FM HEADERS FOR LU-LU SESSION TYPE 6

USING FMH-4 FOR DESCRIBING DATA

FMH-4 may be used by one session partner to describe the data it is sending to the other session partner. The data is handled on each end of the session by transaction programs. The transaction program that uses FMH-4s is called logical message services (LMS).

The message manager within LMS is responsible for adding messages (received from the session partner's LMS) to the appropriate queue. It is also responsible for dequeuing messages requested by other programs at its end of the session and passing the messages to them for processing.

Each message LMS receives from the session partner is preceded by an FMH-4.

Transmission Types

The transmission type is specified by a 2-byte field in a header. The first byte indicates the data stream type, and the second byte provides any qualifying information. The data stream types are:

Type code used	Denotes
X'0000'-X'3F00'	Reserved
X'4ncc'	Field formatted records. First byte indicates type of records (see explanations below), and second byte identifies separator used between records.
X'5000'-'FE00'	Reserved
X'FF00'	Type is specified only at a lower level.

Field Formatted Records (FFRs)

The data portion of a field formatted record (FFR) comprises a sequence of transmission fields. The data content is not restricted in terms of contained byte-values. The essential property of an FFR is that it can be decomposed into separate fields by the receiving message manager based on a combination of information contained in the (explicit or implicit) headers and in the record data content.

The following description assumes that the message manager obtains the following information when it does a read for an interchange unit:

1. FMH = Yes/No
2. DBA = ATTDBA (deblocking algorithm)
3. Count of byte string
4. Byte string

The FFR form seen by the message manager is:

Byte	Contents
0 to n	A logical record header. The header length is not explicit in the data stream.
n+1 to m	The data portion, called the logical record.

Several FFR formats are defined. The purpose of these formats is to provide data stream conventions for indicating information about transmission-field to stored-logical-record-field relationships. The information concerned covers:

1. Whether an omitted transmission field is to replace a stored logical record field with a zero-length value or is to leave the stored logical record field unchanged

2. To indicate the number of bytes in the transmission field value, and to control the length of the corresponding stored logical record field

3. To indicate which stored logical record field a transmission field applies to

Three formats are defined:

FFR-FNI	X'40'	Fixed fields without field separators
FFR-FS	X'41'	Fixed fields with field separators
FFR-FS2	X'42'	Fixed fields with or without field separators

FFR-FNI format. The FFR-FNI format relies on field ordering and a transmission field for every stored logical record field to indicate c) above; all remaining information must be communicated via embedded data conventions understood by the maps involved.

The logical record comprises a sequence of transmission fields which are matched, in sequence, with fields in the corresponding stored logical record. A transmission field is a sequence of bytes with no leading or trailing separator. Any conventions concerning selector fields are receiver-map understood.

FFR-FS format. The FFR-FS format also relies on field ordering and a transmission field for every field to indicate c) above. Further, a predefined separator character indicates unchanged fields of a) above, and indicates the number of bytes in the transmission field.

The logical record comprises a sequence of fields which are matched, in sequence, with fields in the corresponding stored logical record.

A transmission field is terminated by the first occurrence of a transmission-defined separator character. The separator character is defined following the X'41' FFR-FS code in the transmission type portion of the FMH-4.

If there are no data bytes preceding the separator of a transmission field, the transmission field has no effect on the corresponding stored logical record field. Otherwise, the bytes preceding the separator are applied to the stored logical record field using map-specified rules.

FFR-FS2 format. The FFR-FS2 format is similar to the FFR-FS format, but is characterized by the use of a predefined separator character or the use of a predefined receiver map to determine field length.

Fields are terminated by either the separator character defined in the byte after the X'42' FFR-FS2 code of the FMH-4, or the maximum length of the field as defined in the receiver map.

Command Execution Rules for FMH-4

Command rules are:

1. Make a stored logical block of the specified attributes the new last element in the current logical message.
2. If the block is elementary, make the data portion of the current transmission block the value of the new block.
3. If the block is compound, construct the block value by constructing its component stored logical records.

Message Service Errors

The following sense codes are defined. They constitute negative responses for sender ERP and may be associated with message text flowing with FMH-7 for selective receiver ERP.

Sense Code	Error
1008 C000	Header not supported
1008 C001	Invalid header length
1008 C002	Logical message services block-level error
1008 C003	Version ID mismatch

USING FMH-5 FOR ATTACHING TRANSACTION PROGRAMS

FMH-5 FUNCTIONS: Function management header type 5 (FMH-5) has three functions that can be used by one half-session to control the attachment of transaction programs at the session partner on an LU_T6 session:

- The ATTACH FMH-5 is used to tell the session partner to attach a specific transaction program, called the destination program. The ATTACH FMH-5 contains the following information:

  Data stream profile (DSP)
  Deblocking algorithm (DBA)
  Destination program name (DPN)
  Return destination program name (RDPN)
  Primary resource name (PRN)
  Return primary resource name (RPRN)
  Destination process queue name (DQN)
  Access code

- The DATA DESCRIPTOR FMH-5 is used to tell the session partner of a change in the characteristics of the data stream, such as a new data stream profile or deblocking algorithm.
- The RAP FMH-5 (reset attached program FMH-5) is used to tell the session partner to detach the specified transaction program.

DATA STREAM PROFILE (DSP): The DSP field defines the type of data stream that is to be sent. The field also permits an implementation-defined modifier.

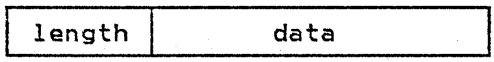
DEBLOCKING ALGORITHM (DBA): The ATTDBA field of the FMH-5 identifies the algorithm used to create the quantity of data seen by a transaction program in a single read operation from its half-session. The defined deblocking algorithms are:

A chain of RUs
Variable-length records

An RU is composed of the FM data following the RH.

A chain of RUs can be composed of any generalized FM data, as delimited by end of chain (EC).

Variable-length records can be composed of any binary data, delimited by a self-defining length. Records are of the form:



The length field is 2 bytes. The length specified includes the length field.

For output purposes, similar algorithms can be selected for the placement of application data into RU(s) and chain(s), for example:

Application data equals chain
Variable-length records

A default DBA selection may be specified for both input and output. The identification of the default deblocking algorithms is receiver defined. DBA is explicitly or implicitly (default) required for each message within an LU_T6 session.

DESTINATION PROGRAM NAMES (DPN): Destination program names provide a mechanism for selecting transaction programs within a transaction processing system (TPS). In the absence of an explicit program name, which is identified in the ATTDPN field of the FMH-5, a default program can be selected.

As programs run synchronously within the half-session, errors associated with the program can be reported if necessary via responses. A program in a queued system might be either the map procedure, that is, an interpretive program as in IMS, or perhaps a TPROCESS Queue in TCAM. For programs that execute within the half-session, a synchronous transaction code is appropriate for the program name. Reasonable defaults might be, respectively:

The normal edit procedure and context router
A customer-defined TCAM application
The existing transaction decode algorithm

NAMING TRANSACTION PROGRAMS: Transaction program names can be comprised of any combination of graphic characters. Implementation- and user-defined program names must have a first character that is a graphic (X'40' to X'FE'). All IBM service program names have a first character that is a nongraphic (less than or equal to X'3F'). The assigned leading character for these service programs is:

IBM service program	Value
System message program	X'01'
Scheduler program	X'02'
Queue program	X'03'
DL/1 program	X'05'

RETURN DESTINATION PROGRAM NAME (RDPN): The sending half-session may identify one program to receive a reply from the destination program in the session partner. The reply (or replies) occurs when the destination program completes the original request(s) of the sending program. The name of the program that is to receive the reply is sent to the destination program in the ATTRDPN field of the FMH-5.

The RDPN may be used to support a pseudo-conversation between two TPSs over a period of time. One TPS sends a message to the other, providing the RDPN in the FHM-5 that will handle the reply. The second TPS processes the request, possibly generating collateral processing, and, by using the RDPN, sends its reply to the originating program.

The receiving (destination) TPS need not use the RDPN.

If an RDPN is not received, a TPS may send a reply to the originating program, or it may select another program in the originating TPS and send the reply to that program.

If a return program name is not received, no default is provided by the receiver, and the ultimate application is responsible for the creation of a program name to be used with the reply if one is to be generated.

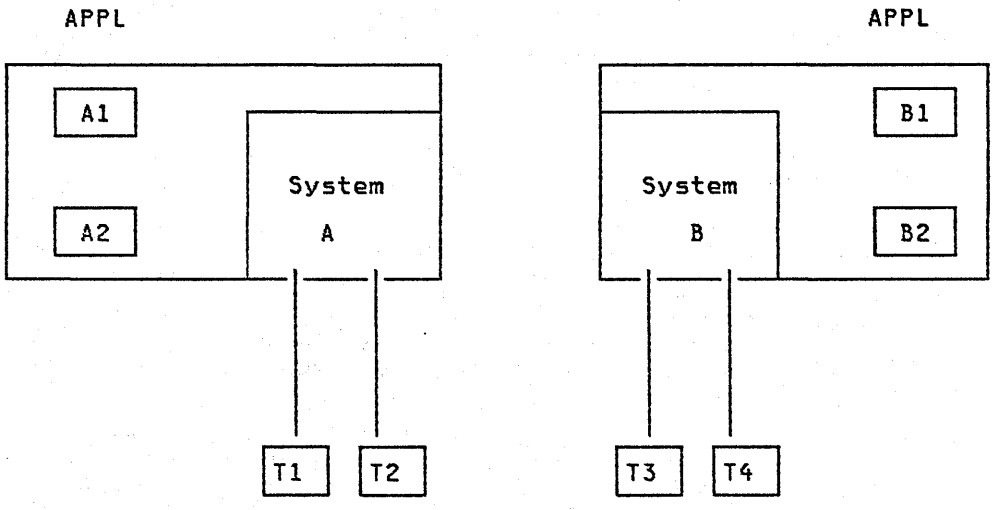
PRIMARY RESOURCE NAME (PRN): In order to select and initiate a program in a TPS, it may be necessary to acquire a number of system resources required by that program for successful initiation. The required resources are normally specified by the requesting system, and must be validated by the receiving system before accepting (responding) to the request. Since these resources are to be acquired by the receiving system, they must be carried with the program initiation request.

Typically within a TPS, a named program is associated with another resource or group of resources, such as the attaching terminal or a database. With LU_T6, it is possible to extend the resources associated with a named program to include a resource in the other half-session as well as a terminal. It is possible to send a request for program initiation on an LU_T6 session and to specify that the program is to be attached only on the acquisition of named resources at the receiver. The named resources are specified in the ATTPRN field in the ATTACH FMH-5.

The ATTPRN field may not be present if the program name does not require resources in order to be successfully attached.

RETURN PRIMARY RESOURCE NAME (RPRN): Because a primary resource can be nominated when initiating a program, a return primary resource name can be associated with the return program. As with the return program name, this RPRN would be routed back with the reply (if generated) to the first system. Application override is possible as with return program name.

The combination of the above four fields (ATTDPN, ATTPRN, ATTRDPN, and ATTRPRN) allows message routing facilities between the two sets of primary resource names as illustrated below:



A message originating at terminal T1 on system A is directed to application B1 on system B for eventual return to terminal T1 via program A1.

```
        System A                          System B

(1)     ATTACH                      (2)   ATTACH
        DPN=B1                            DPN=A1
        PRN=                              PRN=T1
        RDPN=A1                           RDPN=
        RPRN=T1                           RPRN=
```

A message originating at terminal T4 in system B is directed to application A2 in system A for return to terminal T4 via application B2.

```
        System A                          System B

(2)     ATTACH                      (1)   ATTACH
        DPN=B2                            DPN=A2
        PRN=T4                            PRN=
        RDPN=                             RDPN=B2
        RPRN=                             RPRN=T4
```

A conversation can be achieved by always supplying a RDPN in both ATTACH FMH-5s.

SCOPE OF PRN AND RPRN: The PRN is unique within the PRN name space of the receiving LU. The PRN and RPRN are composed of characters from the graphic set (greater than X'40').

DPN QUEUE NAME (DQN): The ATTDQN field of the FMH-5 identifies a queue to be associated with a destination program name. This queue allows a sending system to build up a queue of data and to nominate a program in a receiving system to process it.

INTERCHANGE UNIT TYPE (IUT): The source half-session specifies the limits of an interchange unit of data. The interchange unit may be contained within a single RU chain, or it may span multiple RU chains. The source half-session specifies the way the information unit is packaged in the FMH5IUT field of the FMH-5.

INTERCHANGE UNIT END (IUE): The source half-session identifies the end of an interchange unit by setting the FMH5IUE field of the FMH-5 to B'1'. The FMH5IUE field must be used to terminate an information unit when:

The IU spans multiple RU chains

The IU doesn't span RU chains, but a previous IU is still active that did.

FMH5IUE = B'1' indicates that the IU ended with the last character of data sent prior to the FMH-5. With the exception that the FMH-5 must be at the beginning of the first RU in the RU chain, the source half-session can specify termination of the interchange unit (in the multiple-chain case) independent of any RH indicators.

USING FMH-6 FOR TRANSACTION PROGRAM PROCESSING

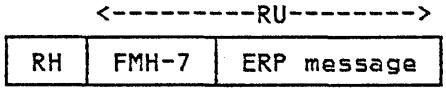
FMH-6 may be used by LU_T6 half-sessions to carry a transaction program's commands, messages, and data to an active transaction program at the session partner. The FMH-6 format is defined by LU_T6, but the commands to be used and the format of the data are defined by the transaction programs that are attached to the LU_T6 half-sessions. Consequently, only the general format of FMH-6 is discussed in this book (see "Function Management Header Type 6" on page 235).

An FMH-6 typically flows after an FMH-5 because it must be sent to an active (attached) transaction program. It may be the first FMH on the session, however, if it is for the default destination. Another FMH may follow the FMH-6; transaction programs can also define FMH-6s where other FMHs cannot be concatenated.

FMH-6 field definitions are described in the publications of products that implement LU-LU session type 6. For additional details, see those publications.

USING FMH-7 FOR ERROR RECOVERY

FMH-7 is sent from one session partner to the other to define the cause of an error and to provide corrective information. The corrective information is called an ERP message, which can be any information understood by the session partners. An ERP message may contain any of the FM level LU_T6 sense codes. The format of the RU that contains the FMH-7 and ERP message is:



The session partners should use the SSCP character set to ensure compatibility of messages between transaction programs. This character set is described in Figure 4-7.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0					SP	&	-									0
1							/						A	J		1
2													B	K	S	2
3													C	L	T	3
4													D	M	U	4
5													E	N	V	5
6													F	O	W	6
7													G	P	X	7
8													H	Q	Y	8
9													I	R	Z	9
A																
B					.	$	,	#								
C						*		@								
D					(	)		'								
E					+			=								
F																

Figure 4-7. Character set used for ERP message text and for LU to SSCP communication.

Typically, an ERP message is sent by the data receiver (the half-session in receive state). When sent by the data receiver, it must be preceded by negative response X'0846'. The RU chain carrying the FMH-7 and ERP message must be sent as one of the following:

```
¬BB,¬EB, CD
¬BB, EB,¬CD
 BB, EB,¬CD
```

An ERP message may be sent by a half-session that is in send state when the half-session detects an SUW abort condition. In this case, the ERP message is not preceded by negative response X'0846', and it may only contain one of the abort unit of work sense codes. The FMH-7 associated with it does not carry a correlation sequence number; in this case the chain carrying the FMH-7 may optionally flow with CD.

Since SUW abort in one half-session leads to a sympathetic abort in the other half-session, half-sessions can exchange ERP messages and thus full information concerning the respective backout actions. ERP message exchange can occur within the failing bracket when the first ERP message carries CD and not EB. The reply may be another ERP message, LUSTAT X'0006', or restart data.

An ERP message is part of a single RU chain; the chain is the first and only FMD chain following negative response X'0846'. When the negative response is sent with QRI=0, it may bypass any chain previously sent. Consequently, the chain containing the ERP message must contain additional correlation information. This information is provided by FMH-7.

USING FMH-10 TO SYNCHRONIZE UNITS OF WORK

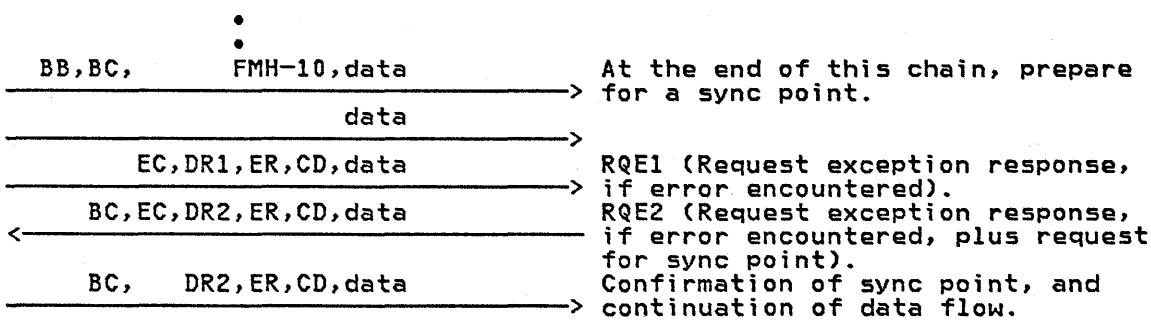
FMH-10 is used to tell the session partner to prepare its resources so that work being done by the transaction programs can be synchronized. If the session partner can guarantee that the resources can be committed (updated) when the sync point is received, it requests a sync point (called request commit). If it can't commit the resources, it returns an FMH-7.

FMH-10 is used primarily by implementations that have resources that must be synchronized across more than one LU_T6 session. The transaction program is responsible for handling the synchronizing of sessions.

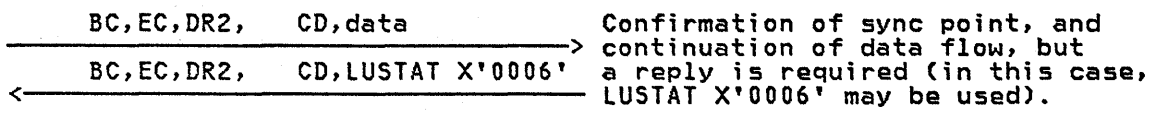
This section describes protocols that are related to FMH-10 on a single session. The sender of FMH-10 must send the FMH-10 as the first FM header of the RU chain, and the last RU in the chain must have the CDI set on. The FMH-10 can be sent requesting a definite or exception response. (With definite response protocol, a positive response only indicates that the RU chain containing the FMH-10 was received properly.)

The following examples show a successful and an unsuccessful sync point. The examples assume RU chains were sent requesting exception responses.

EXAMPLE 1: Successful sync point.



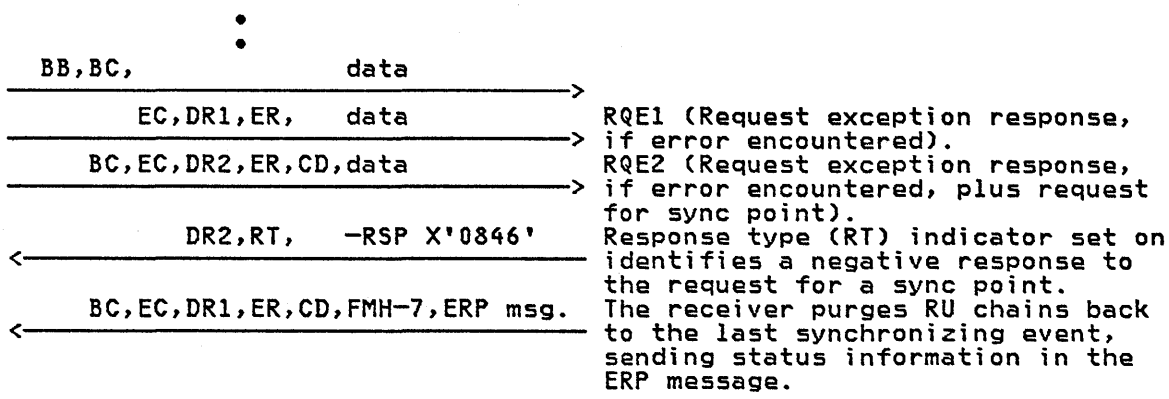
An alternative to the last flow is:



Regardless of whether an FMH-10 is used, the DR2 indicator in the RH requests a sync point. Valid replies for successful sync points are:

- Positive response with DR2 indicator set on if definite response is required (not shown above)
- LUSTAT X'0006' with DR2 indicator set on if definite response is required
- Any reply (request or response), including one with data, that has the DR2 indicator set on.

EXAMPLE 2: Unsuccessful attempt to achieve sync point.



FIELD AND BIT DEFINITIONS FOR FM HEADERS

This section defines field and bit settings for function management headers (FMHs). FM headers can be used on LU-LU session types 0, 1, 4, and 6. (LU-LU session types 2 and 3 do not use FM headers.)

The headers are organized by type (FMH-1, FMH-2, FMH-3, etc.) and then alphabetically within type.

The following headers are defined:

FMH-1 - This header is used to select a destination within an 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 that destination using the FMH-1.

FMH-2 - Once a destination has been selected using an FMH-1, this header is used to handle the data management tasks for that destination.

FMH-3 - This header is used to handle data management tasks that are common to all destinations in the LU-LU session.

FMH-4 - This header is used to carry a logical block command and associated information. A logical block is contained within a logical message as defined for logical message services. Logical message services uses LU-LU session type 6 to transmit messages between end users.

FMH-5 - This header is used to select a named transaction program within the receiving LU. FMH-5 flows from the LU services manager for the sending half-session to the LU services manager for the receiving half-session. It can be followed by other FMHs (for example, FMH-6 and FMH-4), a logical record header (LRH), and FM data. It can optionally be sent with CD or EB.

FMH-6 - This header is used to carry a command from a currently active transaction program attached to the sending half-session to a currently active transaction program attached to the receiving half-session.

FMH-7 - This header is used to send further information about an error after a negative response (0846) has been received.

FMH-10 - This header is used to prepare the session for a sync point. It may be sent with data. The RU chain must have CD set on so that the receiver may, on the next flow, may request the sync point or abort the unit of work. FMH-10 flows from the LU services manager for the sending half-session to the LU services manager for the receiving half-session. Any data accompanying the FMH-10 is passed to the transaction program before the sync point.

FUNCTION MANAGEMENT HEADER TYPE 1 (FMH-1)

FMH-1 is used by one half-session to select a destination located at the session partner. If the originating half-session wishes to change the destination, it can suspend the active destination, select a new destination, end the new destination, and resume transmitting to the original destination by using FMH-1s with different destination select (DSSEL) parameters.

Format:

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC	FMH concatenation. B'0' No FMH follows this FMH-1 B'1' Another FMH follows this FMH-1
	1-7	B'0000001'	FMH-1 Identifier
2	0-3	Medium Select	Desired medium for data:
		X'0'	Console
		X'1'	Exchange
		X'2'	Card
		X'3'	Document
		X'4'	Nonexchange disk
		X'5'	Extended document
		X'6'	Extended card
		X'7'	Data set name selects destination (see Note 3)
		X'8'	WP media 1
		X'9'	WP media 2
		X'A'	WP media 3
		X'B'	Reserved
		X'C'	WP media 4
			All other values reserved.
	4-7		Logical subaddress
		X'0'...X'E'	Specific device in medium class
		X'F'	Any device in medium class

Note 1: The 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 must do so by specifying the desired DSP in byte 3, bits 4-7 of the FMH-1. This selection must adhere to those DSPs allowed on the session as specified in the BIND parameters.

Note 2: Media and logical subaddress values are reserved when DSSEL field is set to B'110' (continue destination selection), B'001' (end), B'100' (suspend), or B'101' (end abort).

Note 3: If medium = X'7' and logical subaddress = X'F', DSNAME field is used to select destination.

Byte	Bit	Content	Meaning
3	0	SRI	Stack reference indicator
		B'0'	Stack to be used is the sender's send stack.
		B'1'	Stack to be used is the receiver's send stack.
	1		Demand select
		B'0'	Receiver may direct data to alternate medium/subaddress.
		B'1'	Receiver must direct data to specified medium/subaddress (spooling is prohibited).
	2-3		Reserved
	4-7	DSP Select	Data stream profile (DSP)
			Profiles are:
		X'0'	Default DSP
		X'1'	Base DSP
		X'2'	General DSP
		X'3'	Job DSP
		X'4'	WP raw-form text DSP
		X'5'	WP exchange diskette DSP
		X'A'	Document interchange DSP
		X'B'	Structured field DSP
			All other values reserved.

Notes: If the DSP select field is X'0', the DSP is implied by the medium select field.

4			FMH-1 properties
	0-2	DSSEL	Destination selection
		B'000'	Resume destination selection
		B'001'	End destination selection
		B'010'	Begin destination selection
		B'011'	Begin/end destination selection
		B'100'	Suspend destination selection
		B'101'	End-abort destination selection
		B'110'	Continue destination selection
		B'111'	Reserved
	3	DST	Data set transmission
		B'0'	Transmission exchange format
		B'1'	Basic exchange format
			When medium select ¬= exchange medium, this field is reserved. Receiver may do spooling and exchange-medium creation locally.
			When medium select = exchange medium (see byte 2), specifying B'0' preserves chain boundaries while spooling, but nonsequential allocation techniques may be used. Specifying B'1' does not preserve chain boundaries, but uses sequential medium allocation. (See "Field Definitions" below.)
	4		Reserved
	5	CMI	Compression indicator (See "Notes" below.)
		B'0'	No compression
		B'1'	Compression
	6	CPI	Compaction indicator (See "Notes" below.)
		B'0'	No compaction
		B'1'	Compaction
	7		Reserved

Byte	Bit	Content	Meaning
5	0-7	ERCL	Exchange record length if medium = exchange medium or card; otherwise reserved. For medium = card, a hexadecimal value indicates maximum card length. The value X'00' indicates an 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. CMI, CPI, and ERCL information received when DSSEL = continue overlays the settings of the BEGIN FMH-1 or the last-received CONTINUE FMH-1.
2. CMI, CPI, and ERCL indicators are meaningful and valid only when specified in a BEGIN, BEGIN/END, or CONTINUE FMH-1.
3. If CMI or CPI is on, the first byte following the FMH (or FMHs) is a string control byte (SCB).

FMH-1 Field Definitions

Length: Specifies the variable number of bytes within the header, including the length byte.

FM Header Concatenated (FMHC): Indicates the presence of a further FM header starting in the first byte following this FM header. Concatenated headers, which carry additional information concerning the destination identified by this type-1 header, are defined as type-2 headers. All concatenated FM headers must have the same setting of the Stack Reference Indicator.

Type: Identifies the functional capability supported by the FM Header. FM header types 1 through 8 and 10 are defined in this book; types 40 through 4F are restricted for use by SNA implementations for private protocols; of those, X'40' through X'43' have been assigned. Types 50 to 7F are assigned for customer use. All other types are reserved.

Medium Select: Identifies a device class to which the sender or receiver directs data. Each type of LU-LU session supports these media differently. Before selecting a medium, you should ensure that your LU-LU session provides the support you require.

WP media 1 through 4 are word processing (WP) media and are defined by the implementation.

Subaddress: Identifies a specific destination (device) to which the sender or receiver directs data. Up to 15 destinations may be explicitly selected. The capability to select any destination in the class (Subaddress = X'F') is also provided. In this case, assignment and selection of the specific destination is done by the receiver of the FMH-1.

The combination of length, medium select, and subaddress fields indicates the destination selection mechanism.

a. Length < 6 is invalid.

b. Length = 6: Medium select and subaddress are used to select the destination.

c. Length > 6 has two possibilities:

   When the medium select and subaddress fields ¬= X'7F', the DSNAME field is used as a subselection field within the selected medium and subaddress (for example, data set on a disk) to identify the destination.

When the medium select and subaddress fields = X'7F', the DSNAME field is the data stream selected as the destination, and the medium and subaddress fields are ignored.

Stack Reference Indicator (SRI): Specifies to which destination this FMH-1 applies. If an LU wants to refer to the destination it began or wants to begin, the bit is 0; if an LU wants to refer to the destination begun by the other LU, the bit is 1.

Only two FMH-1s can be sent with SRI = 1:

END-ABORT (DSSEL = B'101').

BEGIN/END (DSSEL = B'011'), but only for reply-type structured fields (DSP Select = X'B').

Demand Select: Indicates whether the receiver must direct data to the destination specified in the medium and subaddress fields.

Data Stream Profile (DSP) Select: Indicates the data stream profile to be used for the selected destination. A value other than X'0' overrides the default DSP of the medium selected in the medium select field.

Destination Selection (DSSEL): Indicates the beginning, suspension, resumption, continuation, and ending of a destination. The functions performed by these bits are consistent across all destinations.

RESUME (B'000') reactivates a previously suspended destination. The destination that is the second entry in the stack is made the active destination by moving the entry to the top of the stack. All FMH-1 fields except Length, FMHC, and DSSEL are ignored. The top entry in the stack must be empty prior to issuing a RESUME FMH-1.

END (B'001') normally terminates the active destination at the end of the present chain. At the completion of the operation, the top entry in the stack is empty.

BEGIN (B'010') identifies a new destination. The destination becomes the top entry in the stack -- the active destination. The destination remains active until an END, END-ABORT, or SUSPEND FMH-1 is issued.

BEGIN/END (B'011') identifies a new destination for the duration of the chain. The destination remains the active destination until the end of the chain is detected; then it is removed from the top entry in the stack. At the end of the operation, the top entry is empty.

SUSPEND (B'100') moves the top entry (the active destination) in the stack to the second entry, thus suspending the active destination. The RU should contain only the SUSPEND FMH-1; all other RU contents are ignored. At the completion of the operation, the top entry is empty.

END-ABORT (B'101') immediately terminates the active destination. The END-ABORT FMH-1 must be the only contents of the RU. At the end of the operation, the top entry of the stack is empty. (The stack reference indicator (SRI) defines which stack is affected.)

CONTINUE (B'110') enables you to change the compression, compaction, and exchange-record-length characteristics of the active destination. The CONTINUE FMH-1 alters the CMI, CPI, and ERCL fields previously defined by the BEGIN FMH-1 that established the active destination.

Data Set Transmission (DST): Signals that the RU format (beyond this FMH-1 or concatenated FMH-2s) is either user defined (transmission format) or the basic-exchange format. If DST specifies basic-exchange format, then Medium Select should specify exchange format (X'1').

Compression Indicator (CMI): Indicates, if CMI = 1, that the RU contains compressed data. When CMI = 1, a string control byte (SCB) must follow either this FMH-1 or the concatenated FMH-2s.

Compaction Indicator (CPI): Indicates, if CPI = 1, that the RU contains compacted data. When CPI = 1, a string control byte (SCB) must follow either this FMH-1 or the concatenated FMH-2s.

Exchange Record Length (ERCL): Specifies the length of the records that follow the FM headers and SCBs (if present). ERCL is specified in hexadecimal and is provided to assist the receiver in deblocking the subsequent data stream.

When the medium select field specifies Card (X'2'), ERCL indicates the maximum presentation position (MPP) for the card medium. The ERCL value X'00' indicates an 80-column length.

When the medium select field specifies exchange (X'1') and the destination selection is for existing data (for example, a previously created diskette data set), ERCL must equal the record length of the existing data. If the selection is for new data, ERCL indicates the record length of the data set to be created. An ERCL value of X'00' instructs the receiver to use the value associated with existing data; if there is no existing data, the receiver may choose a default.

ERCL values for the basic-exchange format cannot exceed 128 bytes. The data is stored on diskette as fixed records, padded with nulls to 128 bytes, if necessary.

Destination-Name Length (DSLEN): Specifies the length in bytes of the DSNAME field. This field may be omitted when DSNAME is omitted.

Destination Name (DSNAME). Specifies a variable-length destination name. This field may be omitted.

FUNCTION MANAGEMENT HEADER TYPE 2 (FMH-2)

The format of an FMH-2 is shown below. Fields that are common to all FMH-2 formats are discussed. Parameter fields (bytes 3 to n) are not discussed here; they will be described as each FMH-2 is defined.

Byte 2 of the FMH-2 defines the function to be performed by the FMH-2 and contains the stack reference indicator (SRI). Figure 4-8 shows valid combinations of SRI and function codes.

Function Code	Function
X'01'	Peripheral data information record (PDIR)
X'02'	Compaction table
X'04'	Prime compression character
X'07'	Execute program offline
X'20'	Create data set
X'21'	Scratch data set
X'22'	Erase data set
X'23'	Password
X'24'	Add
X'25'	Replace
X'26'	Add replicate
X'27'	Replace replicate
X'28'	Query for data set
X'29'	Note
X'2B'	Record ID
X'2C'	Erase record
X'2D'	Scratch all data sets
X'2E'	Volume ID
X'AA'	Note reply (SRI is always on)

Figure 4-8. FMH-2 functions and their function codes.

General Format:

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows this FMH-2 Another FMH follows this FMH-2
	1-7	B'0000010'	FMH-2 identifier
2	0	SRI B'0' B'1'	Stack reference indicator. FMH-2 pertains to the active destination of the sending half-session's send stack and the receiving half-session's receive stack. 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	Function	FMH-2 function to be performed.
		B'nnnnnnn'	Identifies the function that this FMH-2 is to perform. Note: The FMH-2 descriptions on the following pages combine the SRI and function settings. Figure 4-8 shows valid settings for these fields.
3-n		Parms	Parameter fields. These fields provide the information needed to perform the selected function. They are different for each FMH-2 function. See the individual FMH-2 descriptions below for definitions of the parameter fields.

ADD

Adds a record to a sequential, an addressed direct, or a keyed direct data set named in the FMH-1. For sequential data sets, ADD moves the end-of-data indicator to the next available record indicator. For keyed direct data sets, ADD establishes a valid record ID. For addressed direct data sets, ADD moves the end-of-data indicator and establishes a valid record ID.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'24'	Add FMH-2
3		N	Number of records to be added

Field Description:

N Specifies the number of logical records to be added, which are contained in a chain. If the parameter N is not present in the ADD FMH-2 (length of header = 3), then one record at a time is to be added to the data set named and each chain contains one logical record.

Notes:

1. Key information is carried in the record.

2. A BEGIN FMH-1 followed by FM data causes the sequential addition of a record to the data set named in the BEGIN FMH-1. A record is a chain.

3. If the number of logical records in a chain is less than N, then the chain length overrides N.

4. The logical record length (LRECL) may also be specified. Its presence or absence is dependent on the data set selected by the FMH-1. However, LRECL information is not part of the FMH-2. If present, LRECL is part of each logical record.

   The logical record(s) may be sent or received immediately following the FMH-2 as part of one chain or in the next chain. LRECL is the first four bytes of a logical record and specifies the length, in bytes, of each logical record. The maximum value is 32,760.

Errors:

Record too long (X'10080813')
Data set full (X'10080814')
Number of logical records per chain exceeded (X'10080818')
Data set not found (X'1008081D')

ADD REPLICATE

Replicates (duplicates) a record a specified number of times for the sequential or addressed direct data set named by the FMH-1. Replication starts at the end-of-data indicator. ADD REPLICATE is not valid for keyed direct or keyed indexed data sets.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'26'	Add Replicate FMH-2
3		R	Replication factor

Field Description:

R Indicates the number of times the record is to be replicated. The record either follows the FMH-2 or is sent as the next chain.

Errors:

Function not allowed for destination (X'10080812')

COMPACTION TABLE

Sends a compaction table for use by the destination specified in the FMH-1.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'02'	Compaction Table FMH-2
3		N X'00' X'01' X'02' X'03'-X'10'	Number of master characters. Ignore previously sent table. The value X'00' is used to tell the session partner that subsequent data is no longer compacted. Invalid (see "Errors") Invalid (see "Errors") Number of master characters in the table. Implementations may specify a larger minimum than N = X'03'.
4		Table	Characters to be compacted followed by remaining characters. (Transmitted in row major order, starting at the bottom row and omitting the cells in the upper left corner M-by-M submatrix. (See "Selecting a Compaction Table" in Part 2 Chapter 5 and the example below for further details.) Characters need not be graphics; controls such as new line (NL), form feed (FF), duplicate (DUP), and horizontal tab (HT) may be used.

Errors:

Compaction table outside supported subset (X'10080806'). (The number of master characters does not fall within the valid range.)

Example:

A table of 14 master characters has been chosen for numerical reports that contain numbers, uppercase graphics, and special characters. To use the table, the following header is sent:

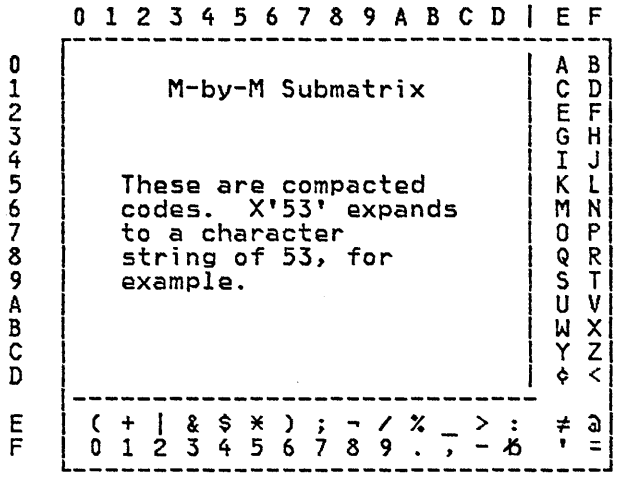
First 3 bytes: X'400202' (For FMH-2)

Next byte: X'0E' (For 14 master characters)

Next 14 bytes: EBCDIC value of the following characters:
0123456789.,-ŏ

Next 18 bytes: EBCDIC value of the following characters:
' = (+ | & $ *) ; ¬ / % _ > : ≠ @

Next 28 bytes: Hexadecimal representation of the following characters:
¢<YZWXUVSTQROPMNKLIJGHEFCDAB



The last row contains the master characters, assigned to X'Fn' for n=0....D

Sample input string for this table: A B C 2 X Y 4 5 6 8

Coded String
X'0E', X'0F', X'1E', X'F2', X'BF', X'CE', X'45', X'68'

In this example, only the last four characters are compactable.

CREATE DATA SET

Creates a data set entry in a directory for a data set to which data records may be added.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'20'	Create Data Set FMH-2
3		DSACCESS X'00' X'01' X'02' X'81'	Data set access method Sequential Addressed direct - basic Keyed direct - unique Addressed direct - restricted All other codes reserved
4		ACCESS X'00' X'01'	Local application access Read/write Read only All other codes reserved
5		RECTYPE X'00' X'01'	Fixed- or variable-length records Fixed length Variable length
6-7		RECLEN	Record length (hexadecimal value, right-justified, zero-padded)
8-11		NUMRECS	Number of records (hexadecimal value, right-justified, zero-padded) All zeros indicate system default.

The remaining fields are optional:

12-n	PASSR	Password for read access for users of the data set being created. Characters left-justified.
n+1-p	PASSW	Password for read/write access and to allow erasure. Characters left-justified. If the respective passwords are not present, PASSR and PASSW are X'00'. If the respective passwords are present, X'xx' indicates the length of the password.

The following fields occur for keyed direct data sets only:

r	KEY1LEN	Length of key 1 (1 byte)
s	KEY1DISP	Displacement to key 1 (2 bytes)
t	KEY1TECH X'01' X'02'	Type of key-1 algorithm (1 byte) Hash type 1 Hash type 2
u	KEY1SPACE	Resolution space (2 bytes) (hexadecimal value, right-justified, zero-padded.)
v	KEY2LEN	Field definitions for bytes v to y are the same as for bytes r to u except for the different key.
w	KEY2DISP	
x	KEY2TECH	
y	KEY2SPACE	

Field Descriptions:

DSACCESS This field defines the type of access to the data set being created. The type selected affects the remaining information carried in the header. The following types of data set access are defined:

Sequential: Records may be added to or erased from the data set; the data set may be queried (read).

Addressed Direct - Basic: Records may be added and subsequently replaced, erased or queried. The data set is created with the end-of-data indicator pointing to the first available record.

Addressed Direct - Restricted: Records may be replaced, erased or the data set queried. The data set is created with the end-of-data indicator pointing one past the last available Record Identifier. The logical records in the data set are not initialized to a particular value. This must be done via the REPLACE FMH-2. If this data is queried following its creation, noninitialized logical records are returned.

Keyed Direct - Unique: Records may be added, erased, or queried.

ACCESS This field defines the local access capability to be applied to the data set -- read/write or read only.

RECTYPE Indicates whether the data set records are fixed or variable length. Fixed length must be specified for addressed direct data sets; fixed and variable length are valid for sequential and keyed direct - unique data sets.

RECLEN Indicates the maximum logical record length when RECTYPE is variable, and actual record length when RECTYPE is fixed.

NUMRECS This value is used to assist in space allocation for sequential and addressed direct data sets. It indicates the maximum number of

records available for subsequent processing. This field has no meaning for keyed direct and keyed indexed data sets.

PASSR This is an optional parameter used to assign a password for read access to the created data set.

PASSW This is an optional parameter used to assign a password for read/write access and erase capability.

KEY1LEN This parameter indicates the length of the key of a keyed direct data set or the length of the first key of a double-keyed data set.

KEY1DISP This parameter defines the displacement from the beginning of each record in the data set to the start of the key.

KEY1TECH This parameter indicates the type of key processing.

KEY1SPACE This parameter indicates the amount of space to be allocated for resolution of keyed records.

Note:

1. A VOLUME-ID FMH-2 must be concatenated to the previous FMH-1 if the data set is to be created on other than the default volume.

Errors:

Data set exists (X'10080819')
No space available (X'1008081A')
Invalid DSACCESS (X'1008081C')
Invalid RECTYPE (X'1008081D')
Invalid ACCESS (X'1008080C')
Invalid RECLEN (X'1008080D')
Invalid NUMRECS (X'1008080E')
Insufficient resolution space (X'1008081E')
Invalid key technique, for example, KEY1TECH=X'20' (X'1008081F')
Invalid key displacement, for example, displacement outside of record length (X'10080820')

ERASE

Erases one or more records from a data set identified in the FMH-1 so that record(s) can no longer be accessed. The RECORD-ID FMH-2 indicates the starting point of the erase operation. ERASE is valid for Addressed Direct and Keyed Direct data sets only.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'2C'	Erase (a record) FMH-2
3		N	Number of records to be erased.

Field Description:

N For addressed and keyed direct data sets, N specifies the number of records to be erased preceding the end-of-data indicator. If N is not present in the FMH-2 for addressed direct data sets, records are erased from the key indicated in the RECORD-ID FMH-2 to the end-of-data indicator. If N is not specified, for keyed direct data sets, ERASE erases a single record indicated by the RECORD-ID FMH-2. If N is specified for keyed direct data sets, N must be one.

Errors:

Function not allowed for data set (X'10080812')
Data set in use (X'1008080F')
Invalid N (N must be 1 if specified for keyed direct data sets) (X'10080822')

ERASE DATA SET

Erases the records within a data set but leaves the data set entry in a directory intact. Records may subsequently be added to the data set without requiring a CREATE DATA SET FMH-2.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'22'	Erase Data Set FMH-2

Notes:

1. If a VOLUME-ID FMH-2 is not concatenated to the previous FMH-1, or not supported, and a cataloged structure does not exist, the receiving LU finds the data set specified by DSNAME in the FMH-1 by serially searching the mounted volumes. The records in the first data set name to match are erased.

2. If a VOLUME-ID FMH-2 is not concatenated to the previous FMH-1, or not supported, and a catalog structure exists, the records in the first cataloged data set specified by DSNAME are erased.

Errors:

Data set not found (X'10080810')
Function not allowed for data set (X'10080812')
Data set in use (X'1008080F')

EXECUTE PROGRAM OFFLINE

Requests the scheduling of a program for execution when the LU-LU session ends.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'07'	Execute Program Offline FMH-2
3			Reserved
4		LENAME	Length of program name (in hex)
5-n		PGMNAME	Number or name of program to be executed (binary encoded)

Errors:

Insufficient resources to perform function requested (X'1008400D')
Invalid PGMNAME -- PGMNAME not found (X'1008082A')

NOTE

Requests the next available record indicator (ID) for the data set named in the FMH-1. The next available record ID is returned by the NOTE REPLY FMH-2, and it must be returned in the next FMH-2 the session partner sends.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'29'	Note FMH-2

Errors:

Function not allowed for data set (X'10080812')

NOTE REPLY

Provides the next available record indicator (ID) to the sender of the NOTE FMH-2. The ID must be returned in the next FMH-2 sent to the session partner after the NOTE FMH-2 is received.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'AA'	Note Reply FMH-2
			Note: The high-order bit (the SRI bit) is set to one, thus pointing to the active destination of the session partner's send stack.
3-n		RECID	Address of the next available record

Errors: None

PASSWORD

Validates the authority of the requester to refer to and change data sets and data set records. This header may be used to establish authority for a named destination (via an FMH-1) so that subsequent FMH-2 operations may be performed.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'23'	Password FMH-2
3-n		Password	Password

Field Description:

Password The word used to validate the authority of the requester. Password must have been specified in the PASSR and PASSW parameters of the CREATE DATA SET FMH-2.

Errors:

Invalid password (X'10080811')

PERIPHERAL DATA INFORMATION RECORD (PDIR)

Requests that the operator perform forms mount, electronic forms control load, train mount, and copy functions.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'01'	PDIR FMH-2
3		PDIR ID X'00' X'01' X'02'	Request identifier. Standard PDIR Job separator PDIR System message PDIR
4-11		DATE	Date of data set creation, in EBCDIC characters. Format: MM/DD/YY Default: 00/00/00
12-19		TIME	Time of data set creation, in EBCDIC characters. Format: HH.MM.SS Default: 00.00.00
20-27		FORMS	Forms to be handled by the operator, in EBCDIC characters, left-justified, blank-padded. Default: blanks, indicating use of the standard forms.
28-35		FCBNAME	FCB name to be used to have the session partner load a printer FCB (electronic forms control buffer) image. EBCDIC characters. Default: blanks, indicating use of the standard FCB.
36-43		TRAIN	Train name (train to be mounted) in EBCDIC characters Default: blanks, indicating use of the standard train.
44-51		COPIES	Number of additional copies to be reproduced. EBCDIC characters right-justified, blank-padded (except for rightmost position). Default = bbbbbbb0
52-59		VOLIO	Volume of I/O in EBCDIC characters, right-justified, suppress leading zeros. Specify the number of print lines or the number of cards. Default: blanks. VOLIO provides the operator with information concerning the relative length of jobs. The value of the parameter therefore need not be precise.
60-99		DSN	DSNAME field, EBCDIC characters, left-justified, blank-padded. Default = blanks. The DSNAME field is subdivided as follows: JOBNAME (bytes 60-67) STEPNAME (bytes 68-75) PROCSTEP (bytes 76-83) DDNAME (bytes 84-91) SPINNO (bytes 92-99) SPINNO (spin number) is a unique identifier to distinguish among data sets with the same JOBNAME.

Note:

1. If a PDIR with the COPIES function specified is received and accepted, the sender transmits the data to be copied n times only once. That is, the receiver of such a PDIR performs the copy function.

Errors:

Invalid function parameters (X'10080801')
Forms function cannot be performed (X'10080803')
Unable to perform copy function (X'10080805')
Invalid PDIR identifier (X'10080807')
Printer train function cannot be performed (X'10080808')
FCB load function cannot be performed (X'10080809')

PRIME COMPRESSION CHARACTER

Sends a prime compression character for use by the destination specified in the FMH-1. The prime character is assumed to be the character blank. If another character is required, this FMH-2 should be sent.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'04'	Prime Compression Character FMH-2
3		Character	Prime compression character if other than blank.

Errors: None

QUERY FOR DATA SET

Requests that the data set named by the FMH-1 be sent to the requester. QUERY FOR DATA SET is valid for sequential, addressed direct, and keyed direct data sets.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'28'	Query for a Data Set FMH-2
3		Mode X'00' X'01'	Query mode. Immediate reply Delayed reply

Field Description:

Mode If immediate reply is specified, the requested data must be sent with the next BEGIN or BEGIN/END FMH-1. If delayed reply is specified, the requested data may be sent on any subsequent BEGIN or BEGIN/END FMH-1.

Error:

Data set not found (X'10080810')

RECORD ID

Selects a specific record within a data set named by the FMH-1.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC	FMH concatenation.
		B'0'	No FMH follows
		B'1'	Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'2B'	Record ID FMH-2
3		KEYIND	Key indicator.
		X'00'	Addressed direct key
		X'01'	Keyed direct KEY1
		X'02'	Keyed direct KEY2
		X'03'	Implementation definition
		X'04'	Control definition
		X'05'	Keyed direct Key 3
		X'06'	Keyed direct Key 4
		X'07'	Keyed direct Key 5
		X'08'	Keyed direct Key 6
		X'09'	Keyed direct Key 7
		X'0A'	Keyed direct Key 8
4-n		RECID	Key of record to be selected.

Notes:

1. For the ADD FMH-2 and REPLACE REPLICATE FMH-2, specify KEYIND of X'00'.
2. For the REPLACE FMH-2 and ERASE FMH-2, specify KEYIND of X'00' if addressed direct data set or X'01', X'02', X'05', X'06', X'07', X'08', X'09', or X'0A' if keyed direct data set.
3. The VOLUME-ID FMH-2 must precede the root FMH-2 that this RECORD-ID FMH-2 accompanies to protect against accessing an identically named data set on another volume.

Errors:

Invalid RECID (X'10080815')
Invalid KEYIND (X'10080823')
Invalid RECID format (X'10080826')

REPLACE

Replaces one or more logical records within the data set named by the FMH-1 with an equal number of logical records. The RECORD-ID FMH-2 indicates the starting point of the replace operation. The end-of-data indicator does not change.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'25'	Replace Record FMH-2
3		N	Number of records to be replaced

Field Description:

N Specifies the number of logical records to be replaced which are contained in a chain. If N is not present in the REPLACE FMH-2 (length of header is 3), then one record is to be replaced in the data set named at a time and each chain contains one logical record.

Notes:

1. The logical length (LRECL) may also be specified. Its presence or absence is dependent on the data set selected by the FMH-1. However, LRECL information is not part of the FMH-2. If present, LRECL is part of each logical record.

2. The logical record(s) may be sent or received immediately following the REPLACE FMH-2 as part of one chain or in the next chain. LRECL is the first four bytes of a logical record and specifies the length, in bytes, of each logical record. The maximum value is 32,760.

Errors:

Function not allowed for destination (X'10080812')
Data set in use (X'1008080F')
Invalid N (X'10080822')

REPLACE REPLICATE

Replaces one or more logical records within the data set named by the FMH-1 with an equal number of identical logical records. The RECORD-ID FMH-2 indicates the starting point of the replace operation. The end-of-data indicator does not change. REPLACE REPLICATE is valid for addressed direct data sets only.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'27'	Replace Replicate FMH-2
3		R	Replication factor

Field Description:

R Indicates the number of records to be replaced with the record that follows. The replication record either follows the FMH-2 or is sent as the next chain following the header.

Errors:

Function not allowed for the data set (X'10080812')
Data set in use (X'1008080F')

SCRATCH ALL DATA SETS

Deletes all data set entries in a directory for the medium, subaddress, and volume-ID specified. Before this FMH-2 is issued, the medium and subaddress are specified in an FMH-1, and the volume ID is specified in a VOLUME-ID FMH-2.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'2D'	Scratch (erase) All Data Sets FMH-2

Notes:

1. If DSNAME is specified in the preceding FMH-1, it is ignored.
2. If a VOLUME-ID FMH-2 is not specified or not supported, the directory entries for all data sets on the medium and subaddress are deleted.

Errors:

Function not allowed for destination (X'10080812')

SCRATCH DATA SET

Deletes the data set entry from a directory for the named data set so that the data set can no longer be referenced. The space associated with the data set name is also freed.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'21'	Scratch Data Set FMH-2

Notes:

1. If a VOLUME-ID FMH-2 is not concatenated to a previous FMH-1, or not supported, and a cataloged structure does not exist, the receiving LU finds the data set specified by DSNAME in the FMH-1 by serially searching the mounted volumes. The first data set name to match is scratched.
2. If a VOLUME-ID FMH-2 is not concatenated to a previous FMH-1, or not supported, and a catalog structure exists, the first data set specified by DSNAME found in the catalog is scratched.

Errors:

Data set in use (X'1008080F')
Data set not found (X'10080810')

VOLUME ID

Specifies a volume ID to be associated with the data set named by the FMH-1. VOLUME ID is not a mount function.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000010'	FMH-2 identifier
2		X'2E'	Volume-ID FMH-2
3-n		VOLID	Volume ID

Errors:

Invalid VOLID (X'1008081B')
Invalid VOLID format (X'10080817')

FUNCTION MANAGEMENT HEADER TYPE 3 (FMH-3)

General Format: The FMH-3 format is identical to the FMH-2 format except that FMH-3s do not have a stack reference indicator (SRI) in byte 2.

An FMH-3 is used when information is needed or used by all destinations managed by the half-session. By contrast, an FMH-2 is used for a specific destination.

Two functions, the COMPACTION TABLE FMH and the PRIME COMPRESSION CHARACTER FMH, can be sent as FMH-2s or FMH-3s. They should be sent as FMH-2s when they apply to a specific destination at the half-session and as FMH-3s when they apply to all destinations at the half-session.

Figure 4-9 shows the FMH-3 functions.

Function Code	Function
X'02'	Compaction table
X'03'	Query for compaction table
X'04'	Prime compression character
X'05'	Status
X'06'	Series ID

Figure 4-9. FMH-3 functions and their function codes.

COMPACTION TABLE

Sends a compaction table for use by all destinations at the half-session.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1		FMHC	FMH concatenation.
	0	B'0'	No FMH follows
	1-7	B'0000011'	FMH-3
2		X'02'	Compaction Table FMH-3
3		N	Number of master characters.
		X'00'	Ignore previously sent table. The value X'00' is used to tell the session partner that subsequent data is no longer compacted.
		X'01'	Invalid (see "Errors").
		X'02'	Invalid (see "Errors").
		X'03'-X'10'	Number of master characters in the table.
			Implementations may specify a larger minimum than N = X'03'.
4		Table	Characters to be compacted followed by remaining characters. Transmitted in row major order, starting at the bottom row and omitting the cells in the upper left corner M-by-M submatrix. (See "Selecting a Compaction Table" in Part 2 Chapter 5 for further details.) Characters need not be graphics; controls such as new line (NL), form feed (FF), duplicate (DUP), and horizontal tab (HT) may be used. An example of a table is shown under "Compaction Table FMH-2" on page 215.

Errors:

Compaction table outside supported subset (X'10080806') (The number of master characters does not fall within valid range.)

PRIME COMPRESSION CHARACTER

Sends a prime compression character for use by all destinations at the half-session. The prime character is assumed to be the character blank. If another character is required, this FMH-3 should be sent.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1		FMHC	FMH concatenation.
	0	B'0'	No FMH follows
	1-7	B'0000011'	FMH-3
2		X'04'	Prime Compression Character FMH-3
3		Character	Prime compression character if other than blank.

Errors: None

QUERY FOR COMPACTION TABLE

Requests that the receiver send the named compaction table.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1		FMHC	FMH concatenation.
	0	B'0'	No FMH follows
	1-7	B'0000011'	FMH-3
2		X'03'	Query for a Compaction Table FMH-3
3-n		CPTBL	Name of the table

Errors:

Invalid compaction table name (X'1008080B')

SERIES ID

Establishes a 4-byte unique identification (ID) for a series of FM headers to assist in correlating the completion of a series of operations. The series is delineated by sending another SERIES-ID FMH-3.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1		FMHC	FMH concatenation.
	0	B'0'	No FMH follows
	1	B'1'	Another FMH follows
	1-7	B'0000011'	FMH-3
2		X'06'	Series ID FMH-3
3-6		SERID	Series identifier

Field Description:

SERID A 4-byte unique identifier to be applied to subsequent FMH operations.

Errors:

Invalid SERID (X'10080824')

STATUS

Communicates the state of completion of a series, user application program, or DSNAME by conveying two bytes of user sense information.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1		FMHC	FMH concatenation.
	0	B'0'	No FMH follows
	1	B'1'	Another FMH follows
	1-7	B'0000011'	FMH-3
2		X'05'	Status FMH-3
3-4		Sense	User-specified status
5		STAT	Type of status.
		X'01'	Series ID
		X'02'	User application program
		X'03'	DSNAME
6-n		STATITEM	Series ID, or user application program name, or data set name.

Errors: None

FUNCTION MANAGEMENT HEADER TYPE 4 (FMH-4)

FMH-4 is used by the sending half-session to describe a block of data (a subset of a logical message) and what to do with the data when it is received by the session partner.

Byte	Bit	Content	Meaning
0		FMHL	Length of header including length byte (≥6)
1	0	FMHC	FMH concatenation (must be B'0').
	1-7	B'0000100'	FMH-4 identifier
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 value is 4.
3		FMH4TT1	Block transmission type
		X'00'-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'	Derivative code
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'	Create
		Other	Reserved
6		FMH4FLAG	Flags. If omitted, X'00' is assumed.
	0-5		Reserved
	6	FMH4BDTF	Block data transform flag
		B'0'	FMH4BDT absent
		B'1'	FMH4BDT present
	7	FMH4RDTF	Reserved
m		FMH4LBN	Length of FMH4BN. Zero, or omitted, if unnamed block.
m+1		FMH4BN	Name of block
n		FMH4LBDT	Length of FMH4BDT. Zero if FMH4BDTF is B'0'.
n+1		FMH4BDT	Block data transform
p		FMH4LVID	Length of FMH4VID
p+1		FMH4VID	Version identifier

FUNCTION MANAGEMENT HEADER TYPE 5 (FMH-5)

FMH-5 is used by the sending half-session to attach a transaction program to the receiving half-session, to change parameters sent in a previous attach, or to reset (detach) a previously-attached transaction program.

ATTACH

The ATTACH FMH-5 is sent by either LU_T6 half-session to select a named transaction program. The ATTACH FMH can be followed by other FM headers and FM data. The ATTACH FMH-5 can optionally be sent with BB, EB, or CD.

Each variable-length field (those fields starting with byte 8) starts with a 1- or 2-byte length field that indicates the number of bytes in the variable-length field. Field FMH5LNSZ indicates whether the length field is one or two bytes; the length indicated does not include the length byte(s).

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC	FMH concatenation.
		B'0'	No FMH follows
		B'1'	Another FMH follows
	1-7	B'0000101'	FMH-5 identifier
2-3		FMH5CMD	Command code
		X'0202'	Attach FMH-5
4		FMH5MOD	
	0	FMH5LNSZ	Length of length fields for variable- and fixed-length parameters
		B'0'	Length field is 1 byte long
		B'1'	Length field is 2 bytes long
	1-2		Reserved
	3	FMH5IUE	Interchange unit end indicator
		B'0'	IU not terminated
		B'1'	IU terminated
	4		Reserved
	5	FMH5IUA	Interchange unit queue access
		B'0'	Not session local queue
		B'1'	Session local queue (only session partners can access queue)
	6-7	FMH5IUT	Interchange unit type
		B'00'	IU spans more than one RU chain
		B'01'	IU terminates at end of RU chain
			Others reserved
5		FMH5FXCT	Length of fixed length parameters
		X'02'	Two bytes of parameters follow
6		ATTDSP	Data stream profile used by transaction program
	0-3	DSP	Data stream profile
		X'0'	User defined
			Others reserved
	4-7	DSPMOD	Modifier for user-defined DSP.
7		ATTDBA	Application data handling algorithm
		X'01'	Variable length, variable blocked
		X'04'	A chain of RUs

Resource Names:

8-m	ATTDPN	Field length (1 or 2 bytes depending on FMH5LNSZ) plus name of transaction program (DPN) to be initiated. For an IBM service (transaction) program, the DPN starts with a nongraphic; that is, the first byte has a value of X'00' to X'3F'(see list below). For all other programs, the DPN starts with a graphic (X'41' to X'FE').

IBM service program	First byte of name
System message program	X'01'
Scheduler program	X'02'
Queue program	X'03'
DL/1 program	X'05'

m+1-n	ATTPRN	Field length (1 or 2 bytes depending on FMH5LNSZ) plus name of primary resource (PRN) for the transaction program being initiated
n+1-p	ATTRDPN	Field length (1 or 2 bytes depending on FMH5LNSZ) plus name of suggested return program name (RDPN)
p+1-q	ATTRPRN	Field length (1 or 2 bytes depending on FMH5LNSZ) plus name of suggested primary resource for the return program (RPN)
q+1-r	ATTDQN	Field length (1 or 2 bytes depending on FMH5LNSZ) plus name of queue to be associated with the DPN
r+1-s	ATTACC	Field length (1 or 2 bytes depending on FMH5LNSZ) plus access code to be validated before session is attached to transaction program (a symbolic name)

Note: Variable- and fixed-length parameters are positional by command code. A length field, which is 1- or 2-bytes long depending on byte 5, precedes each variable-length positional parameter. If the length field is zero the variable parameter is omitted and the next positional variable-length parameter length field occurs followed by its variable-length parameter field.

DATA DESCRIPTOR

The DATA DESCRIPTOR FMH-5 is sent by a half-session when it wishes to change the parameters passed in the initial ATTACH FMH-5 or previous DATA DESCRIPTOR FMH-5. The DATA DESCRIPTOR FMH-5 can be followed by other FM headers and FM data. The DATA DESCRIPTOR FMH-5 must be sent with BB.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC	FMH concatenation.
		B'0'	No FMH follows
		B'1'	Another FMH follows
	1-7	B'0000101'	FMH-5 identifier
2-3		FMH5CMD	Command code
		X'0206'	Data Descriptor FMH-5
4		FMH5MOD	
	0	FMH5LNSZ	Length of length fields for variable- and fixed-length parameters
		B'0'	Length field is 1 byte long
		B'1'	Length field is 2 bytes long
	1-2		Reserved
	3	FMH5IUE	Interchange unit end indicator
		B'0'	IU not terminated
		B'1'	IU terminated
	4-5		Reserved
	6-7	FMH5IUT	Interchange unit type
		B'00'	Multiple-chain IU
		B'01'	Single-chain IU
			Others reserved
5		FMH5FXCT	Length of fixed length parameters
		X'02'	Two bytes of parameters follow
6		ATTDSP	Data stream profile used by transaction program
	0-3	DSP	Data stream profile
		X'0'	User defined
	4-7	DSPMOD	Modifier for user-defined DSP. Reserved when DSP ¬= X'0'.
7		ATTDBA	Application data handling algorithm
		X'01'	Variable length, variable blocked
		X'04'	A chain of RUs

RESET ATTACHED PROCESS (RAP)

The RAP FMH-5 is sent to the attach manager in the receiving half-session to detach the transaction program currently attached. The RAP FMH-5 must be the only FMH in the chain and may not carry FM data. The RAP FMH-5 must carry CD and request a sync point (RQE2 or RQD2).

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC	FMH concatenation.
		B'0'	No FMH follows
		B'1'	Another FMH follows
	1-7	B'0000101'	FMH-5 identifier
2-3		FMH5CMD	Command code
		X'0204'	Reset Attached Process FMH-5
4		FMH5MOD	
	0	FMH5LNSZ	Length of length fields for variable- and fixed-length parameters
		B'0'	Length field is one byte long
		B'1'	Reserved
	1-7		Reserved
5		FMH5FXCT	Length of fixed length parameters
		X'00'	No parameters follow

FMH-5 ERRORS

Error	Sense code
Insufficient resource	0812uuuu
Invalid DSP	080Fuuuu
DPN unavailable	084B6002
PRN unavailable	084B6003
Invalid DPN	10036002
Invalid PRN	10036003
Invalid DBA	10086001
Invalid queue name length	10086004
Invalid DSP	10086006
Invalid ATTACH access code	10086008
Fixed length not 2	10086009
Wrong IUT, IUE not on	1008600A
Invalid command	1008600B
Invalid RAP FMH	1008600F
RAP with inactive attach register	10086010

FUNCTION MANAGEMENT HEADER TYPE 6 (FMH-6)

FMH-6 carries information that relates to work being done by the transaction programs at each end of a type 6 LU-LU session.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC B'0' B'1'	FMH concatenation. No FMH follows Another FMH follows
	1-7	B'0000110'	FMH-6 identifier
2-3		FMH6CMD	Command code (CC2) For IBM service (transaction) programs, the first byte of the command code identifies a program and the second byte identifies a function request within the program.
4		FMH6MOD	Modifier
	0	FMH6LNSZ B'0' B'1'	Length of parameter length fields One-byte parameter length field Two-byte parameter length field
	1-7		Reserved
5-n		FMH6FXCT	Total length of fixed length parameters (LF). This field contains the sum of the lengths of all fixed length parameters which 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		Fxt Parms	Fixed length parameters (FDy). The fixed length parameters are positional by command code.
m+1-p		V length	Length field of first, variable-length parameter (LV1). This field is either 1 byte or 2 bytes in length based on the setting of FMH6LNSZ (0 = one byte, 1 = two bytes). If the length field (LVx) is equal to zero, then the variable parameter is omitted. The next variable-length parameter length (LV2) occurs in byte q+1.
p+1-q		V parm	Variable-length parameter (VP). The LV and VP fields are replicated to represent x number of variable-length parameters according to command code.

FUNCTION MANAGEMENT HEADER TYPE 7 (FMH-7)

FMH-7 is sent from one session partner to the other to define the cause of an error and to provide corrective information. The corrective information is called an ERP message, which can be any information understood by the session partners. An ERP message may contain any of the FM level LU_T6 sense codes.

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC	FMH concatenation.
		B'0'	No FMH follows
		B'1'	Reserved
	1-7	B'0000111'	FMH-7 identifier
2-5		ERPSENSE	SNA sense code, which would appear on error response.
6-7		ERPSEQ	Sequence number of RU chain in which error was detected.

FUNCTION MANAGEMENT HEADER TYPE 10 (FMH-10)

FMH-10 requests that the receiver prepare for a sync point. The receiving session partner, on the next flow, either requests a sync point or aborts the unit of work

The FMH-10 must be the first FM header of an RU chain, and the chain must have the CD bit set on; it may be sent requesting either a definite response or an exception response. For those RU chains that have requested a definite response, a positive response (DR1I = B'1' and RTI = B'0') from the receiving half-session indicates successful receipt and successful validity checking of the RU chain containing the FMH-10 only. For either response category, the receiver may then choose to comply or not comply with the sync point; that is, by sending, on the next flow, a request with the DR2I = B'1' (requesting the sync point), or an FMH-7 (abort).

Byte	Bit	Content	Meaning
0		Length	Length of header including length byte
1	0	FMHC	FMH concatenation.
		B'0'	No FMH follows
		B'1'	Another FMH follows
	1-7	B'0001010'	FMH-10 identifier
2-3		SPCCMD	Command:
		X'0202'	Prepare
			(other values reserved)
4-5		SPCMOD	Modifier. For a prepare command (FMH-10), the modifier indicates DFC 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 DFC 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.

Data compression and data compaction are data-handling techniques used by LU-LU session types 1 and 4 to shorten network transmissions. Data is compressed or compacted by the half-session before the data is sent to the session partner. The session partner then decompresses or decompacts the data before passing it to the end user.

Data can be compressed by eliminating gaps, empty fields, redundancies, or unnecessary data. The technique used within LU-LU session types 1 and 4 is to select one character, called the prime compression character, and replace repetitive sequences of that character with a string control byte (SCB). One SCB can replace up to 63 characters (bytes), but the characters must be a string of the same prime character. An SCB can also define strings of like characters that are not the prime compression character, but to define the string so that the receiver knows which character was compressed, the SCB must be followed by a byte containing the nonprime character.

Data also can be compacted to shorten the length of transmissions. The technique used by LU-LU session types 1 and 4 is to compact 2 bytes of data into 1 byte. Up to 16 characters from the total character set can be compacted, so the end user optionally provides a compaction table to indicate which characters are to be compacted. Again, an SCB is used in the data stream to define the beginning and end of each block of compacted data.

SCB fields are defined in Figure 5-8. Within each SCB, the first 2 bits define the function to be performed, and the last 6 bits provide a count.

When a half-session sends compressed or compacted data, it sets on the compression indicator (CMI) or compaction indicator (CPI) in the FMH-1 beginning the data. Accordingly, a BEGIN, BEGIN/END, or CONTINUE FMH-1 may have the CMI, CPI, or both set on. If either flag is set on by the sending half-session, the half-session must build and insert one or more SCBs into the data stream. An SCB must either begin the chain or follow any FM headers that may be present in the chain.

There is no SCB sensitivity to control information in the data stream. If control information such as SCS controls exists, the SCB treats it as another byte of data.

The settings of CMI and CPI in the FMH-1 control the use of the SCB code points:

CMI	CPI	Definition
0	0	No SCB is present.
0	1	Only code points 00 and 01 are valid.
1	0	Only code points 00, 10, and 11 are valid.
1	1	All code points are valid.

You should use a count greater than 1 and 2 for SCB code points 10 and 11, respectively. Lesser counts result in an expansion of the data.

Before using compaction, you should compare the computation time used to compact and decompact the data against the transmission time saved. Compaction may improve overall throughput when the network's communication paths are overloaded (too much network traffic or possibly slow-speed communication paths). Compaction may not be efficient when the data has few occurrences of master characters and when the network has high-speed or broadband communications paths.

SCB code point	Count	Description
00	000000	Invalid control code
00	nnnnnn	No duplicate characters. Count = number of bytes between this SCB and the next one. nnnnnn is a value from 1 to 63.
10	000000	Invalid control code
10	nnnnnn	Repeated prime character. Count = number of prime characters represented by this SCB. The next byte is the next SCB. nnnnnn is a value from 1 to 63. A space is the default unless a PRIME COMPRESSION CHARACTER FMH-2 or FMH-3 was received before this SCB was received.
11	000000	Invalid control code
11	nnnnnn	Repeated next character. Count = number of times the next character is repeated. The next SCB follows the character to be repeated. nnnnnn is a value from 1 to 63.
01	000000	Invalid control code
01	nnnnnn	Compaction code. Count = number of compacted bytes between this SCB and the next one. nnnnnn is a value from 1 to 63.

Figure 5-8. String control bytes (SCBs). These define how data is compressed and compacted.

CHANGING THE PRIME COMPRESSION CHARACTER

The prime compression character is assumed to be the space character (X'40'). When another character is desired, it can be defined for one destination using the PRIME COMPRESSION CHARACTER FMH-2; or, if desired for all destinations in the session, defined using the PRIME COMPRESSION CHARACTER FMH-3. These FM headers are discussed in Part 2 Chapter 4.

SELECTING A COMPACTION TABLE

Data compaction, as used by LU-LU session types 1 and 4, limits the LU to representing a character using one of two lengths -- 4 bits or 8 bits. Accordingly, 1 byte in the data stream can represent one or two characters of data.

When a character is represented by 4 bits, it is called a master character. There can be up to 16 master characters; however, the more master characters, the fewer the representable characters. This relationship is shown in Figure 5-9. As the number of master characters rises, the maximum number of characters decreases.

Master Characters	Nonmaster Characters	Total Characters
0	Not supported	
1	Note 1	
2	Note 1	
3	244	247
4	236	240
5	226	231
6	214	220
7	200	207
8	184	192
9	166	175
10 (Note 2)	146	156
11	124	135
12	100	112
13	74	87
14	46	60
15	16	31
16	0	16

Note 1: Specifying 1 or 2 master characters is invalid.

Note 2: Using less than 10 master characters is an implementation option.

Figure 5-9. Master character to nonmaster character ratios. The total characters column represents the number of unique characters that may be sent under SCB code point 01 for a given number of master characters.

Since there can be a variety of SCBs in an RU, those that do not use SCB code point 01 are independent of the character restrictions of the compaction set.

End users should examine the characteristics of their data streams and determine the frequency of characters. Those characters used most frequently should be the master characters. Because master characters must occur in pairs in the data stream for them to be compacted, you should search for characters that consistently interrupt master character pairing. The compaction efficiency may be improved if you use the interrupting character(s) as a master character(s).

For compaction efficiency, you should select as many master characters as possible and still represent all the characters in your data stream. By using Figure 5-9, you can determine the correct number.

Once master characters are selected, a table can be constructed and transmitted to the session partner as discussed under "Compaction Table" in both the "Function Management Header Type 2" and "Function Management Header Type 3" sections in Part 2 Chapter 4.

SCB CONCATENATION AND CHAINING RULES

An SCB must either begin the RU chain or follow any FM headers that may be present in the chain.

SCB DATA RULES

If 16 master characters are used, the uncompacted data stream must be an even number of bytes. If it is not even, a second SCB must be created to describe the odd byte.

ACRONYMS AND ABBREVIATIONS

+DR2	Positive response to RU chain sent with DR2I = B'1'
+RSP	Positive response
-RSP	Negative response
¬	Not
=	Equals
::=	Equivalent to
ACTLINK	Activate link
ACTLU	Activate logical unit
ACTPU	Activate physical unit
ADS	Abort destination selection
AID	Attention identification
ASCII	American national standard code for information interchange
ATTN	Attention
BB	Begin bracket
BBI	Begin bracket indicator
BC	Begin chain
BCI	Begin chain indicator
BDS	Begin destination selection
BEDS	Begin/end destination selection
BETB	Between brackets
BIU	Basic information unit
BLK	Block
BSC	Binary synchronous communication
CD	Change direction
CINIT	Control initiate
CDI	Change direction indicator
CDS	Continue destination selection
CMI	Compression indicator
CONT	Contention, continue
CPI	Compaction indicator
CR	Carriage return
CRT	Create
CRV	Cryptographic verification
DACTLU	Deactivate logical unit
DACTPU	Deactivate physical unit
DAF	Destination address field
DBA	Deblocking algorithm
DEL	Delete
DFC	Data flow control
DLC	Data link control
DMU	Destination-unique message unit
DP	Data processing
DPN	Destination program name
DQN	Destination queue name
DR1	Definite response one
DR1I	Definite response one indicator
DR2	Definite response two
DR2I	Definite response two indicator
DSP	Data stream profile
DSSEL	Destination selection
DUP	Duplicate
EAU	Erase all unprotected
EB	End bracket
EBI	End bracket indicator
EBCDIC	Extended binary-coded decimal interchange code
EC	End chain
ECI	End chain indicator
EDI	Enciphered data indicator
EDS	End of destination selection

EM	End of message
EO	Eight ones
EOC	End of card
ERCL	Exchange record length
ERI	Exception reponse indicator
ERP	Error recovery procedure
ERPR	Error recovery procedure, receive
ERPS	Error recovery procedure, send
EUA	Erase unprotected to address
EW	Erase/write
EWA	Erase/write alternate
EXR	Exception request
FAP	Format and protocol
FF	Flip flop
FI	Format indicator
FF	Form feed
FFR	Field formatted record
FFR-FNI	Fixed fields without field separators
FFR-FS	Fixed fields with field separators
FFR-FS2	Fixed fields with or without field separators
FM	Function management, field mark
FMD	Function management data
FMDS	Function management data services
FMH	Function management header
FMH-n	Function management header type n
FMP	Function management profile
FSM	Finite-state machine
GCID	Global coded graphic character set identifier
GE	Graphic escape
HDX	Half-duplex
HDX-CONT	HDX contention
HDX-FF	HDX flip-flop
HSCB	Half-session control block
HSID	Half-session identification
IC	Insert cursor
ID	Identification
INB	In bracket (state)
INPID	Inbound partition identifier
INOP	Inbound operation
ITA	International telephone alphabet
IU	Interchange unit
IUE	Interchange unit end
IUT	Interchange unit type
LCID	Local coded graphic character set identifier
LLID	Structured field length (LL) and identifier (ID)
LRC	Longitudinal redundancy character
LRH	Logical record header
LU	Logical unit
LU_Tn	LU-LU session type n
LUSTAT	Logical unit status
MF	Modify field
MGR	Manager
MPL	Maximum presentation line
MPP	Maximum presentation position
MS	Media services
MSG	Message
MSR	Magnetic slot reader
MU	Message unit
NAU	Network addressable unit
NC	Network control
NS	Network services
NU	Nonshared and unnamed
NUL	Null
OAF	Origin address field

P-->S	Primary to secondary
PC	Path control
PIU	Path information unit
PLU	Primary logical unit
PRI	Primary
PRN	Primary resource name
PS	Presentation services, programmed symbols
PSA	Presentation space altered
PSE	Presentation space error
PT	Program tab
PU	Physical unit
QC	Quiesce complete
QEC	Quiesce at end of chain
QRI	Queue response indicator
RA	Repeat to address
RAP	Reset attached program
RB	Read buffer
RCD	Request change direction
RCV	Receive
RDPN	Return destination program name
RDS	Resume destination selection
RELQ	Release quiesce
RH	Request/response header
RM	Read modified
RMA	Read modified all
RPL	Replace
RPRN	Return primary resource name
RQD	Request definite response
RQD2	Request definite response with DR2I on
RQE	Request exception response
RQE2	Request exception response with DR2I on
RQR	Request recovery
RSHUTD	Request shutdown
RSP	Response
RT	Response type
RTI	Response type indicator
RTR	Ready to receive
RU	Request unit
RWS	Read/write storage
S-->P	Secondary to primary
SA	Set attribute
SBA	Set buffer address
SC	Session control
SCB	String control byte
SCS	SNA character string
SDLC	Synchronous data link control
SDS	Suspend destination selection
SDT	Start data traffic
SEC	Secondary
SF	Start field
SFE	Start field extended
SHUTC	Shutdown complete
SHUTD	Shutdown
SIG	Signal
SLU	Secondary logical unit
SN	Shared and named
SNA	Systems network architecture
SP	Start print
SPS	Session presentation services
SRI	Stack reference indicator
SS	Start sentinel
SSCP	System services control point
STSN	Set and test sequence numbers
SU	Shared and unnamed
SUB	Substitute
SUW	Synchronized unit of work
SVC	Service(s)
SYNC	Synchronization

TC	Transmission control
TH	Transmission header
TM	Top margin
TS	Transmission dubsystem
TSP	TS profile
UPD	Update
UPM	Undefined protocol machine
VFC	Vertical forms control
WCC	Write control character
WP	Word processing
WSF	Write structured field
3270	IBM 3270 information display system
3277	IBM 3277 display station

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

Note: A value of X'ab' in byte 10 or byte 11 of BIND represents a•2**b. For example, X'C5' represents (in decimal) 12 times 2 to the 5th power, or 384.

S

Systems Network Architecture — Sessions Between Logical Units Systems

GC20-1868-2

READER'S COMMENT FORM

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

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

_____

_____

_____

_____

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

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

GC20-1868-2

Reader's Comment Form

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

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 824
1133 Westchester Avenue
White Plains, New York 10604

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

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

Cut or Fold Along Line

Systems Network Architecture — Sessions Between Logical Units Systems Printed in U.S.A. GC20-1868-2