IBM

3270 Information
Display System

Data Stream Programmer's Reference

3270 Information
Display System

Data Stream Programmer's Reference

GA23-0059-4

File Number
S360/S370/53/4300/8100-30

| **Fifth Edition (December 1988)**

This publication introduces and explains the functions of the 3270 Information Display System data stream. Changes are continually made to the information herein; before using this publication in connection with the operation of IBM systems, refer to the latest IBM System/360 or System /370 SRL Newsletter for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

The names and addresses used in the examples that appear in this manual are fictitious, and any similarity to the names and addresses of actual persons is entirely coincidental.

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 is provided at the back of this publication. Address additional comments to IBM Corporation, Communication Products Information Development, Department E02, PO Box 12195, Research Triangle Park, North Carolina, U.S.A. 27709. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Choosing the Right Book from the 3174 Library

The 3174 library contains information for installing, customizing, operating, maintaining, and programming the data stream for the 3174 control unit. The list below shows the manuals you need to perform these tasks.

To Organize Library Materials:

Binders and Inserts, SBOF-0089
Binder, SX23-0331
Inserts, SX23-0332

To Become Familiar with the 3174:

Master Index, GC30-3515
3270 Information Display System Introduction, GA27-2739

To Prepare Your Site for the 3174:

Site Planning, GA23-0213
Physical Planning Template, GX27-2999

To Set Up and Operate the 3174:

Models 1L, 1R, 2R, and 3R User's Guide, GA23-0337
Models 51R, 52R, and 53R User's Guide, GA23-0333
Models 81R and 82R User's Guide, GA23-0313

To Plan for and Customize the 3174:

Customizing Guide, GA23-0214
Central Site Customizing User's Guide, GA23-0342

To Install Features or Convert Models on the 3174:

Encrypt/Decrypt Adapter Customer Installation and Removal Instructions, GA23-0262
Fixed Disk and Diskette Drive Customer Installation and Removal Instructions, GA23-0263
Terminal Multiplexer Adapter Customer Installation and Removal Instructions, GA23-0265
Model Conversion Customer Setup Instructions, GA23-0295
IBM Token-Ring Network 3270 Gateway Customer Installation and Removal Instructions, GA23-0329
Storage Expansion Feature Customer Installation and Removal Instructions, GA23-0330
Communications Adapter Customer Installation and Removal Instructions, GA27-3830
Asynchronous Emulation Adapter Customer Installation and Removal Instructions, GA23-0341

To Plan for and Use the Asynchronous Emulation Adapter Feature:

Asynchronous Emulation Adapter Introduction, GA23-0331
Terminal User's Reference for Expanded Functions, GA23-0332

To Use the Multiple Logical Terminals Function:

Terminal User's Reference for Expanded Functions, GA23-0332
Customizing Guide, GA23-0214

To Perform Problem Determination:

Customer Extended Problem Determination, GA23-0217
Status Codes, GA27-3832

To Obtain Data Stream Programming and Reference Information:

Functional Description, GA23-0218
Data Stream Programmer's Reference, GA23-0059
Character Set Reference, GA27-3831

To Perform Maintenance (Service Personnel):

Models 1L, 1R, 2R, and 3R Maintenance Information, SY27-2572
Models 51R, 52R, and 53R Maintenance Information, SY27-2573
Models 81R and 82R Maintenance Information, SY27-2584

To Find Translations of Safety Notices:

Safety Notices, GA27-3824

Preface

Who This Book Is For

This book is for the programmers who need to know what is involved in using the 3270 data stream to produce panels or information at displays and printers.

For those programmers who write the access method macro instructions or other input/output (I/O) instructions, this manual is to be used in conjunction with the appropriate access method or IBM program product publications.

This book is also for those programmers who plan and code for support of Systems Application Architecture (SAA). This architecture is supported by the 3270 data stream and this book contains information about the architecture and its relationship to the 3270 data stream.

How This Book Is Organized

This book has twelve chapters and six appendixes:

- Chapter 1, "The 3270 Data Stream: Overview and Concepts," introduces the 3270 data stream, gives an overview of it, and discusses 3270 data stream concepts.
- Chapter 2, "Partitions," covers partitions and functions related to partitions, such as INPID, INOP, and PWAIT.
- Chapter 3, "3270 Data Stream Commands," describes the commands used in the 3270 data stream and their operations. Those commands that are necessary for SAA support are designated.
- Chapter 4, "3270 Data Stream Orders and Attributes," describes the orders and attributes used in the 3270 data stream and how they function. It also explains character set properties. Those orders that are necessary for SAA support are designated.
- Chapter 5, "Outbound Structured Fields" lists the outbound structured fields in alphabetic order, gives their syntax, and describes how they function. It also lists the outbound/inbound structured fields and gives the same information for them. Those outbound structured fields that are necessary for SAA support are designated.
- Chapter 6, "Inbound Structured Fields" lists the inbound structured fields, including query replies, in alphabetic order, gives their syntax, and describes how they function. Those inbound structured fields that are necessary for SAA support are designated.
- Chapter 7, "Magnetic-Reader, Keyboard, and Selector Pen Operations," describes how magnetic readers work with the 3270 data stream, the keyboard functions that affect data stream operation, and the use of the selector pen.
- Chapter 8, "Printer Considerations," covers printer functions with the 3270 data stream and the local-copy function for Systems Network Architecture (SNA) and binary synchronous communications (BSC).

- Chapter 9, "Binary Synchronous Communications (BSC) Environment," discusses a BSC environment and describes the differences in operation from an SNA environment for the 3270 data stream.
- Chapter 10, "Non-SNA Environment (Locally Attached Devices—3272 Version)," discusses a non-SNA environment of locally attached devices (3272 version) and describes the differences in operation from an SNA or BSC environment.
- Chapter 11, "Auxiliary Devices and Work Stations."
- Chapter 12, "Double-Byte Coded Character Set (DBCS) Asia."
- Appendix A, "SNA Sense Codes," summarizes the sense codes returned for data-stream errors.
- Appendix B, "SNA Sense Codes for 3270 Data Stream Structured-Field Errors," summarizes the sense codes returned for structured-field errors.
- Appendix C, "Reset Actions," summarizes the reset actions for the 3270 data stream.
- Appendix D, "12-, 14-, and 16-bit Addressing," explains the addressing used in the 3270 data stream.
- Appendix E, "Special Applications."
- Appendix F, "Functions Supporting Systems Application Architecture (SAA)"

Other Books You May Need

The following publications provide a general introduction to the 3270:

- *IBM 3270 Information Display System Introduction*, GA27-2739
- *IBM 3270 Information Display System Library User's Guide*, GA23-0058
- *IBM 3270 Information Display System Color and Programmed Symbols*, GA33-3056
- *IBM 3270 Information Display System*
3271 Control Unit
3272 Control Unit
3275 Display Station
Description and Programmer's Guide, GA23-0060
- *IBM 3270 Information Display System*
3274 Control Unit
Description and Programmer's Guide, GA23-0061
- *IBM 3270 Information Display System*
3276 Control Unit Display Station
Description and Programmer's Guide, GA18-2081
- *IBM Systems Reference Library*
General Information - Binary Synchronous Communications, GA27-3004
- *IBM 3278 Display Station Description*, GA18-2127
- *IBM 3192 Display Station Description*, GA18-2535
- *IBM 3193 Display Station Description*, GA18-2364
- *IBM 3179 Color Display Station Description*, GA18-2177

- *IBM 3194 Device Functional Interface Programming Guide*, SA23-0314
- *IBM 3290 Information Panel Description and Reference*, GA23-0021
- *IBM 3174 Subsystem Control Unit Functional Description*, GA23-0218
- *IBM Office Information Architectures: Concepts*, GC23-0765.

The following publications provide a general introduction to Systems Application Architecture (SAA):

- *An Overview*, GC26-4341
- *Common User Access: Panel Design and User Interaction*, SC26-4351
- *Writing Applications: A Design Guide*, SC26-4362
- *Application Generator Reference*, SC26-4355
- *C Reference*, SC26-4353
- *COBOL Reference*, SC26-4354
- *Communications Reference*, SC26-4399
- *Database Reference*, SC26-4348
- *Dialog Reference*, SC26-4356
- *FORTRAN Reference*, SC26-4357
- *Presentation Reference*, SC26-4359
- *Procedures Language Reference*, SC25-4358
- *Query Reference*, SC26-4349.

For access methods, refer to the appropriate access method library used in your environment.

Contents

Figures

Tables

Summary of Changes

This revision incorporates the following new information:

- Outbound Structured fields

  Begin/End of File
  Load Color Table
  Load Programmed Symbols
  Request Recovery Data
  Restart
  Select Color Table
  Set Printer Characteristics

- Inbound Structured fields

  Query Replies
    Query Reply (Character Sets)
    Query Reply (Cooperative Processing Requestor)
    Query Reply (Data Chaining)
    Query Reply (Data Streams)
    Query Reply (DBCS-Asia)
    Query Reply (Device Characteristics)
    Query Reply (Settable Printer Characteristics)
    Query Reply (Text Partitions)
    Query Reply (Transparency)
    Query Reply (Usable Area)

In addition to the above, the following functions that enable 3270 Data Stream support for Systems Application Architecture are included:

- Query Replies

  Query Reply (Character Sets)
  Query Reply (Implicit Partition)
  Query Reply (Null)
  Query Reply (Summary)
  Query Reply (Usable Area)

- Structured Fields

  Read Partition
  Erase/Reset
  Outbound 3270DS.

- Basic 3270 Commands

  Erase All Unprotected (EAU)
  Erase/Write
  Erase/Write Alternate
  Read Buffer
  Read Modified
  Read Modified All
  Write
  Write Structured Field

- Basic 3270 Orders

  Start Field
  Set Buffer Address
  Program Tab
  Insert Cursor
  Repeat to Address
  Erase Unprotected to Address

- 3270 Controls/Special Characters.

Changes are indicated by a vertical line to the left of the changed text.

Chapter 1. The 3270 Data Stream: Overview and Concepts

Introduction

This chapter introduces the 3270 data stream, gives an overview of what it is and what it does, and discusses data stream concepts. This manual describes all 3270 data stream functions that are implemented at the time of publication. However, just because this manual describes a function does not mean that your particular device implements that particular function. Consult your product library for the 3270 data stream functions implemented by your products.

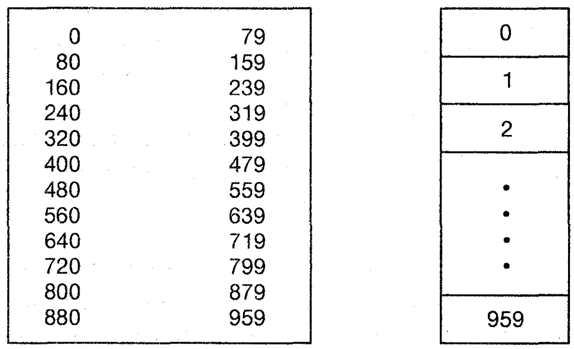
The 3270 data stream is a formatted data stream used for transmitting data between an application program and a terminal. The 3270 data stream supports both display and printer functions. The SNA format is:

LH	TH	RH	3270 Data Stream	LT

The LH, TH, RH, and LT are the link header, transmission header, request/response header, and link trailer, respectively. See Chapter 9, "Binary Synchronous Communications (BSC) Environment" and Chapter 10, "Non-SNA Environment (Locally Attached Devices—3272 Version)" for non-SNA formats.

Overview

The 3270 data stream is based upon the presence of a mapped character buffer in the device. There is a fixed one-to-one relationship between each character storage location in the buffer and each character position on the display. For example, if the display surface consists of 12 rows and 80 columns, row 1 maps to the first 80 character storage positions in the character buffer, row 2 maps to the second 80 character storage positions, and so on. The sequence is the same regardless of the size of the display. (See Figure 1-1.)



80-Character-per-Line Display

960-Byte Character Buffer

Figure 1-1. Mapping the Display to the Character Buffer

The data stream controls the processing and formatting of data with commands, orders, control characters, attributes, or structured fields. The command byte defines the function to be performed by the display.

The data stream allows the application program to divide the display surface into several areas, one of which will be an active area. Each area is related to a *partition*, and the partition that is active is the one that the operator is using to enter data or requests.

The character buffer may contain codes for graphic characters or field attributes. Because each storage location in the character buffer is mapped to a position on the display screen, the field attribute takes up a character position on the display screen. The 3270 field attribute defines a field as that field attribute position plus the character positions up to, but not including, the next field attribute in the character buffer.

Warning: Installation of the Country Extended Code Page (CECP) RPQ may cause a loss of data integrity in databases and applications. Please read the *IBM 3174 Functional Description*, GA23-0218.

Data Stream Format

An application program sends data (and control instructions) to the display by means of commands. An *outbound* data stream is a data stream sent from the application program to the device, and contains:

Command	WCC	Data

OR

WSF	Structured Field	....	Structured Field

The write control character (WCC) is used with write type commands only. (WSF in the example above stands for Write Structured Field, a command that is explained later in this book.)

An *inbound* data stream is sent from the device to the application program and consists of an attention identifier (AID) followed by data. An inbound data stream may also consist of an AID (X'88') followed by structured fields.

AID	Cursor Address (2 bytes)	Data

OR

AID (X'88')	Structured Fields

In many cases the data is optional.

Attention Identifiers (AIDs)

An AID, which is always the first byte of an inbound data stream, describes the action that caused the inbound data stream to be transmitted.

3270 Data Stream Commands

Commands control such things as whether the application program writes to or reads from a display and whether the screen is erased before new data is written.

An application program sends data (and control characters) to the display by means of commands. The commands and data used by the application program and display are as follows:

Command	Data after Command Can Include:
Write	A WCC, orders and data
Erase/Write	A WCC, orders and data
Erase/Write Alternate	A WCC, orders and data
Erase All Unprotected	None
Read Modified	None
Read Modified All	None
Read Buffer	None
Write Structured Field	Structured fields

In the 3270 data stream, some write commands are followed by a control character, the write control character (WCC). The WCC is used to perform such functions as sounding the audible alarm, formatting the printer, and restoring or enabling the keyboard.

3270 Data Stream Orders

Orders are instructions in the 3270 data stream that provide control information. The orders that can be sent with the write commands are:

- Set Buffer Address (SBA)
- Start Field (SF)
- Start Field Extended (SFE)
- Modify Field (MF)
- Set Attribute (SA)
- Insert Cursor (IC)
- Program Tab (PT)
- Repeat to Address (RA)
- Erase Unprotected to Address (EUA)
- Graphic Escape (GE).

The orders that can be included in the inbound data stream are:

- Set Buffer Address
- Start Field
- Start Field Extended
- Set Attribute
- Graphic Escape.

3270 Data Stream Attributes

Attributes determine the properties of a field or of characters within a field. The display uses three kinds of attributes: field attributes, extended field attributes, and character attributes.

Data

Data is the information transferred between the application program and the display. It may be used or acted upon by either the application program or the operator, or by both.

Structured Fields in the 3270 Data Stream

Structured fields are used to convey additional control functions and data to or from the terminal. The Write Structured Field (WSF) command is used to transmit structured fields outbound. There are three types of structured fields: outbound, outbound/inbound, and inbound. The outbound and the outbound/inbound structured fields are defined in Chapter 5, "Outbound Structured Fields."

An AID (X'88') indicates inbound structured fields. The inbound structured field functions that can be sent by the display are defined in Chapter 6, "Inbound Structured Fields."

Concepts

The display shows data that has been transmitted to it from an application program or data that has been entered by the operator. The displayed data can be modified or deleted by the operator (or by further transmissions from the application program), and the revised data can be transmitted back to the application program for storage or additional processing.

Data received from the application program, or data to be transmitted to the application program, is stored in a buffer and is displayed on the screen in the form of alphanumeric characters and symbols. The displayed data is updated when the buffer data is modified by the operator and when new data is received from the application program.

Unformatted and Formatted Screens

An application program can use either a *formatted* or an *unformatted* screen to communicate with a display operator:

- A formatted screen is organized into fields by the application program. Fields are defined by the field or extended field attributes.
- An unformatted screen is regarded as one with a storage and display area that the operator uses in free-form manner. There are no fields defined.

The example in Figure 1-2 on page 1-6 illustrates the versatility of formatted fields. In this example, the solid characters represent the displayed form of characters stored in the character buffer. The squares represent buffer locations occupied by control characters called *field attribute characters* (which are actually displayed as blanks). The dotted characters represent a field of data stored in the buffer but defined by the program as nondisplayable that is not visible to the operator.

NAME : JOHN B DOE
SALARY 1 2 5 2 3
JOB TITLE : WRITER
PHONE #: 383-7628

Figure 1-2. Example of Formatted Fields

Kinds of Attributes

Three kinds of attributes are used in the 3270 data stream: field attributes, extended field attributes, and character attributes. Field and extended field attributes define the start of a field and control the characteristics of the field. Character attributes control the characteristics of a character.

The field attribute occupies a character location in the character buffer and is stored in such a way that it is always displayed as a blank. The extended field attributes and character attributes do *not* occupy positions in the character buffer, but are nevertheless stored and do control the characteristics of the field and characters, respectively.

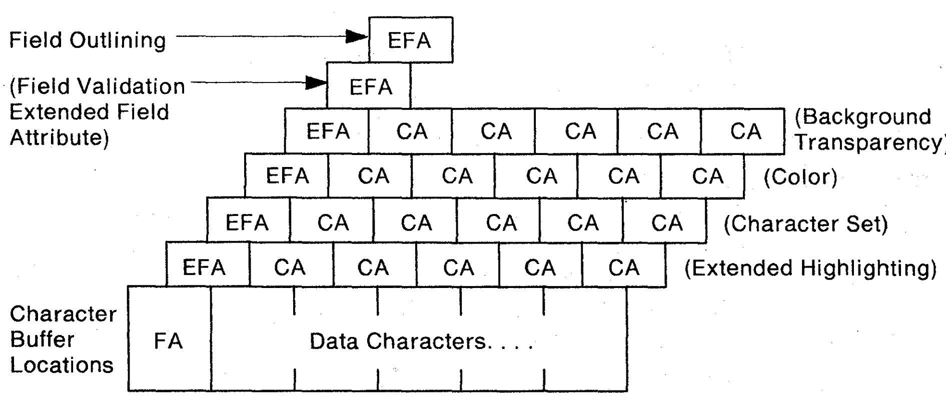
Conceptually, the extended field attributes are extensions of the field attribute, as shown in Figure 1-3.

Field Attributes

The field attribute is defined by the application program for the following purposes:

- To define the start of a field. A field consists of the field attribute and all the data following it, up to (but not including) the next field attribute. A field can wrap (continue) from the end of one row to the beginning of the next row within the presentation space. A field can also wrap from the last location in the presentation space to the first location. In any case, the field is terminated by the next field attribute.

  There is no limit to the number of fields that can be defined, other than that imposed by the screen size.



CA = Character attribute
EFA = Extended field attribute
FA = Field attribute

Figure 1-3. Character and Extended Field Attributes—A Conceptual View

- The 3270 field attribute defines the following field characteristics:
  - *Protected or Unprotected.* A protected field is protected from modification by the operator. An unprotected field is available for the operator to enter or modify data. The unprotected definition classifies a field as an input field.
  - *Alphanumeric or Numeric.* Subject to its being unprotected, an alphanumeric field is one into which an operator enters data normally, using the shift keys (uppercase/lowercase or numeric/alphabetic) as required.

    Fields defined as numeric will accept all uppercase symbols and numerics from a data entry-type keyboard. On a typewriter-type keyboard, numeric has no meaning and all entries are accepted.
  - *Autoskip.* A field defined as protected *and* numeric that causes the cursor to skip over this field.
  - *Nondisplay or Display/Intensified Display.* The selected characteristics apply to the entire field. *Nondisplay* means that any characters entered from the keyboard are entered into the buffer for possible subsequent transmission to the application program but they are not displayed. *Intensified display* means the intensified characters appear on the screen brighter than the nonintensified characters. Some devices may not be able to intensify characters on the screen and will therefore highlight in a different manner.
  - *Detectable or Nondetectable.* A field defined as detectable can be detected by the selector pen or the cursor select key, subject to the use of a designator character.

Base color (four colors) can be produced on color displays and color printers from current 3270 application programs by use of combinations of the field intensify and field protection attribute bits. For more information on color, refer to *IBM 3270 Information System: Color and Programmed Symbols*, GA33-3056.

Extended Field Attributes

The extended field attribute provides additional field definition beyond that provided by the field attribute. The extended field attribute defines field characteristics such as color, character set, field validation, field outlining, and extended highlighting. The extended field attribute is always associated with a field attribute. (See Figure 1-3 on page 1-6)

Character Attributes

A character attribute is associated with an individual character to define characteristics such as character color, character highlighting, or character set. The extended field attributes of any single character are superseded by the character attributes associated with it. However, characters in nondisplay fields are never displayed. The attribute structure used for character attributes is the same as for extended field attributes. (See Table 1-1.)

The attribute structure used for extended field attributes defines all characteristics with attribute type-value pairs, as shown in Table 1-1. Each attribute type has associated with it a set of attribute values.

Attribute Pair -	Attribute Type	Attribute Value

Table 1-1 (Page 1 of 2). Attribute Pairs	
Attribute Type	**Attribute Value**
Extended highlighting	Default—If used as a character attribute, the default is the characteristics defined by the extended field attribute. If used as an extended field attribute, the default becomes those characteristics indicated by the Query Reply (Highlight) structured field. Underscore—Each character or field is underlined. Blink—Each character or field affected is caused to flash on and off. Reverse Video—In each character cell affected, the on/off illumination state every display point is reversed. The effect is analogous to white on black becoming black on white.
Color	Default—If used as a character attribute, it assumes the characteristics of the extended field attribute. If used as an extended field attribute, it is indicated by the Query Reply (Color) structured field. Multicolor—Indicates that the color is defined by a triple-plane Programmed Symbol set. All others—Assigned to the color identifications as indicated by the Query Reply (Color) structured field.
Character set	Default—If used as a character attribute, it assumes the characteristics of the extended field attribute. If used as an extended field attribute, it is the nonloadable character set that has the LCID of X'00' in the Query Reply (character sets) structured field. Local Character Set ID—For the loadable or nonloadable character set.
Field validation	Mandatory Entry—A field that must be modified by the operator before the operator can transmit any data from the display. Mandatory Fill—A field that, if modified by the operator, must be filled with characters other than the null character before the operator can move the cursor out of the field or transmit any data from the display. Trigger—A field that, if modified by the operator, is transmitted inbound as soon as the operator tries to move the cursor out of the field. This allows the application program to receive and to validate fields one by one.

Table 1-1 (Page 2 of 2). Attribute Pairs	
Attribute Type	**Attribute Value**
Field outlining	Default—No fields outlined. Outlining—Sixteen kinds of outlining can be defined by the combinations of the four horizontal and vertical lines.
Transparency	Default—If used as a character attribute, it assumes the characteristics of the extended field attribute. If used as an extended field attribute, the default is determined by whatever the device supports on its inbound response to a query reply. Transparent—Picture elements (pels) of character background are ignored: underlying presentation space can be viewed. Opaque—Pels set as indicated. Underlying presentation space cannot be viewed.

The Cursor

The cursor is a special mark that is displayed on the screen to indicate where the next action from the keyboard will take effect (for example, where the next character keyed in will appear on the screen).

The cursor can be moved by the operator or by instructions from the application program.

A character entered at the keyboard is stored in the display buffer (and is displayed) at the cursor position. Then the cursor advances one position and is ready for the next character to be entered. In order to enter a character, the cursor must be positioned in an unprotected field. If a character already exists at the current cursor position, then (except in insert mode) that character is overwritten by the entered character.

Partitions

If the display supports partitioning (see Chapter 2, "Partitions" for more information on partitions), the application program can define a logical area called a *partition* that may differ in both size and shape from the physical display screen. The partition is defined by use of the Create Partition structured field. (Structured fields are discussed in Chapter 5, "Outbound Structured Fields" and Chapter 6, "Inbound Structured Fields.") Once a partition has been created, data is transmitted to and from the partition as if it were a physical screen with the geometrical characteristics specified in the Create Partition structured field. The mapping of the presentation space to the physical screen is transparent to the application program.

Explicit Partitioned and Implicit Partition States

The 3270 data stream supports displays in both explicit partitioned and implicit partition states. The initial state of a display supporting partitions is the implicit partition state (for example, power-on reset). Interaction with implicit partition 0 on a display that supports partitioning is the same as on a display that does not support partitioning.

The application program can replace the implicitly created partition by *explicitly* creating one or more partitions of its own, thereby placing the display in explicit partitioned state.

The display may be returned to implicit partition state from explicit partitioned state by sending an Erase/Write or Erase/Write Alternate command with bit 1 of the WCC set equal to 1.

Read Functions for a Partition

The transmission of data from a partition can be initiated either by the application program or by the display operator the same as for a nonpartitioned display surface. The display operator, however, can initiate a transmission only from an active partition. The application program can initiate a transmission from any partition by using the Read Partition structured field.

The Read Partition structured field provides the same read functions for partitions as the read commands do for nonpartitioned screens, for example, Read Modified and Read Buffer.

Chapter 2. Partitions

Introduction

Partitioning allows the application program to define a presentation space that may be different in both size and shape from the physical display screen. Multiple partitions can be defined for the display that will allow the display screen to be divided into several rectangular areas called *viewports*, where data from multiple partitions may be displayed on the same physical display screen. The viewport is the area used to display and enter data for its partition. (See Figure 2-1 on page 2-2.)

The application program can organize a partition as either formatted or unformatted.

Each partition has a unique partition identifier (PID) assigned at creation time. The PID identifies the partition so that the application program can send data to, or receive data from, individual partitions.

Similarly, the operator can enter, delete, or modify data in any selected partition (except in protected fields) by positioning the cursor appropriately within the partition's viewport.

If the application program does not define any partitions, the device assumes a single partition of default size with the PID equal to 0. This is referred to as the *implicit partition.*

Character Buffer and Concept of Presentation Space

The amount of storage in the character buffer that is used by the partition is defined by the application program. Data in this buffer may be interacted with by the application program and by the operator. The character buffer provides storage for characters that may be displayed on the display screen. This character buffer is simply addressable storage that contains as many locations as there are character positions in the partition. Each buffer location contains one character and is separately addressable.

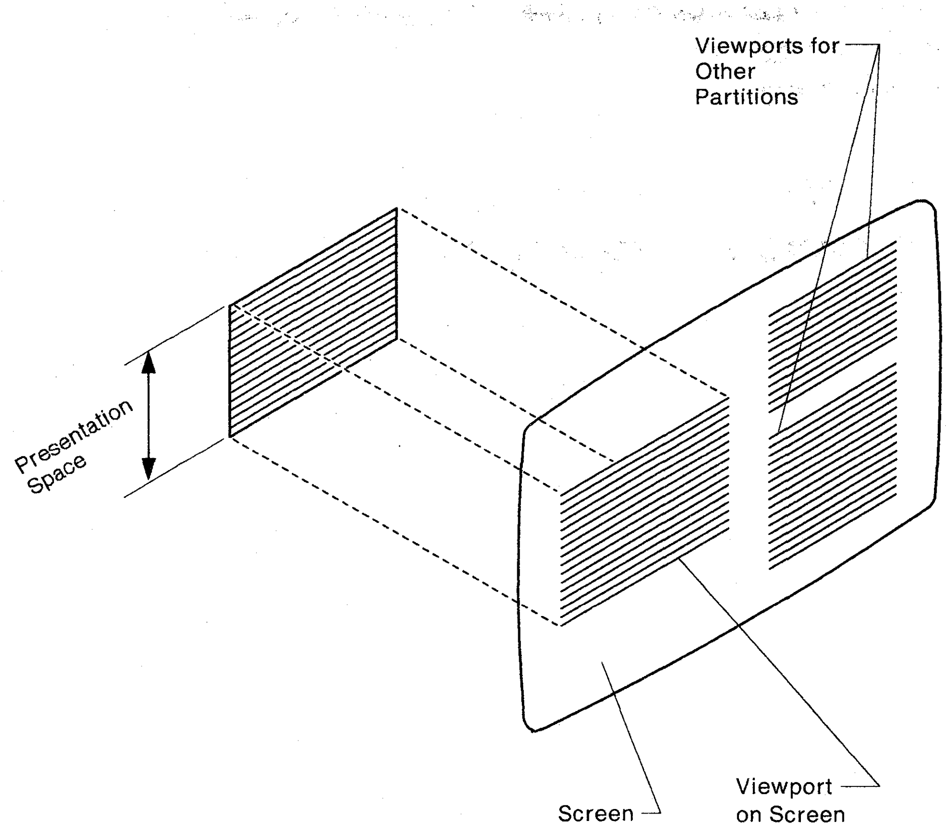


Figure 2-1. Presentation Space and Viewport (without Scrolling)

Conceptually, however, a partition can be regarded as a two-dimensional presentation space whose size is defined in terms of its depth H (number of rows) and its width W (number of columns). Thus, the character buffer associated with this partition is defined to be WxH bytes. The addresses of the character buffer locations range from 0 to (W x H) -1.

Presentation Spaces, Windows, and Viewports

A partition has associated with it a conceptual two-dimensional surface, called the *presentation space.* Data may be thought of as being presented on this two-dimensional surface, although the surface does not exist physically as such on the device. (See Figure 2-2 on page 2-3.) A *window* on the presentation space identifies that part of the presentation space available for viewing on the display surface.

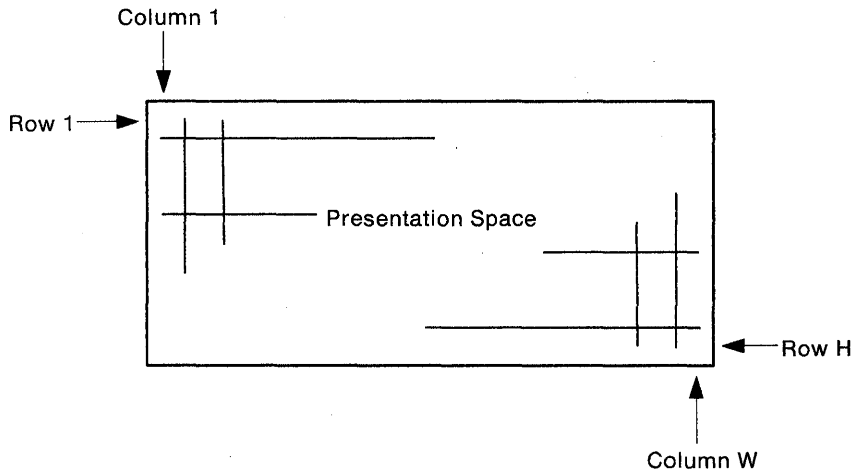


Figure 2-2. The Presentation Plane—A Conceptual View

The viewport is that area on the display surface where the terminal operator sees the area of the presentation space bounded by the window. Each viewport is related to a window so that the area of the presentation space within the window appears on the screen within the viewport. The combination of the viewport and associated presentation space is a partition. (See Figure 2-2.)

For processing 3270 data streams, a coordinate system must be defined on the presentation space. Rather than formatting data on the presentation space in row/column coordinates, 3270 compatibility requires lineal addressing of the presentation space (the character buffer) by use of the 3270 Set Buffer Address order. (Orders are discussed in Chapter 4, "3270 Data Stream Orders and Attributes.")

Relationship Between Presentation Space and Viewport

When the display function scrolling is not implemented, a partition's presentation space and viewport have the same dimensions. (See Figure 2-1.) Assuming it is not in a nondisplay field, each data character in the presentation space is displayed in the corresponding row and column of the viewport.

When the presentation space is larger than the viewport, the viewport displays the data from the window. (See Figure 2-3 on page 2-4.) The window and the viewport have the same dimensions (rows and columns). Assuming it is not in a nondisplay field, each data character in the window is displayed in the corresponding row and column of the viewport. The position of the window in relation to the data on the presentation space can be altered by an action known as *scrolling*.

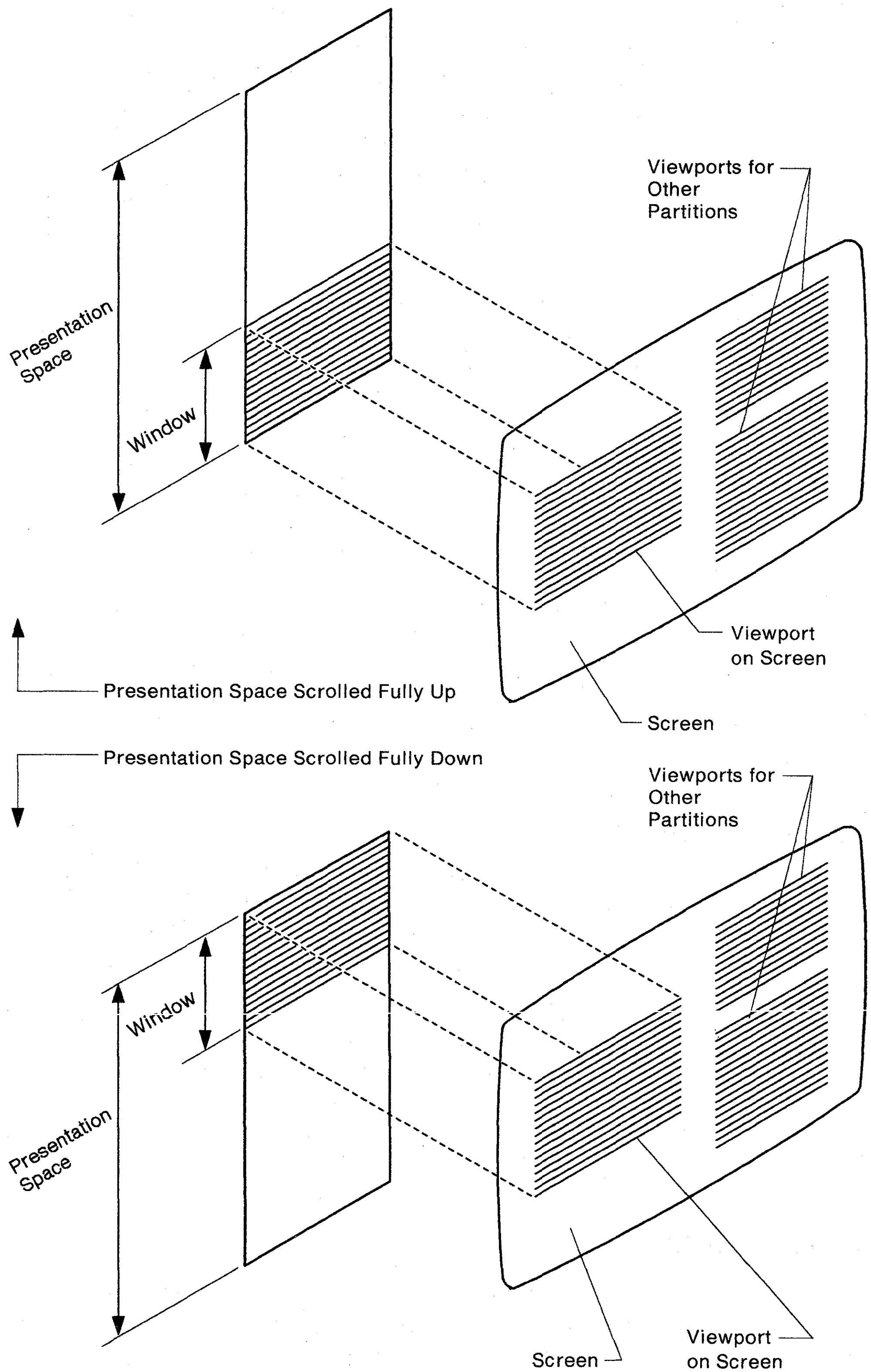


Figure 2-3. Presentation Space, Window, and Viewport (with Scrolling)

Scrolling

In a scrollable partition, the presentation space data can be viewed by:

- The operator using the keys for scrolling
- The application program.

With this function, the application program positions the window by specifying the number of rows (see "Set Window Origin" in Chapter 5) by which the top of the window is to be offset from the top of the presentation space.

The Cursor in Partitions

As on the nonpartitioned screen, the cursor can be moved by operator keystrokes or by instructions from the application program; but, always, the range of cursor movement on the screen is constrained to the bounds of a viewport. The cursor can be moved out of a given viewport only into another viewport; this can be accomplished, for example, by means of the jump-partition key or an Activate Partition structured field sent from the application program. The cursor can never appear at a screen position that is outside the bounds of a viewport. The partition associated with the viewport that contains the cursor is known as the *active partition*. The operator can enter data only into the active partition.

Associated with each partition is a current cursor position on the presentation space that determines where alphanumeric data is placed in the presentation space during operator keystroking. The cursor on the display screen is displayed at the current cursor position of the active partition's presentation space. Data entry or cursor movement causes the current cursor position of the partition to be changed.

Multiple Partitions

The physical screen can be divided into several viewports, allowing data from multiple partitions to be displayed on the same physical screen. The partition with which the operator interacts is called the *active* partition, and only one partition may be active at a time. Each partition is identified by a partition identifier (PID).

The operator may cause the screen cursor to jump from one partition to the next by pressing the designated jump-partition key. Pressing this key moves the cursor to the current cursor position of the next partition and makes that partition the active partition. If the partition ID has the value N, the next active partition is that partition with the smallest PID greater than N. If no such partition exists, the next active partition is that partition with the lowest PID.

Activate Partition

The application program may activate any partition by using the Activate Partition structured field and may deactivate any partition by activating another partition. The host application program may destroy any partition by using the Destroy Partition structured field. (These structured fields are described in Chapter 5, "Outbound Structured Fields.")

The Implicit Partition

The display is placed in implicit partition state at BIND time (SNA) or when the display is powered on (non-SNA). A single implicitly defined partition is created automatically and assigned a PID of 0 with the default screen size. In implicit partition state, the size of the implicit partition is controlled by the Erase/Write (EW) and Erase/Write Alternate (EWA) commands in the data stream.

EW redefines implicit partition 0 with the default screen size. EWA redefines implicit partition 0 with the alternate screen size. The default and the alternate size are specified in BIND SESSION.

The characteristics of the implicit partition are as follows:

- Partition parameters expressed in row/column coordinate system
- Partition size = Screen size
- Window size = Partition size
- Viewport size = Window size
- Viewport origin = Screen origin
- No scrolling permitted
- Unprotected (operator interaction allowed).

Note: For non-SNA environments, the default size is the largest 3277 size that fits on the display screen. The alternate size is implementation defined.

Explicit Partitions

The Create Partition structured field is used to replace the implicitly created partition 0 with a partition that is explicitly defined. The first Create Partition causes the implicit partition to be destroyed and the display to be placed in an explicit partitioned state.

The difference between the implicit partition, hereafter called *implicit partition 0*, and explicit partition 0 is that implicit partition 0 is assigned partition characteristics by default, whereas the application program can specify partition characteristics for explicit partition 0 by using the Create Partition structured field. In addition, implicit partition 0 operates in implicit partition state, while explicit partition 0 operates in explicit partitioned state.

No matter which partition 0 is used, commands can be sent to the partition as the first byte of the data stream or enclosed in a structured field. When data is returned from either implicit or explicit partition 0, it is transmitted without use of the Inbound 3270DS structured field.

States and State Transitions

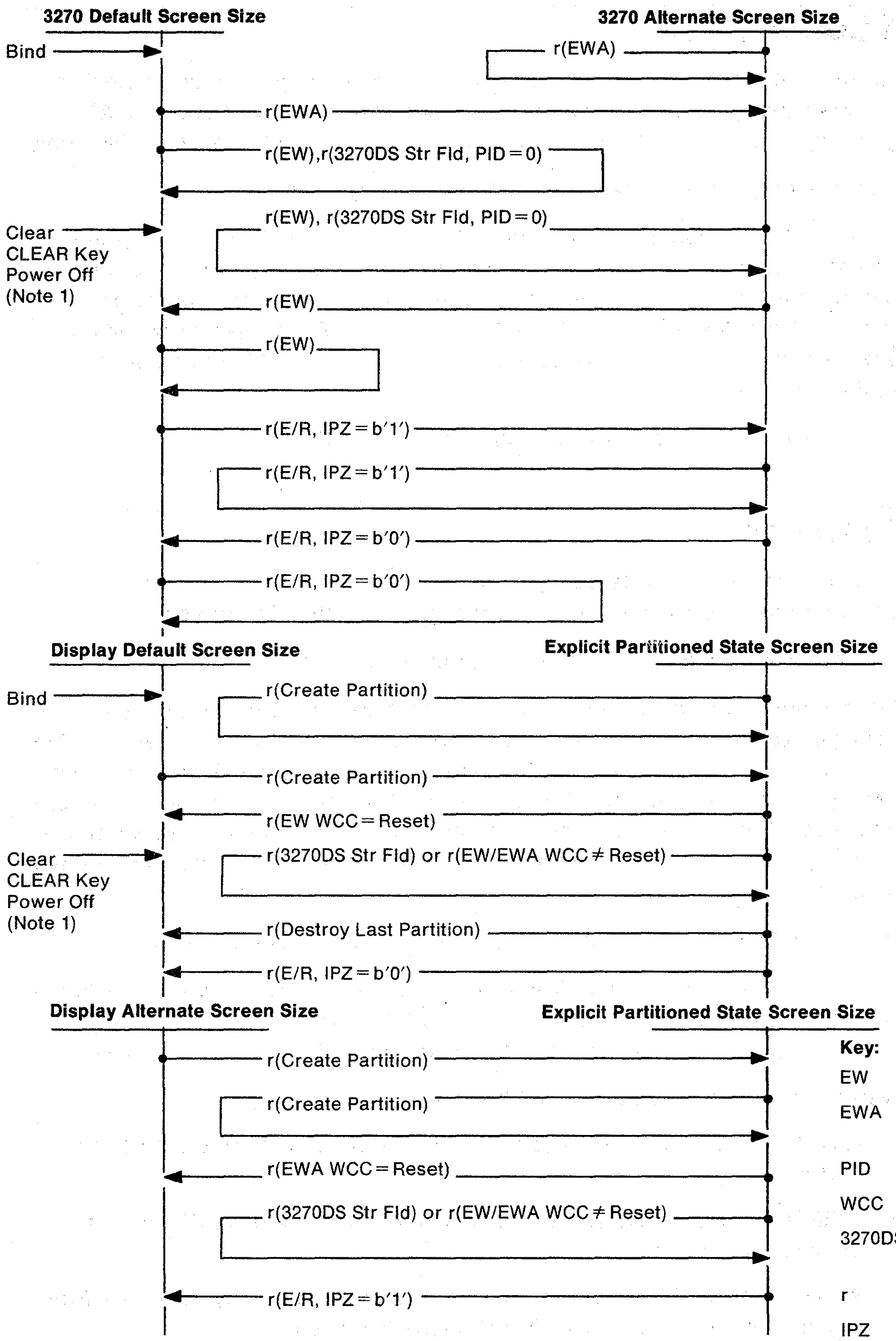
Displays using the 3270 data stream can operate in one of two states. When operating without partitions (operating with implicit partition 0), the device is said to be in implicit partition state. When partitions have been created explicitly, with the Create Partition structured field, the device is said to be in the explicit partitioned state. In each state, all the orders and commands described in this manual are valid. The distinction between the two states relates to the way in which the usable area is managed. In implicit partition state, the size of the viewport equals the screen size. In explicit partitioned state, there can be more than one viewport on the usable area. For more information, see the Create Partition structured field in Chapter 5, "Outbound Structured Fields."

The presentation space controls the value at which buffer addresses wrap. For those devices that permit scrolling, the presentation space may be larger than the viewport. For devices that do not permit scrolling, the presentation space is the same size as the viewport.

The management of presentation spaces is summarized in Figure 2-4 on page 2-8. The commands, reset function, and structured fields shown in this figure are described in detail in the manual.

When in implicit partition state, EW establishes a default screen size and EWA establishes an alternate screen size.

The functions of the EW, EWA, Write, and EAU commands on a partition with a nonzero PID are achieved by transmitting a WSF command containing an Outbound 3270DS structured field. This structured field contains the corresponding partition command; that is, it indicates whether the operation to be performed is to be an EW, EWA, Write, or EAU operation.



Key:

EW	= Erase/Write command
EWA	= Erase/Write Alternate command
PID	= Partition Identifier
WCC	= Write Control Character
3270DS Str Fld	= Outbound 3270DS structured field
r	= Receive
IPZ	= Implicit Partition Size

Figure 2-4. Management of Presentation Spaces

Notes:

1. Any local action that resets the screen, such as device return from local-test or SSCP use, may also reset the screen size to the default value.
2. If a structured field cannot be processed because of a data stream error, no state transition occurs.

Partition Wait Condition (PWAIT)

The partition-wait condition (PWAIT) is a partition-related input-inhibit condition that, when activated, prevents operator keystroking into that partition. PWAIT is set by any operator enter action except a trigger action. The following rules apply:

- PWAIT is only reset by the host by acknowledging the INOP.
- While the PWAIT condition is active for the active partition, the appropriate indicator is displayed. The operator can use the jump-partition key to jump to another partition and keystroke into that partition. The PWAIT indicator will be removed from the screen, but the PWAIT condition will be remembered, and when the original partition becomes active again the PWAIT indicator will again be displayed.
- The PWAIT condition can only be applied against INPID.

The System Lock Condition

System Lock is a partition-related input-inhibit condition that, when activated, prevents operator keystroking into that partition. The following rules apply:

- System Lock can be reset by the host or by the operator.
- While System Lock is active for the active partition, an appropriate indicator is displayed (provided a higher-priority condition, such as PWAIT, does not exist). The operator can use the jump-partition key to jump to another partition and keystroke into that partition. The System Lock indicator will be removed from the screen, but the System Lock condition will be remembered, and when the original partition becomes active again the System Lock indicator will again be displayed.
- At any one time, several partitions can have a system-lock condition.
- System Lock is activated, together with PWAIT, by any operator enter action except a trigger action. If the display is in CONTENTION state, BID or begin bracket (BB) will activate System Lock on partition 0 if partition 0 is active.
- System Lock is removed by any of the following:
  - A write with keyboard restore removes System Lock from the partition addressed by the write.
  - A RESET key pressed by the operator removes System Lock from the active partition.
  - An end bracket (EB) indicator in the RU chain removes System Lock from the inbound partition whose PID is defined by INPID.

The Terminal Wait (TWAIT) Condition

TWAIT is a terminal-related input-inhibit condition that prevents the operator from keystroking. TWAIT is activated when there is only one partition and the operator performs any enter action. Then TWAIT is a special case of the PWAIT condition, and the rules defined for PWAIT apply to TWAIT. If the display is in CONTENTION state, BID or begin bracket will activate TWAIT. TWAIT is removed by a write with keyboard restore to any partition or by any transmission that after processing leaves the display in SEND or CONTENTION state.

Chapter 3. 3270 Data Stream Commands

Introduction

As described earlier, the 3270 data stream consists of commands, orders, control characters, attributes, data, and structured fields. This chapter describes the commands and how they function in the data stream. The 3270 data stream commands and user-provided data are transmitted between the application program and the display.

The outbound data stream usually consists of write commands and a WCC followed by orders and data. If the write command is a WSF, however, no WCC byte follows this write command in the data stream. The format of the write type command is as follows:

Write Command	WCC	Orders and Data

or

WSF	Structured Field(s)

Commands

Commands are sent to a display to initiate the total or partial writing, reading, or erasing of data in a selected character buffer. Commands are sent as a command code in the first byte of a request/response (RU) unit chain (referred throughout this manual as RU chain or chain), or they may be sent in structured fields.

Commands within Structured Fields

Structured fields are used to extend the function provided by the commands. When structured fields are used, the entire Request/Response unit (RU chain) must be made up of structured fields. Therefore, certain structured fields have been defined to allow sending command functions, orders, attributes, and so on, in the same RU chain with other types of structured fields. (See Chapter 5, "Outbound Structured Fields" and Chapter 6, "Inbound Structured Fields" for a description of the structured fields.)

The Outbound 3270DS structured field provides the write-type command functions (Write, Erase/Write, Erase/Write Alternate, Erase All Unprotected), and the Read Partition structured field provides the read-type command functions (Read Buffer, Read Modified, Read Modified All).

In general, the command protocol is the same whether the function is initialized by a command code (first byte of the 3270 data stream) or by an Outbound 3270DS or Read Partition structured field. There are some differences, however, which are detailed in this chapter.

Command Codes

The command codes are not unique code points; they rely on position to resolve ambiguity. Only one command is allowed per RU chain. The command must be the first byte of the 3270 data stream.

The figure below contains codes and command abbreviations:

Table 3-1. Command codes and abbreviations

Command	Abbreviation	EBCDIC	ASCII
Write	W	X'F1'	X'31'
Erase/Write	EW	X'F5'	X'35'
Erase/Write Alternate	EWA	X'7E'	X'3D'
Read Buffer	RB	X'F2'	X'32'
Read Modified	RM	X'F6'	X'36'
Read Modified All	RMA	X'6E'	X'3E'
Erase All Unprotected	EAU	X'6F'	X'3F'
Write Structured Field	WSF	X'F3'	(Note)

Note: The use of structured fields requires that the full 8 bits of a byte be used; therefore, WSF is not supported in an ASCII environment.

The Write Control Character (WCC) Byte

The WCC is not a unique code but is identified by position; that is, it is the byte following the write-type command. If the WCC is omitted, whatever follows the write-type command is interpreted as the WCC. The data stream is normally a minimum of a W, EW, or EWA command and the WCC. If any write command (except EAU) is sent with no WCC or data, it is treated as a no operation (no-op).

Although no WCC follows the WSF command, there may be a WCC in the Outbound 3270DS structured field. When the WCC specifies an operation that does not apply to the display, the specified operation is not performed. For example, the Sound Alarm is a no operation if the display does not have an audible-alarm feature.

All WCC functions except for Reset MDT are deferred until data is written and orders are performed. See Table 3-2 on page 3-3 for a description of each WCC bit and Table 3-3 on page 3-4 for a summary of the reset actions.

When a data stream contains multiple WCCs (because they may appear in structured fields), the WCC functions are executed as follows:

Reset	Executed in each structured field as it is encountered.
Start print	Executed at the end of the RU chain, after the write operation has been completed. Only the last structured field may have a WCC that specified Start Print; otherwise, the chain is rejected (sense code X'1001' or sense code X'1005').
Sound alarm	Executed for each structured field, at the end of the operation specified for the structured field.
Keyboard restore	Executed for each structured field, at the end of the operation specified for the structured field.
Reset MDT	Executed for each structured field, prior to the writing of any data or the executing of any orders in the data stream.

Table 3-2. Write Control Character (WCC) Bit Definitions for Displays

Bit	**Explanation**
0	If the reset function is not supported, the only function of bits 0 and 1 is to make the WCC byte an EBCDIC/ASCII-translatable character. Bits 0 and 1 are set in accordance with Figure D-1. If the reset function is supported, bit 1 controls reset/no reset and bit 0 has no function. When bit 1 is used for the reset function the WCC byte is no longer always EBCDIC/ASCII-translatable; therefore the reset function cannot be supported in an ASCII environment.
1	WCC reset bit. When set to 1, resets partition characteristics to their system-defined defaults. When set to 0, the current characteristics remain unchanged (no reset operations are performed). See Note.
2 and 3	Reserved.
4	Start-printer bit. When set to 1, initiates a local-copy operation of the display surface at the completion of the write operation. When no printer is available, a negative response (0801) is returned. (See Chapter 8 for details of local-copy operation.)
5	Sound-alarm bit. When set to 1, sounds the audible alarm at the end of the operation if that device has an audible alarm.
6	Keyboard-restore bit. When set to 1, restores operation of (unlocks) the keyboard. It also resets the AID byte.
7	Reset modified data tag (MDT) bits in the field attributes. When set to 1, all MDT bits in the device's existing character buffer are reset before any data is written or orders are executed.

Table 3-3. Write Control Character (WCC) Reset Actions (for Displays)			
Reset Condition	**Partitions Not Supported**	**Partitions Supported, but Display in Implicit Partition State**	**In Explicit Partitioned State**
1. WCC following an Erase/Write or an Erase/Write Alternate command.			
a. WCC = Reset.	Execute the command; reset the inbound reply mode to field (if applicable.)	Execute the command; reset the inbound reply mode to field.	Rest the display to the implicit partition state; execute the command.
b. WCC = No reset.	Execute the command.	Execute the command.	Execute the command against explicit partition 0; if not explicit partition 0, reject the command.
2. WCC following a Write command.			
a. WCC = Reset or no reset.	Execute the command.	Execute the command.	Execute the command against explicit partition 0; if no explicit partition 0, reject the command.
3. WCC in Outbound 3270DS, and the function is Erase/Write or Erase/Write Alternate.			
a. WCC = Reset.	If the PID equals 0, execute the function and reset the inbound reply mode (if applicable). If the PID does not equal 0, reject with -RSP.	If the PID equals 0, execute the function and reset the inbound reply mode. If the PID does not equal 0, reject with -RSP.	Reset the designated (PID) partition, and execute the function against the designated partition. If the designated partition does not exist, reject with -RSP.
b. WCC = No reset.	If the PID equals 0, execute the function. If the PID does not equal 0, reject with -RSP.	If the PID equals 0, execute the function. If the PID does not equal 0, reject with -RSP.	Execute the function against the designated partition. If the designated partition does not exist, reject with -RSP.
4. WCC in Outbound 3270DS, and the function is Write.			
a. WCC = Reset or no reset.	Execute the function if the PID equals 0; otherwise, reject with -RSP.	Execute the function if the PID equals 0; otherwise, reject with -RSP.	Execute the function against the designated partition. If the designate partition does not exist, reject with -RSP.

Write Operation

The process of sending a write-type command and executing that command is called a *write operation*. Five write commands are initiated by the application program and executed by the display:

- Write (W)
- Erase/Write (EW)
- Erase/Write Alternate (EWA)
- Erase All Unprotected (EAU)
- Write Structured Field (WSF).

The W, EW, EWA commands are used by the application program to load, format, and selectively erase a character buffer or presentation space at the display. These commands can also initiate certain display operations, such as copying the contents of the display screen, restoring the keyboard, and sounding the audible alarm.

Write and Erase/Write operations are identical except that EW causes complete erasure of the character buffer before the write operation is started. Thus, EW is used to load the buffer with completely new data, whereas Write can be used to add to or modify existing buffer data. EWA is identical with EW except that EW sets and uses the default display screen size while EWA sets and uses the alternate display screen size.

Write, EW, and EWA, when sent in the first byte of the data stream, are used for write operations in partition 0. They must be encoded in a structured field if used for partitions with nonzero IDs.

The WSF must be used for any write operation to partitions with nonzero IDs. The command is followed by one or more structured fields, which are interpreted and executed by the display. The structured field identifies the specific partition by its partition identifier.

Write Command

This command writes data into specified locations of the character buffer of partition 0 without erasing or modifying data in the other locations. Data is stored in successive buffer locations until an order is encountered in the data stream that alters the buffer address, or until all the data has been stored. During the write operation, the buffer address is advanced one location as each character is stored.

This command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The buffer location where the entry of data starts depends on the starting location specified by the Set Buffer Address order that follows the WCC. If an SBA does not follow the WCC, the starting location is the buffer address where the cursor is positioned. The formatting and placement of write data and the modifying of existing buffer data are described under "Orders" in Chapter 4.

Erase/Write Command

Execution of the Erase/Write command performs the following functions:

1. If in implicit partition state, it sets the implicit partition size to the default size.
2. Erases the character buffer by writing null characters into all buffer locations.
3. Sets all the associated character attributes and extended field attributes to their default value (X'00').
4. Erases all field validation attributes.
5. Sets the current cursor position to 0. If directed to a partition, autoscroll is performed if necessary to position the window at offset (0,0).
6. If bit 1 of the WCC is set to B'1':
   a. Resets the inbound reply mode to Field.
   b. Resets to implicit partition state, if currently in explicit partitioned state. It destroys all partitions, creates implicit partition 0 with default screen size, and sets inbound PID to 0 and INOP to Read Modified.
7. Provides an acknowledgment of any outstanding read or enter if the keyboard-restore bit in the WCC is set to B'1'.
8. Provides a negative trigger reply.
9. Performs a write operation.

This command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Bit 1 in the WCC carries reset information used by Erase/Write. If no WCC is defined following an Erase/Write, the command is considered a no-operation (no-op). Therefore, no erasing or resetting occurs and any outstanding read or enter operation is not acknowledged. However, it is treated as a negative reply to a trigger field AID.

To perform the E/W function in a named partition, other than partition 0, a WSF command must be used. An Outbound 3270DS structured field will contain the PID and the partition command (Erase/Write).

Erase/Write Alternate Command

This command performs the same operation as described for the Erase/Write command, but it uses the alternate implicit partition size. If there is no alternate screen size, the EWA is treated the same as an Erase/Write command.

This command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Write Structured Field Command

The WSF command is used to send structured fields from the application program to the display. On the application-to-display flow (outbound), structured fields can be sent only with the WSF command.

Devices not supporting structured-field data streams must reject this command (sense code X'1003').

The format of a Write Structured Field data stream is:

WSF Command	Structured Field	Structured Field ...

The data stream may contain one or more structured fields. Each structured field contains a length count that enables the receiver to calculate where the current structured field ends and the next one begins. Some structured fields may have a length field that equals 0, but only when sent as the last structured field in the RU chain.

The Outbound 3270DS is an example of a structured field. It allows one of four operations to be performed within the named partition:

- Write
- Erase/Write
- Erase/Write Alternate
- Erase All Unprotected.

The Write command can either be sent as the first byte of the data stream to write to partition 0 or be enclosed in the Outbound 3270DS structured field to write to any explicitly created partition. See "Create Partition" in Chapter 5 for information on creating a partition.

Erase All Unprotected (EAU) Command

This command does the following:

- Clears all the unprotected character locations of the partition to nulls and sets any character attributes affected to their default values.
- Resets to 0 the MDT bit in the field attribute for each unprotected field.
- Unlocks the keyboard.
- Resets the AID.
- Repositions the cursor to the first character location, after the field attribute, in the first unprotected field of the partition's character buffer.

This command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

If the entire buffer is protected, buffer data is not cleared and MDT bits are not reset. However, the keyboard is unlocked, the AID is reset, and the cursor is repositioned to the first buffer address in the partition.

Attention Identification (AID)

The AID byte appears only in the inbound (display to application program) data stream and must be the first byte of the inbound data stream. The AID indicates the source or type of data that follows in the data stream. Usually, there is only one AID byte in an RU chain. The exception is an RU chain containing an Inbound 3270DS structured field that itself may contain an AID byte.

When the terminal operator initiates an enter operation, the display includes an AID byte in the input to the application program indicating the operator action. Operator actions that initiate an enter operation include:

- Pressing a program-function or program-attention key
- Pressing the ENTER, CLEAR, or Clear Partition key
- Reading a magnetic stripe with a magnetic reader
- Detecting on an attention field with the selector pen.

The possible AID bytes are shown in Table 3-4 on page 3-9. All AID bytes transmitted by the display are a result of operator actions except for:

- No AID generated
- No AID generated (printer)
- Structured field
- Read partition.

The display sends the no-AID-generated AIDs for unsolicited reads, errors, and unusual conditions. It sends the structured field AID whenever a structured field is sent inbound. The structured field may be sent as a result of some operator action, or it may be a reply to a previous application program request. The display sends the structured field AID (X'88') and the Read Partition AID when replying to a Read Partition structured field requesting a read operation. (See Table 3-4 on page 3-9.)

When data is transmitted to the application program, the most recent AID value is transmitted. The display records the most recent AID byte value. This value, initially set to *no-AID-generated*, may be set to another value by operator action. (See Table 3-4 on page 3-9.) The application program may reset the AID value to *no AID generated* by sending a write command with the keyboard-restore bit set on in the write control character.

Once the AID is set, it remains set and input is inhibited until one of the following commands is issued:

- Any write command that has the keyboard-restore bit on in the WCC
- Erase All Unprotected.

Table 3-4. Attention Identification (AID) Bytes Sent from the Display to the Application Program

AID	EBCDIC (Hex)	ASCII (Hex)	EBCDIC (Graphic)
No AID generated	60	2D	
No AID generated (printer only)	E8	59	Y
Structured field	88[1]		h
Read partition	61[2]		/
Trigger action	7F		"
Test REQ and SYS REQ	F0	30	0
PF1 key	F1	31	1
PF2 key	F2	32	2
PF3 key	F3	33	3
PF4 key	F4	34	4
PF5 key	F5	35	5
PF6 key	F6	36	6
PF7 key	F7	37	7
PF8 key	F8	38	8
PF9 key	F9	39	9
PF10 key	7A	3A	:
PF11 key	7B	23	#
PF12 key	7C	40	@
PF13 key	C1	41	A
PF14 key	C2	42	B
PF15 key	C3	43	C
PF16 key	C4	44	D
PF17 key	C5	45	E
PF18 key	C6	46	F
PF19 key	C7	47	G
PF20 key	C8	48	H
PF21 key	C9	49	I
PF22 key	4A	5B	¢
PF23 key	4B	2E	.
PF24 key	4C	3C	<
PA1 key	6C	25	%
PA2 key (CNCL)	6E	3E	>
PA3 key	6B	2C	,
CLEAR key	6D	5F	-
CLEAR PARTITION key	6A		
ENTER key	7D	27	'
Selector-pen attention	7E	3D	=
Magnetic Readers:	E6	57	W
Operator ID reader Mag Reader Number	E7	58	X

[1] When structured fields are sent inbound, the first byte of the inbound data stream is always X'88'. Some structured fields also contain an AID, so the inbound data stream that contains structured fields may have more than one AID present. For example, the Inbound 3270DS contains an AID of X'61' if it is sent inbound as a result of a read-partition structured field. It contains an AID of X'7D' if sent inbound as a result of the ENTER key.

[2] Can never be the first byte in an inbound data stream.

Read Operation

The process of sending data inbound is called a *read operation*. A read operation can be initiated by:

- The Host application sending an explicit read command.
- The Host application program sending a Read Partition structured field specifying Read Buffer, Read Modified, or Read Modified All.
- An operator action, for example, pressing the ENTER key.

A read operation sends an inbound data stream (from the terminal to the application program) with an AID byte as the first byte of the inbound data stream. The inbound data stream usually consists of an AID followed by the cursor address (2 bytes). These 3 bytes of the read data stream, the AID and cursor address, are known as the *read heading*. The inbound data stream format is:

AID	Cursor address (2 bytes)	Data

An inbound data stream may also consist of an AID (X'88') followed by structured fields.

AID	Structured Field	......	Structured Field

Read Commands

Three read commands may be sent by the application program: Read Buffer, Read Modified, and Read Modified All.

When the display receives the Read Buffer command, the entire contents of the character buffer are sent to the application program. The Read Modified and Read Modified All commands start read operations that transfer only those character buffer fields that have the MDT bit set on in the field attribute.

The Read Partition structured field may initiate a read-buffer, read-modified, or read-modified-all operation, or it may initiate a query operation.

For both implicit and explicit partition 0, the read command can be the first byte of the data stream or it can be encoded in the Read Partition structured field.

The information the display transfers to the application program in reply to Read Partition is either read data from one partition or a reply to a query. The read data is returned in the Inbound 3270DS structured field. The reply to the query is returned in one or more Query Reply structured fields.

The contents of an inbound data stream depend on the inbound reply mode. The inbound reply mode is set by the application program's sending a Set Reply Mode structured field. The inbound reply modes that may be set are Field, Extended Field, and Character; the default is Field mode:

- Field mode supports inbound data that contains field attributes. Start Field and Set Buffer Address orders, field attributes, characters, and graphic escape code X'08' may be included in the inbound Field Mode transmissions.
- Extended Field mode supports field attributes and extended field attributes. Start Field Extended and Set Buffer Address orders, field attributes, extended field attribute, characters, and the graphic escape code X'08' may be included in the inbound transmission.
- Character mode supports the field attributes, extended field attributes, and character attributes. Start Field Extended, Set Buffer Address, and Set Attribute orders, field attributes, extended field attributes, character attributes, and the graphic escape code (X'08') may be included in the inbound transmission.

Read Buffer Command

Execution of the Read Buffer command causes all data in the addressed display buffer, from the buffer location at which reading starts through the last buffer location, to be transmitted to the host. For displays, the transfer of data begins from buffer address 0.

This command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Note: Because of the large quantities of data that are transferred and processed, using the Read Buffer command will significantly increase the 3270 subsystem and teleprocessing network response times. The Read Buffer command is intended primarily for diagnostic purposes.

Read Buffer Field Mode

No extended field or character attribute information is returned when the inbound data stream is in this form. Only Start Field (SF) orders occur in the data stream. The format of the Read Buffer data stream is:

AID	Cursor Address (2 bytes)	SF Order	Attribute	Data

Read Buffer Extended Field Mode

In this mode, SFE orders are generated in place of SF orders. No character attributes are transmitted inbound in this form. The form of the Read Buffer Extended Field mode is:

AID	Cursor Address	SFE Order	No. of Attribute Type Value Pairs	Attribute Type	Attribute Value	...... Data

Attributes with default values are not transmitted inbound in the SFE order.

Read Buffer Character Mode

In this form, the inbound data stream is as defined for Extended Field mode above, except that Set Attribute (SA) orders may be inserted. SA orders will be generated only when the attribute value of an attribute type listed in the Set Reply Mode changes.

All attributes are assumed to have their default values at the beginning of the inbound transmission. Therefore, the first SA order generated will be for the first attribute type not equal to its default value. However, SA orders will not be generated for attribute types not listed in the Set Reply Mode structured field. The format of the Character mode data stream is:

AID	Cursor Address	SA	Attr Type	Attr Value	Data	SFE	No. of Attr Type-Value Pairs	Attr Type	Attr Value	...	Data

Read Modified Command

Read Modified initiates one of two operations, as determined by operator actions at the display station: (1) read modified or (2) short read.

This command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Read-Modified Operation

During a read-modified operation, if an AID other than selector pen attention, PA key, or CLEAR key is generated, all fields that have been modified by keyboard, selector pen, or magnetic-reader activity are transferred to the application program. A major feature of the read-modified operation is null suppression. Only non-null character data and corresponding character attribute data (in Character mode) are transmitted. All null-character data and all extended attributes for null-character data are suppressed.

If a space or null selector-pen-attention AID is generated, fields are not transferred to main storage during the read-modified operation. Instead, when a set MDT bit is found (indicating selector pen and/or keyboard activity), only the read heading, the SBA order code, and the attribute address +1 are transferred.

Note that if fields are modified by the keyboard, but completion of the modification is signaled by a selector-pen-attention operation on other than ampersand character-designator fields, a resulting read-modified operation will read only the address of the modified fields, not the modified data. A Read Modified command can be used to obtain both the address of, and the data in, each field that has the MDT bit set to 1.

The search for modified fields ends when the last buffer location is checked.

If the buffer is formatted (contains fields) but none of the fields have been modified, the read data stream consists of the 3-byte read heading only.

If the buffer is unformatted (contains no fields), the read data stream consists of the 3-byte read heading followed by all alphanumeric data in the buffer (nulls are suppressed), even when part or all of the data has not been modified. Since an unformatted buffer contains no attribute bytes, no SBA codes with associated addresses or address characters are included in the data stream, and the modification of data cannot be determined. Data transfer starts at address 0 and continues to the end of the buffer. At the end of the operation, the buffer address is set to 0.

Short-Read Operation

The Read Modified command causes a short read operation if the CLEAR, CNCL, or a PA key has been pressed at the selected device. During the short-read operation, only an AID byte is transferred to the application program. This AID byte identifies the key that was pressed.

Read Modified Field Mode

No character attribute information is returned in this form of the inbound data stream. Only SBA orders occur in the data stream. The format of the Read Modified Field Mode data stream is:

AID	Cursor Address (2 bytes)	SBA	Attribute Address + 1	Text

Read Modified Extended Field Mode

This inbound data stream is exactly like the Read Modified Field Mode data stream above.

Read Modified Character Mode

This data stream is identical with that defined for the Read Modified Field Mode above except that an SA order is inserted into the data stream as required to define the change in the attribute value for the graphic characters transmitted. SA orders are generated only when the attribute value of an attribute type listed in the Set Reply Mode changes. All attributes are assumed to have their default values at the beginning of the inbound transmission. Therefore, the first SA generated will be for the first attribute type not equal to its default. However, SA orders are not generated for attribute types not listed in the Set Reply Mode structured field. The format of this data stream is:

AID	Cursor Address (2 bytes)	SA	Attr Type	Attr Value	Data	SBA	Attr Address + 1	SA	Attr Type	Attr Value	Data

Read Modified All Command

This command operates exactly like the Read Modified command except that it will always cause both the addresses and the data from all modified fields to be sent to the application program independent of the AID code. This includes those AID codes that would normally result in a short read, as well as for trigger action (X'7F'), Clear Partition (X'6A'), and Read Partition (X'61').

This command is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Read Modified All Field Mode

This command operates exactly like the Read Modified command.

Read Modified All Extended Field Mode

This command operates exactly like the Read Modified command.

Read Modified All Character Mode

This command operates exactly like the Read Modified command.

Read Commands in Structured Fields

For an application program to initiate a read operation from a partition with a nonzero PID, the Read Partition structured field must be used. The Read Partition structured field may also be used in addition to the RB, RM, and RMA commands to initiate a read from implicit or explicit partition 0. The Read Partition structured field allows the application program to specify an RB, RM, or RMA operation. The inbound reply to a Read Partition structured field (directed to a nonzero partition) specifying an RB, RM, or RMA operation will be sent in an Inbound 3270DS structured field.

Read Operations from Partitions

In addition to an operator *enter* action or an application-program-initiated (unsolicited) read, a read operation can result from an application program retry. During the processing of read and enter operations, the display goes into retry state from the time that the data is transferred until the application program acknowledges that the transfer of data was successful. (The retry state is discussed under "Read States.")

The addition of partitions to the 3270 data stream made it necessary to provide a formalized host application retry/acknowledgment mechanism for certain read operations. These read operations, called *inbound operations*, are defined as read operations caused by either:

- An operator enter action (for example, the Enter key)
- A host application issuing a partition command (Read Partition structured field = RM, RMA, RB, Query, or Query List).

INOP here refers to the *current inbound operation* and INPID refers to the *partition associated with the current INOP.*

Note that the INPID can be either implicit or explicit partition 0.

In an SNA environment, an inbound operation causes the display to go into a retry state. Once in the retry state, another inbound operation cannot be initiated until INOP is acknowledged by the host application. Once the acknowledgment is received, the display reverts to the "normal" state and ends the current inbound operation (INOP).

What constitutes an acknowledgment of an inbound operation depends on the inbound operation and the environment (SNA or non-SNA). The acknowledgment requirements are described in detail later in this section.

When the display is in a retry state, the host application can cause a repeat of the INOP by sending a RM command (**not** a partition RM command).

The RM command is the *only* mechanism for a complete retry of the INOP. When in a retry state, an RMA or RB will result in executing the RMA or RB on the INPID. That is, although the use of INPID is repeated, the inbound operation performed is *not* necessarily the same as INOP.

Read operations that are *not* considered to be inbound operations are those operations initiated by an RM, RMA or RB command when the display is in the normal read state. The RM, RMA or RB is executed against either implicit or explicit partition 0 and does *not* cause a transition from the normal state. There is no retry/acknowledgment associated with these commands when the display is in the normal state. INOP and INPID are not relevant to this situation.

Read Operations (SNA)

For the non-SNA operation, see Chapter 9, "Binary Synchronous Communications (BSC) Environment" and Chapter 10, "Non-SNA Environment (Locally Attached Devices—3272 Version)."

Operator Enter Actions

An operator enter action causes an AID and, optionally, data to be sent to the application program. For example, a PA key or the CLEAR key causes only an AID to be sent, while the ENTER key causes an AID and data from all fields with the MDT bit set to be sent.

An enter action:

1. Sets INOP and INPID so that operation and partition identifier are known if a retry is needed
2. Raises the enter-inhibit condition so that the operator cannot generate an AID from any partition
3. Raises the PWAIT and system-lock conditions if the enter action is not a trigger action (AID = X'7F')
4. Places the device in retry enter state.

Application-Initiated Reads

An application-initiated read is one that is unsolicited by the device. It may be:

- A Read Partition structured field.
- A Read Buffer, Read Modified, or Read Modified All command received for implicit or explicit partition 0.

A Read Partition Structured Field

A Read Partition structured field:

1. Sets INPID if the operation is RB, RM, or RMA
2. Sets INOP to RM, RMA, or RB
3. Raises the enter-inhibit condition
4. Sends the data to the application program and places the device in retry-read state.

Explicit partition 0 may be read with a read command and does not require a Read Partition structured field operation. The Read Partition structured field identifies a partition through its partition identifier (PID). The partition is assigned a PID when it is created by the Create Partition structured field.

The display's reply (except from PID 0) is an inbound data stream containing one or more structured fields of the form:

AID = X'88'	Structured Fields

where the inbound control information is encoded into structured fields as defined in Chapter 5, "Outbound Structured Fields" and Chapter 6, "Inbound Structured Fields."

The display begins the session with no partitions explicitly defined or, thought of another way, begins with the entire display surface as one partition. This partition, called *implicit partition 0*, occurs in the following situations:

- After session activation but before the first Create Partition structured field
- After all partitions created by Create Partition are deleted
- After the CLEAR key is pressed
- When an Erase/Write command, Erase/Write Alternate command or an Erase/Reset structured field is received as the first byte of an outbound data stream and the WCC Reset bit is on.

Since some application programs may not send the Create Partition, those sessions use implicit partition 0 for the duration of their session.

Note: The Create Partition structured field can define explicit partition 0 by setting PID = X'00'. The difference between implicit partition 0 and explicit partition 0 is that implicit partition 0 is assigned partition characteristics by default, while the application program can specify partition characteristics for explicit partition 0 by using the Create Partition structured field.

No matter which partition 0 is used, commands can be sent to the partition as the first byte of the data stream or enclosed in a structured field.

In the event that an error causes the application program to retry the Read Partition, the display must remember the read operation that was performed and the partition on which the operation was performed.

INOP records the type of operation (read buffer, read modified, read modified all, query) that is being performed so that it can be repeated if a retry of a Read Partition is required.

INPID defines the partition from which the data will be retransmitted.

To ensure data integrity for retry of operator enter actions, an input-inhibit condition, called *partition wait* (PWAIT), prohibits the operator from working in the partition until the application program acknowledges that the transfer of data was successful.

To ensure that the operator cannot change the value of INPID or INOP until the application program acknowledges that the transfer of data was successful, an input-inhibit condition, called *enter inhibit*, prohibits the operator from generating an AID from any partition.

The inbound data stream contains an Inbound 3270DS structured field with an AID byte of X'61' to identify that the data is a result of Read Partition.

X'88'	Length	X'80'	PID	X'61'	Cursor Address	Data

Notes:

1. The X'80' identifies the structured field as an Inbound 3270DS.
2. From the AID X'61' to the end of the structured field, the data stream is the same as that sent by a corresponding RB, RM, or RMA command in implicit partition state.

An RB, RM, or RMA command received for implicit or explicit partition 0:

1. Transmits data from partition zero.
2. Leaves the device in its current read state.

A Host Retry

When the device is in any of the retry states, the read modified command performs a retry of the inbound operation on INPID.

The RB or RMA commands perform their respective read operations in the inbound partition. The RB and RMA commands do not necessarily retry the inbound operation.

Read States

The device has one of two states with respect to read operations: normal read or retry:

- Normal-read state. Prior to initiation of a read operation, the device is in normal-read state. An operator enter action or a Read Partition structured field causes the device to transmit the data and to go into a retry state.
- Retry state. There are two forms of retry state.
  - Retry enter, when the entered data has been transmitted to the application program.
  - Retry read, when the data read has been transmitted to the application program.

  While in retry state, the last inbound operation (INOP) can be retried by using a Read Modified command. The application program must acknowledge receipt of the read data. This acknowledgment causes the device to revert from a retry state to the normal-read state. There is no Retry state for any read command sent as the first byte of the data stream.

Normal Read State

In normal-read state, either an operator enter action or a host-initiated read is accepted and processed.

Transition Event	Next State	Operations
Enter action	Retry Enter	Transmit Read Modified data stream.
Any read command[1]	Normal Read[2]	Transmit data stream defined by command.
Read Partition[3]	Retry Read	Transmit data stream defined by Read Partition.

[1] Partition 0, whether implicit or explicit, must exist or the read command sent as the first byte of the data stream is rejected.
[2] There is no retry state.
[3] The addressed partition must exist.

Retry Enter State

In retry-enter state, an operator enter action or a Read Partition is not processed and is rejected. When the inbound transmission is acknowledged, the device returns to normal-read state. In retry-enter state, a read modified command sent as the first byte of the data stream is interpreted as a retry of INOP.

Transition Event	Next State	Operations
Read Modified command	Retry Enter	Retransmit data from partition identified by INPID.
Host acknowledge	Normal Read	Terminate read operation.

Retry Read State

In retry-read state, an operator enter action or a Read Partition is not processed and is rejected or queued. In this state, a read command sent as the first byte of a data stream is interpreted as a retry.

When the inbound transmission is acknowledged, the device returns to normal-read state.

Transition Event	Next State	Operations
Read Modified command	Retry Read	Retransmit data from command INPID.
Host acknowledge	Normal Read	Terminate read operation.

Read-State Transitions

The read-state transitions are summarized in Figure 3-1:

Read States	**Normal Read**	**Retry**	
		Enter	**Read**
Enter Action	2	R	R
Any Read Command	1	G	G
Read Partition	3	R	R
Host Acknowledge	1	1	—

Key:

R Reject, no state transition
G Retry, no state transition
— No state change, no action
1 Normal-read state
2 Retry-enter state
3 Retry-read state

Note: Read Partition is rejected (sense code X'0871') when the device is in Retry state.

Figure 3-1. Read-State Transitions

Inbound Operation (INOP)

The display records the inbound operation (INOP) so that it knows the operation to perform when it transmits data inbound. The display also uses INOP to know which operation to retry when a host-initiated retry is received.

INOP is set as follows:

- With the exception of a Read Partition structured field directed to partition 0, whenever INPID is set to 0, INOP is set to Read Modified.
- An operator enter action, including a trigger action, sets INOP to Read Modified.
- A Read Partition structured field sets INOP to the specified operation, namely, RB, RM, RMA, Query.
- Acknowledgment of an inbound transmission sets INOP to Read Modified.

Inbound Partition Identifier (INPID)

The display records the inbound partition identifier (INPID) so that it determines which partition to use when it transmits data inbound. The display also uses INPID to know which partition to use when an application program – initiated retry is received.

INPID is set as follows:

- When an implicit partition is created, INPID is set to 0.
- An operator enter action, including a trigger action, sets INPID to the PID of the active partition.
- A Read Partition structured field causes INPID to be set to the PID value specified in the Read Partition structured field unless the PID value is X'FF' (Query operation). If the Read Partition structured field PID equals X'FF', INPID is left unchanged.
- Destruction of the inbound partition sets INPID to 0.

Enter Actions

If the enter action is other than a trigger action, the following conditions are set when the device (logical terminal) goes into the Retry Enter state:

- Either the TWAIT or PWAIT input inhibit condition
- The System Lock input inhibit condition
- The Enter Inhibit condition.

Input Inhibit Conditions

1. TWAIT/PWAIT

   TWAIT will be used when in the implicit partition state or when there is only one explicit partition. PWAIT is used when there is more than one partition. PWAIT applies only to INPID. Only one PWAIT condition may exist for a device (logical terminal). The indicator for TWAIT is "X Clock"; for PWAIT it is "X <-Clock->." The arrows of the PWAIT indicator indicate that the operator can "go elsewhere," that is, jump to another partition. When the TWAIT condition exists, the Jump key does nothing; it is a No-op. Both TWAIT and PWAIT are input inhibit conditions; no keystroking is allowed. TWAIT or PWAIT is set for the duration of the Retry Enter state. TWAIT and PWAIT *cannot* be cleared by the operator (for example, by the RESET Key).

   Note: Independent of any enter action, the TWAIT condition is set whenever the device (logical terminal) is in the SNA Contention state and receives a Bid or begin brackets (BB). When set as a result of a Bid or BB, TWAIT is cleared by any of the following:

   - A partition Write, EW, or EWA command (with keyboard restore) to any existing partition
   - A partition EAU command to any existing partition
   - A Write, EW, or EWA command, with keyboard restore (implicit or explicit partition 0 must exist)
   - An EAU command (implicit or explicit partition 0 must exist)

- The device (logical terminal) receives a transmission that leaves it in the Send or Contention state (for example, CD or EB).

2. System Lock

   System Lock, like TWAIT and PWAIT, is an input inhibit condition. However, unlike TWAIT and PWAIT, System Lock can be cleared by the RESET key as well as by the host.

   The System Lock is raised for any enter action that raises TWAIT or PWAIT. However, the System Lock condition is overridden as long as a TWAIT or PWAIT condition exists. The System Lock indicator ("X System") will not be displayed while TWAIT or PWAIT exists.

   With one exception, System Lock is cleared by any Read Acknowledgment that changes the Retry Enter state back to the Normal Read state. The exception is a transmission in which the only valid Read Acknowledgment is (SNA) Change Direction (CD). The CD causes a transition to the Normal Read state and clears TWAIT or PWAIT. The System Lock condition remains set, however, and the "X System" indicator is displayed. The operator is forced to use the RESET key to clear the System Lock condition before resuming keystroking. This provides a means for the host application to attract the attention of the operator, for example, a host message.

Enter Inhibit Condition

Any enter action, *including a trigger action*, will set the Enter Inhibit condition for the device (logical terminal). The Enter Inhibit condition allows keystroking but does not allow an enter action.

The Enter Inhibit condition has significance only for the partitions other than INPID. This is because the input inhibit conditions associated with INPID (for example, TWAIT, PWAIT or System Lock) override the Enter Inhibit condition.

The operator may jump from INPID to one of the other partitions and keystroke into that partition as long as no attempt is made to initiate an enter action. There is no indicator associated with the Enter Inhibit condition. However, if an attempt is made to initiate an enter action, it will be inhibited and the appropriate indicator will be activated.

The Enter Inhibit remains up for the duration of the Retry Enter state.

Processing of Enter Actions

An operator enter action is processed as follows:

1. If the device has an enter-inhibit condition, the enter action is rejected.
2. If the display is in receive (RCV) state, the enter action is rejected.
3. If an input-inhibit condition exists, the action is ignored.
4. If none of the preceding conditions pertains, steps 5 through 11 are performed.
5. Enter Inhibit is activated for the device.
6. INPID is set to the PID of the active partition. For the selector pen, the INPID is set to the PID of the partition containing the detected field.
7. INOP is set to Read Modified.

8. If the enter action is anything other than a trigger action, then PWAIT and SYSTEM LOCK are activated for the inbound partition.
9. The device is placed in retry-enter state.
10. The data is transmitted inbound.
11. The display is placed in RCV state.

Processing of Read Commands (Alphanumeric)

A read command (RB, RM, or RMA as the first byte of the data stream) is processed as follows:

1. If any of the following conditions occurs, the read command is rejected.
   a. The display is not in RCV or contention (CONT) state.
   b. The RU does not specify CD.
   c. The RU specifies EB.

   Otherwise step 2 or 3 is performed.
2. If the device is in normal-read state, the command performs the read. Data from partition 0 is transmitted inbound as defined by:
   a. The command (RB, RM, RMA)
   b. The AID (RM)
   c. The inbound reply mode of partition 0.

   If partition 0 does not exist, the command is rejected. The device remains in normal-read state.
3. If the device is in retry state, then the command performs a retry.
   - Data is transmitted inbound according to the following definition:
     a. If the command is Read Modified, and INOP specifies Query or Query List, then the appropriate query replies are transmitted inbound.
     b. If the command is Read Modified, and INOP specifies RB, RM, or RMA, then data is transmitted from the inbound partition (INPID) as defined by:
        1) The INOP (RB, RM, or RMA)
        2) The AID (only when INOP=RM)
        3) The inbound reply mode of the inbound partition.
     c. If the command is RB or RMA, and INOP specifies Query or Query List, then the RB or RMA is performed on partition 0 if it exists. Otherwise, the command is rejected.
        - The enter inhibit condition remains in effect.
        - The Read State does not change.
     d. If the command is RB or RMA, and INOP specifies RM, RMA, or RB, then data is transmitted inbound from the inbound partition (INPID) as defined by:
        1) The command RB
        2) The inbound reply mode of the inbound partition.

Read Acknowledgment - SNA

An inbound operation (defined as a read operation caused by a Read Partition structured field that equals RM, RMA, RB, or Query List, or by an operator enter action) is acknowledged by any transmission that after execution *leaves* the display in a Send or Contention state. Note that although the transmission may put the display in the Send state, if the transmission also causes the display to send input, it is not an acknowledgment because after executing the transmission the display is not left in a Send or Contention state.

In *addition* to the above, there are other host application actions that constitute a Read Acknowledgement or ACK.

1. When the Inbound Operation is a Query or Query List.

   In general, a Query or Query List operation is acknowledged by any outbound transmission *except* one with a read command.

   Thus, the following will acknowledge a Query or Query List:

   - A WSF command with or without following structured fields. The transmission is an acknowledgment regardless of an error being detected in the accompanying structured fields as long as the WSF is accepted.

   - An EW, EWA, Write, or EAU command with or without a WCC or data. If data is present and an error is detected in the data, the transmission is *not* an acknowledgment.

2. When the Inbound Operation is an Operator Enter, or a RM, RMA, or RB Partition Command.

   In general, a RM, RMA, or RB Partition Command or an enter operation is acknowledged by either writing to the inbound partition (the partition associated with the inbound operation) with a transmission that restores the keyboard or destroying the inbound partition.

   Thus, any of the following will constitute a Read Acknowledgment (assuming the acknowledging function is supported):

   - If the inbound partition is implicit or explicit partition 0 either of the following:

     - An EW, EWA, or Write command with WCC = Keyboard Restore (Note 1 at the end of this section).
     - An EAU command.

   - If in the explicit partition state, an EW or EWA command with the WCC = reset (Note 1 at the end of this section).

   - A WSF command followed by an Outbound 3270DS structured field to the inbound partition with either of the following (Note 2 at the end of this section):

     - An EW, EWA or Write partition command with WCC = Keyboard Restore.
     - An EAU partition command.

   - A WSF command followed by a Destroy Partition structured field to the inbound partition including explicit and implicit partition 0 (Note 2).

   - A WSF command followed by a Create Partition to the inbound partition (Note 2).

- If in implicit partition state, a WSF command followed by a Create Partition structured field (Note 2).
- A WSF command followed by an Erase/Reset structured field. This applies to both implicit and explicit partition state (Note 2).

Notes:

1. If data follows the WCC and an error is detected in the data, the transmission is not a Read Acknowledgment.
2. If there is a detected error prior to or within the structured field providing the Read Acknowledgment, the transmission is not a Read Acknowledgment.

Chapter 4. 3270 Data Stream Orders and Attributes

Orders

Orders are included in the inbound and outbound data streams to provide additional control function. Unlike commands, a number of orders can occur within a chain. Orders have unique 1-byte codes and can occur in any position in the data stream. Many orders permit additional information to follow the order code, for example, buffer addresses or attributes. These longer sequences, called *control sequences*, must be contained within the RU chain. If a control sequence attempts to span a chain, it is ended on the chain boundary.

In the inbound data stream, only the following orders are permitted:

- Start Field (SF)
- Start Field Extended (SFE)
- Set Buffer Address (SBA)
- Set Attribute (SA)
- Graphic Escape (GE).

All orders can be included in the write data stream, either alone or intermixed with display or print data. Orders can be either buffer control or format control orders.

Buffer control orders are executed by the display as they are received in the write data stream. They are not stored in the character buffer. They are provided to: (1) position, define, and format data being written into the character buffer, (2) erase selected unprotected data in the character buffer, and (3) reposition the cursor.

All order codes have an EBCDIC value in the range of hexadecimal 00 (X'00') through hexadecimal 3F (X'3F'). Order codes with values in this range but not defined in this chapter are rejected.

The order codes are as follows:

Order	EBCDIC	ASCII
Start Field (SF)	X'1D'	X'1D'
Start Field Extended (SFE)	X'29'	Note
Set Buffer Address (SBA)	X'11'	X'11'
Set Attribute (SA)	X'28'	Note
Modify Field (MF)	X'2C'	Note
Insert Cursor (IC)	X'13'	X'13'
Program Tab (PT)	X'05'	X'09'
Repeat to Address (RA)	X'3C'	X'14'
Erase Unprotected to Address (EUA)	X'12'	X'12'
Graphic Escape (GE)	X'08'	Note

Note: The use of these orders requires using the full 8 bits in the associated parameter bytes; therefore, these orders are not supported in ASCII.

Start Field (SF)

This order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The SF order indicates the start of a field. In the read data stream, the display automatically inserts the SF order immediately before each field attribute when it responds to a Read Buffer command, enabling the application program to identify the field attributes.

In a write data stream, this order identifies to the display that the next byte is a field attribute. (The field attribute is described under the heading "Attributes.") The display then stores the field attribute at the current buffer address and increments the buffer address by 1.

The byte following the SF order in the write data stream is always treated as a field attribute.

If the display receives an SF order, it sets the associated extended field attribute to its default value.

Start Field Extended (SFE)

This order is also used to indicate the start of a field. However, the SFE control sequence contains information on the field's properties that are described in the extended field attribute. The SFE order has the following format:

X'29'	Number of Attribute Type-Value Pairs	Attribute Type	Attribute Value

Unlike the SF order, the field attribute does not immediately follow the order. The field attribute is included as a type-value pair in the form:

Type	Value
X'C0'	Field Attribute

Other field properties—specifically those contained in the extended field attribute—may be included in the SFE control sequence. They are also included as type-value pairs. The number of type-value pairs is indicated in the byte following the SFE order. Valid type-value pairs for describing extended field attributes are shown under "Attribute Values and Selection Rules."

Each attribute type has a specified default value. When the SFE order is used, all unspecified attribute types are assigned their default value. For example:

SFE	2	3270	Value	Color	Red

results in a field with red characters that have normal highlighting and the default character set.

If SFE is sent with no type-value pairs (zero value for number of pairs), defaults are set.

Attribute types and values that are unknown or cannot be maintained and returned inbound by an implementation will be rejected.

All attribute types and values are checked for validity. If the same attribute type-value pair appears more than once, the last specification for a repeated attribute type takes effect.

Set Buffer Address (SBA)

This order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The SBA order specifies a new character buffer address from which operations are to start or continue. The buffer address is relative to the origin of the buffer of the partition to which the order and data are directed. The first byte of the control sequence is the SBA order; the last 2 bytes contain the 2-character buffer address:

X'11'	Buffer Address (2 bytes)

In a write data stream, Set Buffer Address orders can be used to write data into various areas of the buffer. An SBA order can also precede another order in the data stream to specify (1) the starting address for a PT, RA, or EUA order, (2) the address at which a field attribute is to be stored by an SFE or MF order, or (3) the address at which the cursor is to be repositioned by an IC order.

If the SBA order specifies an invalid address (out of range), the write operation is terminated immediately (with a negative response).

In a read data stream sent in response to a Read Modified or Read/Modified All command, SBA orders replace field attributes. (SBA orders are not used in response to a Read Buffer command.)

When the display receives a Read Modified or Read Modified All command, it searches the buffer for field attributes with the MDT bit on. When found, the display inserts into the data stream an SBA order followed by the buffer address of the field attribute +1.

In the Create Partition structured field, the partition may be defined to operate with either 16-bit addressing or 12- or 14-bit addressing. When 16-bit address mode is specified for a partition, outbound buffer addresses are interpreted as 16-bit binary (all 8 bits of each byte are used for the address), and inbound addresses are generated as 16-bit binary. If the addresses are not in the proper form, the structured field is rejected.

When 12- or 14-bit address mode is specified for the partition (implicit partition 0 is always set to 12/14), bits 0 and 1 of the first address byte following an SBA are flag bits and have the following significance:

Setting	**Meaning**
B'00'	14-bit binary address follows
B'01'	12-bit coded address follows
B'10'	Reserved
B'11'	12-bit coded address follows

When the flag bits are 00, the next 14 bits (the remainder of this byte and all 8 bits of the next byte) contain a buffer address in binary form. No address translation is necessary.

If the flag bits are 01 or 11, the next 14 bits are to be interpreted as a 2-character address (6 bits in each byte). The 6 low-order bits of each byte are joined to provide a 12-bit address. The address specifies the buffer position, not the line and column position on the display surface. For example, on a 480-character display, the buffer addresses are from 0 to 479. To specify a 12-bit buffer address of 160 (binary 000010100000), bits 2-7 of the first byte are set to 000010. Bits 2-7 of the second byte are set to 100000.

The flag bit setting of 10 is reserved. Receipt of a 12/14-bit buffer address beginning with the flag bits 10 causes the data stream to be rejected. (See Figure A-1.)

If 12/14-bit addressing mode is specified in Create Partition, the format of buffer addresses transmitted inbound depends on the size of the partition being read. If the partition is greater than 4096 characters, then all addresses from that partition are in 14-bit form. If the partition is less than 4096 characters, then all addresses are generated in 12-bit form.

When 16-bit addressing is specified, the partition will always generate 16-bit addresses in the inbound data stream.

Set Attribute (SA)

The SA order is used to specify a character's attribute type and its value so that subsequently interpreted characters in the data stream apply the character properties defined by the type/value pair. The format of the SA control sequence is:

X'28'	Attribute Type	Attribute Value

An SA order alters the set of character attribute type-value pairs to be applied to all subsequent characters until:

- A new SA order changes it.
- Another write-type command is sent.
- CLEAR key is pressed.
- Power at the display is switched off.

These four actions all return the established set of character attribute type-value pairs to their default value.

The attribute type X'00' is always supported by the SA order. All other attribute types are determined by the function sets being supported (see Figure A-1). All type-value pairs are defined under "Attribute Values and Selection Rules."

An Erase/Write or Erase/Write Alternate command resets the specified portion of the character buffer to nulls, including any field attributes, and resets any extended field attributes and character attributes associated with the nulled characters to their default values. Thus, unless a field attribute overrides the default, any subsequently interpreted characters are displayed by use of these defaults.

The set of type-value pairs applied during character processing is a composite, by attribute type, of the last value specified in previously encountered SA orders. For example:

	"Current" Character Attribute Characteristics		
	Highlight	**Color**	**Character Set**
Erase/Write Command	Default	Default	Default
Set Attribute <color> <red>	Default	Red	Default
Set Attribute <char.set> <X'F3'>	Default	Red	X'F3'
Set Attribute <color> <blue>	Default	Blue	X'F3'
Set Attribute <highlight> <blink>	Blink	Blue	X'F3'
⋮	⋮	⋮	⋮
Set Attribute <X'00'> <X'00'>	Default	Default	Default

Inbound data streams will be generated with the assumption that the character attributes of the current set are all set to their default values. Set Attribute orders will be generated as required to indicate changes. Set Attribute (SA) will be generated only when a character attribute value changes, and only for those character attributes that change and are specified in the inbound reply mode.

Attribute types and values that are unknown or that cannot be maintained and returned inbound by an implementation will be rejected.

Modify Field (MF)

The MF order begins a sequence that updates field and extended field attributes at the current buffer address. After the attributes have been updated, the current buffer address is incremented by 1.

The MF control sequence has the following format:

X'2C'	Number of Attribute Type/Value Pairs	Attribute Type	Attribute Value

Attribute types not specified remain unchanged. For example, if a field is defined at buffer location xx with the following properties:

- Protected
- Alphanumeric
- Normal intensity
- Nonselectable
- Nonloadable character set
- Red
- No highlight
- No validation.

Then, interpretation of the order sequence:

SBAxx	MF	2	Color	Blue	Highlight	Blink

leaves the field with the following properties:

- Protected
- Alphanumeric
- Normal intensity
- Nonselectable
- Nonloadable character set
- Blue
- Blink
- No validation.

If the current buffer address does not contain a field attribute, the MF order is rejected.

Note that if no attributes are specified (the number of type-value pairs field is 0) then the MF order checks whether there is a field attribute at the current buffer address. If so, the current buffer address is incremented by 1, and no change is made to the field's properties.

If the same attribute type appears more than once, all attribute types and values are checked for validity, but only the last value specified for a repeated attribute type takes effect.

Invalid attribute types are rejected. Attribute values that are unknown or cannot be maintained and returned inbound by an implementation must be rejected.

Insert Cursor (IC)

This order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The IC order repositions the cursor to the location specified by the current buffer address. Execution of this order does not change the current buffer address. For example, if IC is issued when the current buffer address is 160 and the cursor is at 80, the cursor is removed from 80 and inserted at 160. The current buffer address at the end of this operation remains 160.

If the current buffer address places the cursor outside the window, the IC order causes automatic scrolling. The position of the cursor after automatic scrolling completes is described in "Scrolling" of Chapter 2, "Partitions."

Program Tab (PT)

This order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The PT order advances the current buffer address to the address of the first character position of the next unprotected field. If PT is issued when the current buffer address is the location of a field attribute of an unprotected field, the buffer advances to the next location of that field (one location). In addition, if PT does not immediately follow a command, order, or order sequence (such as after the WCC, IC, and RA respectively), nulls are inserted in the buffer from the current buffer address to the end of the field, regardless of the value of bit 2 (protected/unprotected) of the field attribute for the field. When PT immediately follows a command, order, or order sequence, the buffer is not modified.

The PT order resets the character attribute to its default value for each character set to nulls.

The display stops its search for an unprotected field at the last location in the character buffer. If a field attribute for an unprotected field is not found, the buffer address is set to 0. (If the display finds a field attribute for an unprotected field in the last buffer location, the buffer address is also set to 0.)

To continue the search for an unprotected field, a second PT order must be issued immediately following the first one; in reply, the display begins its search at buffer location 0. If, as a result of a PT order, the display is still inserting nulls in each character location when it terminates at the last buffer location, a new PT order continues to insert nulls from buffer address 0 to the end of the current field.

Repeat to Address (RA)

This order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The RA order stores a specified character in all character buffer locations, starting at the current buffer address and ending at (but not including) the specified stop address.

The buffer address is relative to the origin of the buffer of the partition to which the orders and data are directed, and only the character buffer locations of that partition are stored into.

The stop address is identified by the 2 bytes immediately following the RA order. The character to be repeated follows the stop address. For data streams that support graphic-escape (GE) sequences or 2-byte coded character sets, the character-to-be-repeated field can be 2 bytes long. The format of the RA order is:

X'3C'	Stop Address (2 bytes)	Character to Be Repeated

or:

X'3C'	Stop Address (2 bytes)	GE	Character to Be Repeated

The 12/14-bit and 16-bit addressing considerations discussed under "Set Buffer Address (SBA)" above apply to the RA order. Also, if an invalid stop address is specified (for example, greater than 1919 for a 1920-character screen), the write operation is terminated at this point without storing the character. The character to be repeated may be any 8-bit EBCDIC code or 2-byte graphic-escape (GE) sequence. If an EBCDIC code, including control codes, has a valid graphic representation in the device, then that graphic is displayed. Otherwise, the RA order is rejected (sense code X'1003' or X'1001').

Attribute values defined by a previous SA order(s) are applied to each repeated character.

When the stop address is lower than the current buffer address, RA wraps from the last buffer location to the first. When the stop address equals the current address, the specified character is stored in all buffer locations.

The current buffer address after successful completion of RA is the stop address, that is, one greater than the last buffer location stored into by RA.

Field attributes and their corresponding extended field attributes are overwritten by the RA order, if encountered.

Erase Unprotected to Address (EUA)

This order is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The EUA order stores nulls in all unprotected character locations, starting at the current buffer address and ending at, but not including, the specified stop address.

The buffer address is relative to the origin of the buffer of the partition to which the orders and data are directed and only the character buffer locations of that partition are filled with nulls.

The stop address is identified by the 2 bytes immediately following the EUA order. The format of the EUA order is:

X'12'	Stop Address (2 bytes)

The 12/14-bit and 16-bit addressing considerations discussed under "Set Buffer Address (SBA)" above apply to the EUA order.

If an invalid address is specified, the write operation is terminated at this point and no erasure (storing of nulls) or change of current buffer address occurs (a negative response of X'1005' returned).

When the stop address is lower than the current buffer address, EUA wraps from the last buffer location to the first. When the stop address equals the current buffer address, all unprotected character locations in the buffer are erased.

Field attributes and extended field attributes are not affected by EUA. Character attributes for every character changed to nulls are reset to their default.

The current buffer address after successful execution of EUA is the stop address.

Graphic Escape (GE)

The Graphic escape order is used to introduce a graphic character from an alternate character set.

Support of graphic escape is optional; when not supported, it is rejected. When supported, its 2-byte format is:

X'08'	Character Code

The use of graphic escape to select characters from an alternate character set follows these rules:

1. The GE + Character sequence always takes precedence over any other character set specified in the extended field attribute or character attribute, and applies to both inbound and outbound data streams, independent of the inbound reply mode.
2. GE + Code points of X'00' through X'3F' and X'FF' are undefined and are rejected (sense code X'1003').

3. GE + Code points of X'40' through X'FE' are valid graphic character codes.
4. If there is no alternate character set, the X'40' to X'FE' code points are rejected (sense code X'0863').

Format Control Orders

The following special control codes are classified as format control orders. These format control orders are stored in the character buffer and displayed as shown below:

Order	EBCDIC	ASCII	Displayed as
NUL	X'00'	X'00'	A blank, suppressed on Read Modified
SUB	X'3F'	-----	A solid circle
DUP	X'1C'	X'1C'	An overscore asterisk
FM	X'1E'	X'1E'	An overscore semicolon
FF	X'0C'	X'0C'	A blank
CR	X'0D'	X'0D'	A blank
NL	X'15'	X'0A'	A blank
EM	X'19'	X'19'	A blank
EO	X'FF'	-----	A blank

NUL is read back as a null (X'00') on a Read Buffer operation, but not read back on Read Modified operations.

NL, EM, FF, and CR are printer control codes with no display function. However, the code must be supported to the extent of being accepted and, on reading back, must appear as NL, EM, FF, and CR respectively. All are displayed as a space.

FM and DUP are displayed as above. When read back, they appear as the FM and DUP codes.

SUB, FM, and DUP may be entered from the keyboard. They are stored in the display buffer as controls; the current character set selection has no effect on them. They are transmitted to the application program as control codes.

Character Sets

There is a maximum of 190 language-dependent graphics in any character set. Typically, there are fewer than 128 unique graphics in most languages. Those codes not corresponding to defined graphics in a character set are displayed as undefined graphics; in that character set (what is displayed is an implementation option). Any data stream attributes associated with an undefined graphic still apply. If a GE code (X'08') is encountered, then the next character defined is from an alternate character set. The alternate character set, if used, is applied as the data stream is interpreted.

If graphic code point integrity is supported, the integrity of the undefined code points (X'40' through X'FE') must be maintained. That is, if a read operation follows a write operation, assuming no intervening operator alterations, the code points that are returned will be the same as those that were transmitted.

Two types of character sets are used by the 3270 data stream: nonloadable and loadable.

Nonloadable Character Sets

These character sets are provided with the device. They are assigned an identifying number (LCID) so that session partners can designate which set to use if more than one is provided, and so that nonloadable sets can be distinguished from loadable character sets. The nonloadable character set values are interpreted as follows:

- X'40' and X'0840' are a space.
- X'08nn' with nn = X'00' through X'3F' and X'FF' are control codes.
- X'41' through X'FE' and X'08nn', with nn = X'41' through X'FE', are language-dependent graphics.

Loadable Character Sets

When a device allows a user to define and store a character set, the sets are called *loadable character sets*. The loadable-character-set values are interpreted as follows:

- X'00' through X'3F' and X'FF' are control codes.
- X'40' is a nonloadable space.
- X'41' through X'FE' are loadable graphics.

There is a maximum of 190 loadable graphics. Data stream values X'41' through X'FE' are used to reference these loaded graphics.

Attributes

Attributes are sent in the 3270 data stream to define a field, the properties of a field, or a character or characters in a field.

Three attribute categories are used with the 3270 data stream:

- Field attribute
- Extended field attribute
- Character attribute.

The field attribute describes the field properties and occupies the first character position of each field in the character buffer and on the screen. It is displayed as a blank.

The extended field attribute, which provides additional field properties beyond that provided by the field attribute, defines field characteristics such as color, extended highlighting, the character set to be used, field outlining and what field validation to perform.

The character attribute is associated with individual characters. Each character in the buffer (except for field attributes) has a character attribute that defines a characteristic such as the color, extended highlighting, or the character set of the character. The extended field attribute is subordinate to the character attribute; that is, if a character attribute is present and defines a color, extended highlighting, or character-set value other than default, it overrides one or more of the attribute types identified in the extended field attribute.

Neither extended field attributes nor character attributes occupy a position in the character buffer, and they are not displayed or printed. Only the attribute properties kept in the extended field attribute and character attribute are sent in properties kept in the extended field attribute and character attribute are sent in the data stream. (The control sequences used to send this information are defined under "Orders.") No matter which control sequence is used, the attribute's properties are sent as a type and value pair:

Attribute Type	Attribute Value

For example, the data stream contains an order containing an attribute type of *color* and an attribute value of *green*, or a type of *extended highlighting* and a value of *blink*. This pairing of types and values is discussed later in this chapter.

Field Attributes

A field attribute, in addition to defining the start of a field, defines the characteristics of all character locations in the field. These characteristics are shown in Figure 4-1 for the 3270 Field attributes.

Fields can be protected from modification by the operator, or unprotected (available for the operator to modify or enter data). The unprotected definition classifies a field as an input field.

Alphanumeric fields can contain alphabetic, numeric, and symbol characters, plus space and null. Numeric fields are limited to numeric characters, the minus and decimal-sign characters, and the duplicate (DUP) control. Numeric fields have special meaning for protected fields, Data Entry keyboards, and the Numeric Lock special feature.

Tab stops are at the first character position following the field attributes of unprotected fields.

Bits 0-1	2	3	4, 5	6	7
Graphic Converter	U/P	A/N	D/SPD	Reserved	MDT

Table 4-1. Bit Definitions for 3270 Field Attributes

Bit	Description
0, 1	The function of bits 0 and 1 is to make the field attribute an EBCDIC/ASCII translatable graphic character. Bits 0 and 1 are set in accordance with Figure D-1.
2	0 = Field is unprotected. 1 = Field is protected.
3	0 = Alphanumeric. 1 = Numeric (causes an automatic upshift of data entry keyboard). **Note:** Bits 2 and 3 equal to B'11' cause an automatic skip of a protected field.
4,5	00 = Display/not selector-pen-detectable. 01 = Display/selector-pen-detectable. 10 = Intensified display/selector-pen-detectable. 11 = Nondisplay, nondetectable (not printable).
6	Reserved. Must always be 0.
7	Modified data tag (MDT); identifies modified fields during Read Modified command operations. 0 = Field has not been modified. 1 = Field has been modified by the operator. MDT can also be set by a program in the data stream.

The field attribute also indicates whether the field can be "detected" with a selector pen and whether it is to be displayed at a brighter level (intensified) than normal. The intensify function is the only highlighting function in the 3270 field attribute. The remaining highlighting functions are in the extended field attribute and character attribute.

Each field attribute occupies a location in the buffer but cannot be displayed. During a display, its character location appears as a space.

Field attributes are treated as characters that are protected from operator access; that is, they cannot be replaced by alphanumeric characters entered from the keyboard or modified by use of the selector pen. They can be altered, however, by the following actions:

- When the operator changes an unprotected field, which includes pressing the erase end of file (EOF) key, the display sets the modified data tag (bit 7) of the field attribute to 1.
- When the operator uses the Clear key, field attributes and all other characters in a formatted buffer are set to nulls (X'00').
- When the application program writes data, field attributes are not protected from being overwritten by alphanumeric data in the data stream.

- When the ERASE INPUT key is pressed, it sets off the MDT bit on all unprotected fields.
- When the erase end of file (EOF) key is pressed, it sets on the MDT bit for the field in which the EOF key is used.

In an SNA/EBCDIC environment, the display ignores bits 0 and 1 in the field attributes received from the application program, bits 0 and 1 are set in accordance with Figure D-1, in Appendix D, in attributes. This is for compatibility with previously used application programs.

Extended Field Attributes

Extended field attributes specify these field properties:

- 3270 field attributes (see "Field Attributes")
- Field validation
- Extended highlighting
- Foreground color
- Background color
- Character set
- Field outlining
- Transparency.

The **field-validation** functions are mandatory fill, mandatory entry, and trigger. Mandatory fill requires the terminal operator to enter a character for each character position in the field if the content of the field is modified in any way. Mandatory entry requires the operator to enter data in the field. Trigger causes the data entered by the operator to be sent to the application program as soon as the operator fills each such field.

The **extended-highlighting** functions that can be specified in the extended-field attribute are blink, underscore, and reverse video.

Foreground Color allows for selection of the symbol color in a character cell when normally highlighted in an alphanumeric position.

Background Color allows for selection of the color of the area surrounding a symbol in a character cell when normally highlighted in an alphanumeric partition. Refer to "Query Reply (Alphanumeric Partitions)" on page 6-18 for more information.

Character set allows users to select the character sets to be used to display data when the user does not wish to use the default nonloadable character set provided with the device.

Field outlining allows the user to emphasize a field by putting a line on the left, right, top, and bottom of a field.

Transparency allows the user to determine whether or not the presentation space behind a character can be viewed. It can have its own value or the value of the extended field attribute, depending upon how it is used.

Character Attributes

Character attributes specify these character properties:

- Extended highlighting
- Extended color (referred to as color throughout this manual)
- Character set.

The highlighting, color, and character-set functions are identical with the functions described under "Extended Field Attributes" above, except that they apply to a specific character rather than to the entire field.

If the character attribute contains a default for highlighting, color, or character set, the attribute value for that attribute type is obtained from the extended field attribute if it exists. The rules for when a character defaults to the field's properties are discussed in greater detail next under "Conflict Resolution Between Attributes."

Conflict Resolution Between Attributes

Extended field attributes may be used to define the color, highlighting, field validation, and character set of the field. At times, however, there may be a conflict in the properties of the field as specified in the extended field attributes, if it exists, and the properties of individual characters in the field as specified in the character attributes. This conflict is resolved as follows:

- If there are no field attributes in the character buffer, that is, the buffer is unformatted, then the character attributes are always used in displaying or printing each character. If a character attribute specifies *default* for a particular property, the device default for that property is used in displaying or printing that character.
- If there are field attributes in the character buffer and if a character attribute specifies *default* for any character property (color, highlighting, or character set), the character is displayed using the value of that property established for the field in the extended field attribute. Otherwise, the character attribute overrides the field attribute.

Character attributes are associated with a character and not with the character's position in the buffer. Thus, whenever a character is overwritten by a new character (or cleared or erased), the old character attribute is overwritten by the character attribute of the new character. If a character is moved, for example, because of an insert or delete operation, the attribute moves with the character.

Following is a summary of the rules for resolving attribute conflicts:

If the screen is	And the character attribute is set to[1]	Then
Formatted	Default	The extended field attribute is used to display/print that character.
Formatted	Nondefault	The character attribute is used to display/print that character.
Unformatted	Default or nondefault	The character attribute is used to display/print that character.

[1]Each attribute type (color, highlighting, character set) is treated uniquely.

Attribute Types and Selection Rules

Extended field and character attributes are sent in the data stream as an attribute type-value pair following the SA, SFE, or MF order sequence.

Attribute types are coded according to the following rules:

X'00' - X'7F' Attribute value is a 1-byte binary number; it does not have bit significance.

X'C0' - X'FF' Attribute types have values that are bit-encoded (for example, Field Validation).

Following are the attribute types defined in this chapter; attribute types not shown are reserved and are rejected (sense code X'1003'):

		May be used to define:	
Type Code	Attribute Type	Extended Field Attribute	Character Attribute
X'00'	All character attributes	NA	Yes
X'C0'	3270 Field attribute	Yes	NA
X'C1'	Field validation	Yes	NA
X'C2'	Field Outlining	Yes	NA
X'41'	Extended highlighting	Yes	Yes
X'42'	Foreground color	Yes	Yes
X'43'	Character set	Yes	Yes
X'45'	Background color	Yes	Yes
X'46'	Transparency	Yes	Yes
Other	Reserved	NA	NA
NA = Not Applicable			

Attribute Values and Selection Rules

The value bytes for all defined attribute types are shown below. Attribute values not shown are rejected. The attribute value of X'00' is mandatory for all supported attribute types. Attribute values that are unknown or cannot be maintained and returned inbound by an implementation also must be rejected.

All Character Attributes

This attribute type resets all character attributes types that are specifiable in the Set Attribute order to their default value. Attribute types affected are color, highlighting, and character set. The only valid value setting is X'00'; all others are reserved.

Type	Value
X'00'	Reset All Attribute Types (X'00')

The attribute type X'00' may appear only in the SA order.

3270 Field Attribute

This attribute type identifies the value byte as a field attribute:

Type	Value
X'C0'	Field Attribute

The definition of each bit position of the field attribute value byte is shown in Table 4-1 on page 4-13.

If the 3270 field attribute specifies non-display, this will override any extended field or character attribute in respect to displaying or printing that field. However, the attribute values specified in the extended field attribute or the character attribute will be transmitted inbound.

If the 3270 field attribute specifies protected and the extended field attribute specifies mandatory fill, mandatory fill is ignored.

Extended Highlighting

This attribute type identifies the highlighting property of a field or character. The field may have only one highlighting property specified by the extended field attribute (such as blink or reverse video, but not both). A highlighting property specified by the extended field attribute does not affect the intensify property specified by the field attribute.

When a character is assigned a highlighting property using the character attribute, the character's property overrides (for that character) the property defined by the extended field attribute.

Type	Value
X'41'	Highlighting

Following are valid settings for the highlighting value byte:

Setting	**Highlighting Property**
X'00'	Default
X'F0'	Normal*
X'F1'	Blink
X'F2'	Reverse video
X'F4'	Underscore

All others reserved

*As determined by the 3270 field attribute

Color

This attribute type identifies the color properties of a field or character. The field may have only one color property.

When a character is assigned a color property using the character attribute, the character's property overrides (for that character) the property defined by the extended field attribute.

Type	Value
X'42'	Foreground color
X'45'	Background color

Following are valid settings for the color value byte:

Setting	**Color Property**
X'00'	Default color (defined by Query Reply [Color])
X'F7'	Neutral
All others	As indicated by Query Reply (Color)

The relationship of a particular color to a data stream value is indicated by the Query Reply (Color) structured field. A device not capable of displaying a particular color may accept the data stream value for the color and display the device default color indicated in the Query Reply (Color) structured field.

The settings X'00' and X'F7' have unique data stream meanings. The X'00' value selects the device default color indicated in the Query Reply (Color) structured field. The X'F7' value indicates that the color is defined by a triple-plane character set. If a single-plane or nonloadable character set is referenced, the color defaults to the single color specified for the X'F7' value by Query Reply (Color). With triple-plane character sets, the color of each character is composed by combining the red, green, and blue planes defined for that character in the field. Where the planes overlap, the color that is displayed or printed is an additive color, that is, it comprises all colors in the overlap. For example, for a display, if a blue dot and a green dot overlap, the result is a turquoise dot if the color attribute value is X'F7'. If the color attribute value is other than X'F7', the character is a single color.

If a device receives and accepts the color attribute type (X'42') and (X'45'), only the colors specified in Query Reply (Color) are used in the presentation of data.

For additional information, see "Query Reply (Color)."

Character Set

Identifies the character set to be used by the device to display the field or character. The field may have only one character set local character set identifier (LCID) assigned, but when a character (or characters) is assigned an LCID, the character's ID overrides (for that character) the field's LCID.

Type	Value
X'43'	Character Set

Following are valid settings for the character-set value byte:

Setting	**Character Set**
X'00'	Default character set.
X'40' to X'EF'	Local ID for loadable character sets.
X'F0' to X'F7'	Local ID for nonloadable character sets.
X'F8' to X'FE'	Local ID for two-byte coded character sets (See Chapter 12, "Double-Byte Coded Character Set (DBCS) Asia.")
All others reserved	

The contents of the nonloadable character sets X'F0' through X'FE' cannot be copied or read from the device.

X'FF' signifies a "free" character set and may appear as an LCID for any character set, but it is invalid if sent as a value byte (sense code X'1003').

For more information on the LCID, see "Load Programmed Symbols."

Field Outlining

Type	Value
X'C2'	Field outlining

Field outlining is the displaying and printing function of the field frame by the combination of horizontal lines and vertical lines, in order to improve the readability of the screen and print-out. Field outlining is used to highlight data in tabular form, for example, to show where input fields are.

Field outlining is specified for fields only and surrounds the field by any combination of overline, underline, vertical line on the left edge of the field, and vertical line on the right edge. The vertical line on the left edge of the field is drawn in the field attribute position. The vertical line on the right edge of the field is drawn in the next field attribute position. The height of both vertical lines is from the overline to the underline. The overline is drawn between the previous row and the current row. The underline is drawn between the current row and the next row. The length of both horizontal lines is from the left vertical line to the right vertical line. Four lines make a complete and solid rectangle when all of the four lines are specified to the field.

When the right vertical line is specified for a field and the left line is specified for the next field, these two vertical lines must be overlaid. When the underline is specified for a field and the overline is specified for a field on the next row, these two horizontal lines must be overlaid.

Field outlining is always displayed with normal intensity, non-reversed, non-blinking, default fixed color.

There are 16 kinds of field outlining available by using combinations of the four lines.

Value = B'00000000' : No outlining lines
B'00000001' : Underline only
B'00000010' : Right vertical line only
B'00000100' : Overline only
B'00001000' : Left vertical line only
B'00000011' : Underline and right vertical line
B'00000101' : Underline and overline
B'00001001' : Underline and left vertical line
B'00000110' : Right vertical line and overline
B'00001010' : Right and left vertical lines
B'00001100' : Overline and left vertical line
B'00000111' : Rectangle minus left vertical line
B'00001011' : Rectangle minus overline
B'00001101' : Rectangle minus right vertical line
B'00001110' : Rectangle minus underline
B'00001111' : Rectangle

Transparency

Type	Value
X'46'	Transparency

The Transparency attribute can designate that the space surrounding a character be transparent to the screen operator, or it can designate that the space be opaque.

These are the possible settings that the Transparency attribute can have:

Setting	Meaning
X'00'	Default.
X'F0'	Background is transparent (OR).
X'F1'	Background is transparent (XOR).
X'FF'	Background is opaque (non-transparent).

Field Validation

This attribute type defines the validation properties of the field. The field can have more than one property, and all are optional:

Type	Value
X'C1'	Field Validation

Following are valid bit settings for the field-validation value byte:

Bit	Setting	Meaning
0-4	B'0000'	Reserved, must be zero
5	B'1'	Mandatory Fill
6	B'1'	Mandatory Entry
7	B'1'	Trigger

The field validation functions of mandatory fill, mandatory entry, and trigger are controlled by the extended field attribute at the device.

The Start Field Extended (SFE) and Modify Field (MF) orders are the data stream orders used to transmit the types of field validation desired. The attribute type is X'C1', and the type-value pairs that may be specified are described under "Attribute Values and Selection Rules."

If no type-value pairs are transmitted, no field validation occurs. A field may be given more than one field validation property.

Mandatory Fill

Mandatory fill specifies that, if any data is entered into a field, then the field must be completely filled. The operator, for example, fills a field by:

- Entering data in every character position of the field.
- Not entering data in every position, but entering a DUP character or an error character in the last position filled.

An unprotected fill field is *primed* when the operator positions the cursor in the field and presses any of the following keys:

- Any key for entering data
- The delete key
- The ERASE EOF key.

A mandatory-fill field is *unprimed* by any of the following:

- A write operation to that partition
- An operator erase-input keystroke
- Successful fill validation for that field.

Fill validation occurs when the operator tries to move the cursor out of a primed mandatory-fill field. Similarly, when the operator tries to transmit data, if the cursor is positioned in a primed mandatory-fill field, then fill validation occurs.

Before a field is primed, or after it is unprimed, the operator may move the cursor through the mandatory-fill field without causing fill validation.

Fill validation checks the field for a DUP character or for null characters. If the field contains a DUP character, fill validation is successful and the keystroke is processed. If the field contains any null characters:

- The mandatory-fill input-inhibit condition is activated.
- The alphanumeric cursor is not moved out of the field.

If the field contains no null characters, the keystroke that caused fill validation is processed normally.

The RESET key must be used to remove the input-inhibit condition. After pressing RESET, the operator can proceed to fill the field (with data, blanks, and the SUB and DUP characters).

Notes:

1. With multiple partitions, the jump-partition key is not affected by the validation of a mandatory-fill field; that is, while the screen cursor is positioned within a mandatory-fill field, the operator may jump to another partition. The screen cursor is moved, but the current cursor position of the partition is not changed. Consequently, the jump-partition key does not cause fill validation to occur.
2. Similarly, validation is not affected by the jump key. If the operator causes the MDT to be set to 1, and then jumps to another partition and back again, fill validation applies only at field exit time.
3. If a field has an extended field attribute with mandatory fill and trigger set on, fill validation occurs before the trigger action.
4. If the cursor is positioned in a primed mandatory-fill field, any enter action—for example pressing a PF key— causes fill validation to occur.

Programming Notes

1. If mandatory fill is specified for a "protected" field, it has no effect on the field.
2. If the host sends to the display the MDT bit in a mandatory-fill field, the field is transmitted inbound on a Read Modified operation. Thus, if the operator does not type any data into such a field, this field, which may contain some nulls (that is, an "incomplete" fill field), is transmitted inbound.
3. Similarly, if a mandatory-fill field is "selectable," operator selection causes the MDT bit to be set on. Again, if the operator does not type any data into such a field, the incomplete fill field is transmitted inbound on a read-modified operation.

Mandatory Entry

The mandatory entry attribute specifies that the terminal operator cannot transmit character data without first making an entry in the field. The operator may transmit character data using:

- The ENTER key
- A PF key
- A selector-pen detect on a field with a designator of &
- A cursor select on a field with a designator of &
- A magnetic slot reader.

When the operator tries to transmit data, the extended field attributes and field attributes are examined to determine if any field has both the mandatory-entry and unprotected attribute properties. If such a field exists for which the MDT bit is 0, no data is transmitted and an input-inhibit condition is raised.

The cursor is moved to the first field in the presentation space that has an MDT bit of 0 and the unprotected and mandatory-entry properties. The operator must use the Reset key to remove the input-inhibit condition. The operator can then proceed to enter data (at least one character, blank, SUB, or DUP). The operator must redo the enter action for the data that was inhibited.

Trigger

The trigger attribute specifies that the host is to be notified (with an AID) once the terminal operator has finished entering character data in the field. The trigger property can be specified for any field. Trigger fields allow field-by-field validation and editing to be performed by the host. As the operator completes each such field, the field is transmitted to the host for validation. The host may then give an affirmative or negative reply. The details of this process are described below.

A trigger field is *primed* when the operator positions the cursor in the field and presses a data entry key, the delete key, or the ERASE EOF key.

A trigger field is *unprimed* by a write operation to that partition, by an operator erase-input keystroke, or by the trigger action for that field.

Trigger action occurs when the operator tries to move the cursor out of a primed trigger field. Before a field is primed, or after it is unprimed, the operator may move the cursor through the trigger field without causing a trigger action. Trigger action causes AID X'7F' to be sent to the application program together with the cursor address, field address, and field data.

The trigger action is caused by any keystroke (for example, TAB, UP, DOWN, or data entry) that would move the cursor out of the trigger field. This keystroke determines the next position of the cursor. However, the cursor remains in the trigger field while the host is validating the field. If an affirmative reply is received, then the cursor is moved to the next position unless it is overridden by an Insert Cursor order contained in the reply.

While the display is awaiting a reply from the host, the operator may continue to type data. All keystrokes are accepted and queued (except as listed below) until a reply is received. The cursor is not moved, however, and keystrokes are not displayed, until an affirmative reply is received from the host.

The attention and system-request keystrokes are not queued, but are acted on immediately. In addition, system request purges the queue.

The reply from the host can be affirmative or negative. A null RU chain with CD or EB, or a write operation with keyboard restore to the partition containing the trigger field, is an affirmative reply. This reply causes resumption of keystroke processing from the keystroke queue.

The queued keystrokes are processed after the operation is completed, that is, after the orders and data have been processed. If the write operation is caused by an Outbound 3270DS structured field in a WSF command, the queued keystrokes are processed after the Outbound 3270DS structured field is processed and before any subsequent structured fields in WSF are processed.

A null RU without change direction (CD) or end bracket (EB) is not a reply; nor is a read operation to any partition. A Read Partition structured field is rejected because the device is in retry state.

A read command is not a reply, but it is not rejected:

- The RM command transmits the trigger field.
- The RMA command transmits all fields that have the MDT bit set on.

Normally, the MDT bit will be set on (as a result of data entry). If the field is selectable, however, then deselection resets the MDT bit. Consequently, it is possible that the MDT bit is 0 when a trigger field is transmitted inbound.

The keystroke queue is unaffected by the operation and keystroke queuing continues. This permits the host application to obtain data, other than the trigger field itself, that may be needed for validation of the field.

Any other operation is a negative reply. The negative reply causes the keystroke queue to be emptied (purged) and an input-inhibit condition to be raised. (Typically, the host might perform a write, without keyboard restore, to display an error message, highlight the field or character in error, and reposition the cursor.)

Details of the possible replies are given below. The terms and concepts used here are explained in "Read Operations from Partitions" on page 3-14.

In summary, an affirmative reply is:

- A WSF command containing an Outbound 3270DS structured field, addressed to the inbound partition (INPID). The structured field specifies a write command and contains a WCC that specifies keyboard restore.
- If INPID is 0, a write command with a WCC that specifies keyboard restore.
- A null RU chain with CD or EB.

In summary, a negative reply is:

- A WSF command containing any of the following structured fields:
  - Outbound 3270DS, addressed to INPID, that specifies a write partition command and contains a WCC that does not specify keyboard restore.
  - Outbound 3270DS addressed to INPID that specifies an Erase/write or Erase/write alternate command.
  - Outbound 3270DS addressed to a partition where the PID is not INPID.
  - A control operation, namely, those operations requested by the Reset Partition, Load Programmed Symbols, Set Inbound Reply Mode, Create Partition, Destroy Partition, Activate Partition, and Set Window Origin structured fields.
- If INPID is 0, a write command with a WCC that does not specify keyboard restore, or an EAU command.
- If INPID is nonzero, a W, EW, EWA, or EAU command.
- A WSF command with no data included in the structured field.

A read operation does not constitute a reply, and the following are not replies:

- Read Modified, Read Modified All, or Read Buffer command.
- A Read Partition structured field contained in a WSF command.

Where a WSF command contains several structured fields, the first structured field is interpreted as the reply; a subsequent structured field is not regarded as a reply.

If the reply from the host is a write operation that includes an IC order, then the cursor is positioned at the address specified in the data. If the reply is negative, the cursor is moved from its position within the trigger field to where the host has placed it, and the invalid-field input-inhibit condition is raised.

Note: These replies apply only if the command is processed successfully, that is, if there are no data stream errors.

Programming Notes

1. To avoid operator confusion, the application program should not move the cursor out of the trigger field on a negative reply.
2. If the reply is affirmative, the cursor is moved from its position within the trigger field to where the application program has placed it, and the keystroke queue is processed. Thus any data entry keystrokes are placed at the host-specified position.
3. The ability to move the cursor allows "logical cursor movement" to be programmed; that is, the position of the cursor can be made dependent on the data content of a previous field. If such a function is implemented, you should ensure that the operator understands the protocol for that application.

Notes:

1. Trigger action occurs if scrolling action moves the cursor out of a primed trigger field.
2. If the operator performs an enter action when an unmodified mandatory-entry field exits, the mandatory-entry input-inhibit condition is activated, and the cursor is moved to the mandatory-entry field. Thus, if this occurs while the cursor is positioned in a trigger field, the cursor movement causes a trigger action.

   An affirmative reply from the host causes the queued keystrokes to be processed. In this case, however, they will generally be ignored because of the input-inhibit condition.

   In this situation, moreover, and only in this situation, if the reply from the host was a write operation containing an Insert Cursor order, the host-specified cursor position is overridden by the mandatory-entry field cursor position; that is, after the reply from the host, the cursor is positioned in the mandatory-entry field and the mandatory-entry input-inhibit indicator is displayed.
3. A change of the active partition does not cause trigger action.
4. If the keystroke queue becomes full before a write is received from the host, an input-inhibit condition is raised.
5. The MDT bit is not changed by the trigger action. The host must decide what to do when a field check fails and then issue the appropriate orders in the data stream.
6. Normally, the MDT bit is set on as a result of data entry. If the field is selectable, however, deselection resets the bit. Consequently, it is possible that the MDT bit is 0 when a trigger field is sent inbound.
7. With multiple partitions, the jump-partition key is unaffected by the validation of a trigger field; that is, while the screen cursor is positioned within a trigger field, the operator may jump to another partition. The screen cursor is moved, but the current cursor position of the partition is not changed. Consequently, jump-partition does not cause a trigger action. Similarly, validation is not affected by the jump-partition key.
8. If a field has both the mandatory fill and the trigger attributes, fill validation will occur before the trigger action.

9. If the cursor is in a trigger field when the operator performs an enter action, this does not cause a trigger action unless an empty mandatory-entry field exists.

Processing of Character Attributes

When an extended field attribute is used to describe properties of a field, and that field also has characters described by character attributes, the character attribute overrides the extended field attribute for that character. The term *character attribute*, as used in the processing discussion below, applies to the effective character attribute determined from the interaction of the extended field attribute and the character attribute.

For all cases *except* where a triple-plane character set is referenced and the color value byte is X'F7' (neutral), the processing is as follows:

1. If the character attribute contains an ID for a *single-plane* character, the character code point and the character-set-value byte are used to obtain the character definition.
2. If the character attribute contains an ID for a *triple-plane* character set, the character code point and the character-set value are used to obtain *three* character definitions, which are then ORed together to produce a single definition.
3. If the character attribute has a highlight value of *reverse video*, then the character definition obtained from step 1 or 2 is complemented; that is, all 0 bits are changed to 1's, and all 1 bits are changed to 0's (assuming the device supports reverse video highlighting).
4. If the character attribute has a highlight value of *underscore*, then an underscore bar is effectively ORed into the character definition obtained from step 1 or 2.

Where a triple-plane character set is referenced and the color value byte is X'F7', the processing is as follows:

1. The character code point and the character-set-value byte are used to obtain three character definitions, one from each of the red, green, and blue planes.
2. If the character attribute has a highlight value of *reverse video* then each of the three character definitions is complemented (assuming that the device supports *reverse video*).
3. If the character attribute has a highlight value of *underscore*, then an underscore bar is effectively ORed into each of the three character definitions obtained from step 1.
4. Each of the three character definitions is displayed in the color associated with its plane.

Where there is an overlay in character definitions, that is, a dot is defined on more than one plane, then the color displayed for that dot is an additive color, as shown (except as defined in Note 2 on the next page):

Overlap in	**Displays/Prints as**
Red and Blue	Pink
Blue and Green	Turquoise
Green and Red	Yellow

Overlap in	**Displays/Prints as**
Green, Red, and Blue	Neutral

This is illustrated in the following example:

```
@..BB...     ........     ........       ...BB...
@..BB...     ........     ........       ...BB...
@..BB...     ........     ........       ...BB...
@..BB...     ........     GGGGGGGG       GGGTTGGG
@..BB...     ........     GGGGGGGG       GGGTTGGG
@..BB...     .RRRRRR.     GGGGGGGG       GYYNNYYG
@..BB...     .RRRRRR.     GGGGGGGG       GYYNNYYG
@..BB...     .RRRRRR.     ........       .RRPPRR.
@..BB...     .RRRRRR.     ........       .RRPPRR.
@..BB...     ........     ........       ...BB...

 Blue         Red          Green          Result
 Plane        Plane        Plane
```

Where: G = Green, R = Red, B = Blue, T = Turquoise, P = Pink,
N = Neutral, Y = Yellow, and . = none.

Notes:

1. Where identical definitions occur in all three planes, the operation of a triple-plane character set has the same result as if a single-plane character set had been referenced.
2. On devices that do not support a specified color, including colors that result from color-plane overlap, that color is displayed or printed in the device default color. (See "Query Reply (Color)" in Chapter 7 for more information.)

Defaults for Attributes

The attribute defaults are given in Table 4-2.

Extended field attributes always assume the default condition when coded X'00'. Character attributes assume defaults somewhat differently. If the character to which the character attribute applies is in a field (see the "Formatted Screen" column in Table 4-2) the X'00' code specifies that the properties associated with the field are to be used.

Table 4-2. Attribute Default Conditions

	Default Condition		
		For a Character	
Attribute Type	**For a Field**	**Formatted Screen**	**Unformatted Screen**
Field attribute	Unprotected, A/N, attribute display/nondetect, no intensify, MDT off	Not applicable	Not applicable
Color (foreground and background)	As specified in Query Reply (Color)	Default to field color	As specified in Query Reply (Color)
Extended highlight	As specified in Query Reply (Highlight)	Default to field highlight	As specified in Query Reply (Highlight)
Character set	Default to character set X'00'	Default to field character set	Default to character set X'00'
Field validation	None	Not applicable	Not applicable
Field outlining	No outlining line	Not applicable	Not applicable
Transparency	As specified in Query Reply (Field Outlining)	Default to field background transparency	As specified in Query Reply (Background Transparency)

Chapter 5. Outbound Structured Fields

Introduction

As pointed out earlier, the 3270 data stream is a formatted data stream. To provide additional controls and transmit various data types other than character it is necessary to use structured fields. The structured-field syntax permits variable-length data and controls to be encoded in such a way as to facilitate processing a sequence of fields into component fields without having to examine every byte.

Variable-length structured fields are achieved by providing a *length* as the first parameter of the structured field, thus:

```
|<----structured field 1------>|<------structured field 2---->|

 length 1 | ID | information | length 2 | ID | information

|<-------length 1------------->|<----------length 2---------->|
```

In the 3270 data stream, structured fields are introduced with the Write Structured Field (WSF) command. This command does not contain explicit control information as do other 3270 commands; it simply means all subsequent data is in a structured-field format. A WCC does not follow the WSF command as it does other write-type commands.

Following the WSF command, all data in the transmission must be in structured-field format. A structured field is of the form shown in Figure 5-1.

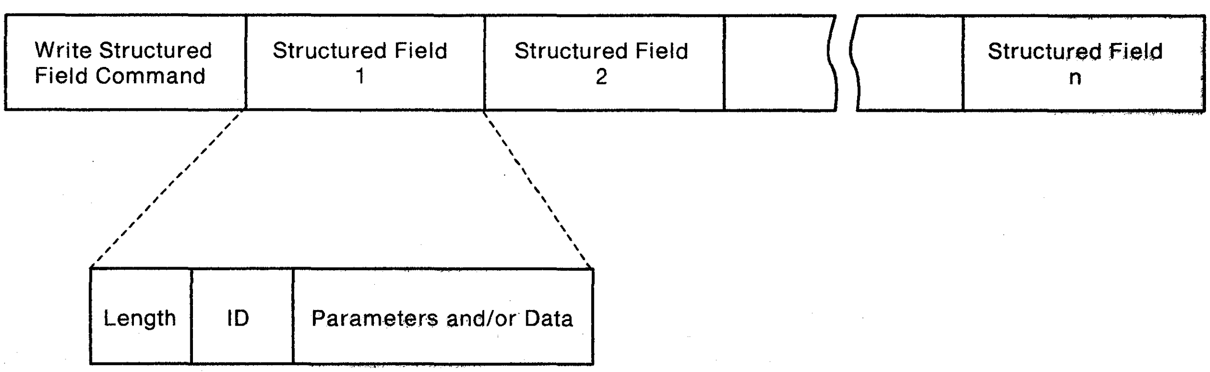


Figure 5-1. Structured-Field Format

Each structured field contains a 2-byte length field. This defines the length of the structured field (including the length bytes). Next follows an ID field that defines the function of the structured field, followed by parameters and data in the format defined by the ID code. A length of zero on an Outbound 3270DS or Inbound 3270DS indicates:

- The length of the structure was not determined.
- This is the last or only structured field in the transmission (for SNA, the transmission = chain).
- The length of the structured field should be determined using the end of the transmission (for SNA, transmission = chain).

The structured fields are divided into outbound and outbound/inbound structured fields and are described in alphabetic order in each section. The following list of structured fields defined in this chapter shows them in hexadecimal order by their identification (ID) field. Major operations within the structured field are listed by their operation code.

Outbound Structured Fields

ID	Name
0E	Activate Partition
0F85	Begin/End of File
0C	Create Partition
0D	Destroy Partition
03	Erase/Reset
0F05	Load Color Table
0F07	Load Line Type
06	Load Programmed Symbols
0F0A	Modify Partition
0F71	Outbound Text Header
0FC1	Outbound Type 1 Text
40	Outbound 3270DS
4B	Present Absolute Format
4C	Present Relative Format
01	Read Partition
	02 = Query
	03 = Query List
	6E = Read Modified All
	F2 = Read Buffer
	F6 = Read Modified
1030	Request Recovery Data
00	Reset Partition
1033	Restart
41	SCS Data
0F04	Select Color Table
4A	Select Format Group
1032	Set Checkpoint Interval
0F01	Set MSR Control
0F84	Set Printer Characteristics
09	Set Reply Mode
0B	Set Window Origin

Outbound/Inbound Structured Fields

ID	Name
0F21	Data Chain
0F02	Destination/origin
0F11	Object Control
0F0F	Object Data
0F10	Object Picture
0F1F	OEM Data
1034	Save/Restore Format
0F83	Select IPDS

Structured Field Grouping

Structured fields must not span transmissions. It is an error condition if the transmission ends before a structured field's length count is satisfied. Also, control sequences (for example, SA) must not span structured fields. It is an error condition if the entire control sequence does not appear in the same structured field.

However, when structured fields are grouped as described in this section, the non-spanning rules apply on a group basis rather than on a structured field basis.

General Description

Structured field grouping allows the relating of a number of structured fields into a single "logical" entity.

Grouping applies only to certain structured fields. Those structured fields that may be grouped must have a Group parameter in bits 0 and 1 of the first flag byte. The definition of the Group parameter is:

- B'00' = continue
- B'01' = end
- B'10' = begin
- B'11' = only

In the first structured field of a group, the Group parameter is set to "begin". The group parameter is set to "continue" in the second through the next-to-last structured fields of the group. The group parameter is set to "end" in the last structured field of the group. When the structured field is not a member of a group of related structured fields, the group parameter is set to "only"; the non-spanning rules apply in this case.

In addition to the explicit termination of a structured field group by a group parameter = 'end', the following will terminate, without error, *all* groups in process to and/or from the device:

- An EW/EWA command with WCC = Reset
- BIND (SNA only)
- Power on reset
- Clear Local Function (for example, the CLEAR key)
- Copy command (BSC printer only).

Spanning

Spanning is an application of the structured field grouping functions. The spanning application is the more general use of structured field grouping. Spanning allows the grouping of multiple structured fields of the *same type* into a single "logical entity." The most significant use of spanning is in situations where the data to be sent exceeds the single structured field length limit (32B) and sending multiple independent structured fields is not acceptable.

The data to be sent may be divided into multiple structured fields without regard to control sequence boundaries; that is, control sequences may span structured fields and transmissions within the group.

When more than one transmission is used to send the group of structured fields utilizing spanning, the following apply:

- The length count of the last structured field in a transmission must be satisfied at the end of the transmission. That is, a structured field cannot span transmissions (one exception is if the Data Chaining structured field is used in conjunction with spanning, limited to the non-SNA environment).

- In SNA on outbound (to the device) more than one transmission may be sent without sending a change direction (CD) until the last transmission of the group.

The last structured field of a transmission may use a length value of zero. The non-spanning rules always apply at the end of the last structured field of the group.

There may be more than one structured field group in process at the same time to (or from) a device providing either of the following is met:

- The structured field groups have different structured field types.

- If two or more of the structured field groups have the same structured field types, then these groups must be associated with different partitions.

Similarly, individual (non-grouped) structured fields may be interspersed with group members, to (or from) a single device, provided either of the following is met:

- The individual structured fields are of a different type than any of the structured fields being grouped.

- Any individual structured field that has the same type as one or more of the structured fields being grouped, must be associated with a different partition than that of the group or groups.

On outbound (to the device) transmissions, the following will cause termination of a group in process and rejection of the transmission.

- Receiving a structured field with the Group parameter = "continue" or "end" when the group is not in existence.
- Receiving a structured field with the Group parameter = "begin" for a partition that already has a group, with the same structured type, in process.
- Receiving an individual (non-grouped) structured field for a partition which has a group, of the same type structured fields, in process.
- Receiving a transmission starting with any 3270 command except WSF or EW/EWA (with WCC = Reset).
- A structured field length count is not satisfied at the end of a transmission.

On inbound (from the device) transmissions, the following are invalid:

- Sending a structured field with the Group parameter = "continue" or "end" when the group is not in existence; that is, group had not been started by a previous structured field with Group parameter = "begin".
- Sending a structured field with the Group parameter = "begin" from a partition which already has a group, with the same structured field type, in process.
- Sending an individual (non-grouped) structured field from a partition that has a group, of the same type structured fields, in process.
- Sending a transmission where the length count of a structured field is not satisfied at the end of the transmission.
- When one or more structured field groups are in process, sending a transmission that does not start with AID X '88'.

Data Chaining (Non-SNA)

Data chaining is a unique application of structured field grouping whose use is limited to the Data Chain structured field.

The SNA protocols provide a chaining function that allows a long message to be divided up into small transmissions to match the capability of a device. The division can be done without regard to control sequence or structured field boundaries. Non-SNA protocols do not provide this type of function; however, a "chaining-like" function is provided at the data stream level by use of the structured field grouping function in the Data Chain structured field.

The data chaining function allows the "grouping" of structured fields of different types, including structured fields that do not support the Group parameter. The data chaining is provided by the Data Chain structured field, which uses the Group parameter to provide the "chaining" control. The data to be sent (which must be in structured field form) can be divided without regard to structured field or control sequence boundaries into a number of transmissions of a size to accommodate the device capability.

A WSF and a Data Chain structured field are added at the start of each transmission. The length value (X'06') of the Data Chain structured field covers just the Data Chain structured field, that is, it does not include the rest of the data in the transmission.

The Data Chain structured field in the first transmission has the Group parameter = "begin". In the next through the next-to-last transmission, the Group parameter = "continue". In the last transmission, the Group parameter = "end". In the first transmission, a structured field must start immediately following the Data Chain structured field. However, in subsequent transmissions, this is not necessarily so. The data following may be a continuation of a structured field started in a previous transmission. That is, the length count of a structured field may span transmissions. If the length count of a structured field is not satisfied at the end of a transmission (excluding the last transmission of a data chain), the remainder of the data to satisfy the count starts immediately *after* the Data Chain structured field in the next transmission. The length count of a structured field could in some cases carry across a number of transmissions.

In addition to independent structured fields, the data being "chained" may contain one or more groups of structured fields using the spanning function.

On outbound (to the device) transmissions, any of the following will cause termination of the data chain and rejection of the transmission:

- Receiving a Data Chain structured field with Group parameter = "continue" or "end' without having previously started a data chain with a Data Chain structured field with the Group parameter = "begin"
- Receiving a Data Chain structured field with the Group parameter = "begin" while a data chain is already in existence
- Receiving a transmission that starts with any 3270 command except a Copy (BSC only), a WSF (followed by a valid Data Chain structured field) or an EW/EWA (with the WCC = Reset).

On inbound (from the device) transmissions, the following are invalid:

- Sending a Data Chain structured field with the Group parameter = "continue" or "end" without having first begun the data chain with a Data Chain structured field with the Group parameter = "begin"
- Sending a Data Chain structured field with the Group parameter = "begin" while a data chain is already in existence.

The INCTRL parameter in the Data Chain structured field allows the host application to enable/disable the use of inbound data chaining. The default (for instance, POR) is inbound data chaining disabled. This prevents a host application receiving a structured field it does not "understand." Once enabled, inbound data chaining is allowed until any one of the following occur:

- Receive an outbound Data Chain structured field with INCTRL = Inbound Data Chaining Disabled
- Receive an EW/EWA command with WCC = Reset
- POR
- Clear local function (for instance, Clear Key).

An outbound Data Chain structured field with INCTRL = no change causes no change in the inbound data chaining state. If inbound data chaining is already enabled, an outbound Data Chain structured field with INCTRL = enable leaves inbound data chaining enabled. If the inbound data chaining is disabled, an outbound Data Chain with INCTRL = disabled leaves inbound data chaining disabled.

Structured Field Self-Defining Parameters

Self-defining parameters are extensions to structured fields that require optional parameters. This parameter format allows for 255 such optional parameters, each containing up to 127 bytes.

Format of Self-Defining Parameters

All self-defining parameters have the following format:

Bytes	Content	Content Description
0	Parameter length	
1	Parameter identifier value	
2-n	Parameters	

Note: All Reserved fields must be zeros; in the tables shown in this chapter, the word *Reserved* represents this value.

Additional Content Description

- **Parameter length** is a 1-byte binary value in the range 2 to 127. Values less than 2 are invalid, values greater than 127 are reserved (the high-order bit of the length byte must be zero). The length value includes 1 byte for Parameter Length and 1 byte for the Parameter Identifier Value. There is no default value.
- **Parameter identifier value** is a 1-byte binary value that characterizes the information content of this self-defining parameter. It may have any value listed in the outlined value list. There is no default value.
- **Parameters** are variable length field(s) that contain data needed to perform the function specified by the Parameter Identifier Value. This field may be from zero to 125 bytes in length.

Outbound Structured Fields Only

The **Outbound Structured Fields** (application to the device) are defined in the following pages. Structured-field-type codes not shown will be rejected. Unless specifically stated to the contrary, any bits or fields classified as reserved are checked for zero value. Nonzero values are rejected (see Appendix B).

Structured-field-type codes not listed are rejected. Unless specifically stated to the contrary, any bits/fields classified as reserved must be checked for zero value; nonzero values are rejected.

Activate Partition

Function

This structured field is used to make an inactive partition active.

1. If the named partition does not exist or has a window of zero extent, the structured field is rejected. (See Table B-1 on page B-2.)
2. The named partition is made active.
3. The screen cursor is displayed at the current cursor position of the named partition.
4. Any saved input inhibit conditions for the partition are raised, and the appropriate indicators are displayed.
5. Input-inhibit conditions are handled as follows:
   a. System-lock and PWAIT conditions are saved for the active partition. Their indicators are removed.
   b. All other input-inhibit indicators remain displayed.

Format

Bytes	Content	Content Description
0-1	X'0004'	Length of this structure
2	X'0E'	Activate Partition
3	PID	Partition identifier (X'00' through X'7E')

Begin/End of File

Function

Indicates to the device that a new file is beginning or that the current file is completed.

Format

The format is shown in the table below.

Byte	Bit	Content	Content Description
0 - 1		X'0007'	Length of structured field
2 - 3		X'0F85'	Begin/End of File
4		PID	Partition ID
5		**Flag 1**	
	0 - 1	B'00'	Reserved
		B'01'	End of File is being sent
		B'10'	Begin of File is being sent
		B'11'	Reserved
	2 - 7		These bits are reserved
6		**Flag 2**	Reserved

Create Partition

Function

Creates a new partition.

If the display is in implicit partition state, implicit partition 0 is destroyed by the Create Partition and a partition identifier equal to PID is created. The display state is changed from implicit partition state to explicit partitioned state and the created partition is made active.

If the display is in explicit partitioned state, the Create Partition will:

- Destroy the partition with identifier equal to PID, if it exists, and replace it.
- If a partition does not exist with an identifier equal to PID, a new partition with an identifier equal to PID is created. The newly created partition is made active if it is the only partition with a window of nonzero extent.

The viewport of the new partition is assigned to origin (RV, CV) relative to the usable area, width WV and height HV. If any of the following conditions exists, the structured field is rejected:

1. WV=0 and modify partition is not supported.
2. HV=0 and modify partition is not supported.
3. RV+HV > screen height.
4. CV+WV > screen width.
5. The viewport overlaps another viewport.
6. The viewport cannot be contained within the usable area.
7. The current buffer address for the new partition is set to zero.
8. The current cursor position for the new partition is set to the window origin.
9. The nonloadable character set is established from the list returned in the character sets query reply according to the value of BASE.

   Whenever the partition is active, keyboard operation (including the association of each key with its I/O code) is established by the specified Base Character Set.
10. The window origin offset is set to RW, CW. If the window is not wholly contained within the presentation space, an error occurs.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2		X'0C'	Create Partition
3		PID	The identifier for the new partition X'00' through X'7E'
4	0-3	UOM B'0000' B'0001' B'0010' Other	The unit of measure: Character cells Addressable points (H thru PH) Addressable points (RV & CV) Reserved
	4-7	A-MODE B'0000' B'0001' B'0010' Other	Address mode: 12/14-bit binary 16 bit binary Text Reserved
5	0	Flags RES	 Reserved
	1	PROT B'0' B'1'	 Unprotected partition Protected partition
	2	COPY B'0' B'1'	Type of host-initiated local copy: Viewport Presentation space
	3-4	RES	Reserved
	5-7	BASE	Base character set index
6-7		H	The height of the presentation space
8-9		W	The width of the presentation space
10-11		RV	The y, or row, origin of the viewport relative to the top edge of the usable area
12-13		CV	The x, or column, origin of the viewport relative to the left side of the usable area
14-15		HV	The height of the viewport
16-17		WV	The width of the viewport
18-19		RW	The y, or row, origin of the window relative to the top edge of the presentation space
20-21		CW	The x, or column, origin of the window relative to the left edge of the presentation space
22-23		RS	The number of units to be scrolled in a vertical multiple scroll
24-25		RES	Reserved
26-27		PW	The number of points in the horizontal direction in a character cell in this presentation space
28-29		PH	The number of points in the vertical direction, in a character cell in this presentation space

Working backwards from byte 29, the above parameters may be progressively omitted by specifying the appropriate length of the structured field.

The actual size and position of the partition viewport on the screen is computed using the parameters PW and PH, or their defaults.

The defaults for omitted parameters are as follows:

Parameter	Default Value
PID	0
UOM	B'0000'
Address Mode	B'0000'
Flags	X'00'
H	Default screen height specified in BIND
W	Default screen width specified in BIND
RV	X'0000' (Top screen row)
CV	X'0000' (Left-most screen column)
HV	Smaller of H and height of usable area
WV	Smaller of W and width of usable area
RW	X'0000' (Top presentation space row)
CW	X'0000' (Left-most presentation space column)
RS	X'0001' if H.HV; X'0000' if H=HV
CS	X'0001' if W.WV; X'0000' if W=WV
PW	AW (Usable area cell width)
PH	AH (Usable area cell height)

Additional Content Description

- **UOM** - Bits 0-3 (byte 4) refer to the units of measure to be used in subsequent parameters of this structured field, which define the geometrical characteristics of the partition.

  For UOM =

  1. B'0000': these parameters are interpreted in the row, column coordinate system.
  2. B'0001': they are interpreted in the all points addressable system.
  3. B'0010': they are interpreted in the row, column coordinate system except for fields RV and CV, which are interpreted in the all points addressable system.

  UOM does not affect the type of data or the coordinate system that must be used when writing to the partition.

  The ability to support the various UOM options are indicated by the device in the Usable Area and Alphanumeric Partitions Query Reply structured fields as follows:

  1. If the CELL parameter of the Usable Area Query Reply is B'0', only UOM = B'0000' is supported.
  2. If the CELL parameter is B'1' and the APA parameter of the Alphanumeric Partition's Query Reply is B'0', UOM = B'0000' or B'0010' is supported.
  3. If the CELL parameter is B'1' and the APA parameter is B'1', UOM = B'0000', B'0001', or B'0010' is supported.

  For UOM = B'0010' instead of B'0001', the following defines the change in operation.

1. The viewport of the new partition is assigned to origin (RV, CV addressing in points) relative to the usable area. Viewport width WV and height HV are addressed with column, row counts. If any of the following conditions occur, the structured field is rejected:

    a. WV=0 and modify partition is not supported.

    b. HV=0 and modify partition is not supported.

    c. RV+(HVxPH) > screen height (points).

    d. CV+(WVxPW) > screen width (points).

    e. The viewport overlaps another viewport.

    f. The viewport cannot be contained within the usable area.

- **A-MODE** - If A-MODE is B'0010', a default text environment is established (Adjust off, no Tabs, text width = W, and left margin = 1).
- **RS** - If RS is X'FFFF', vertical scrolling is device dependent. This means that the device determines the amount of data movement, both for explicit requests from the operator and for automatic scrolling.

Notes:

1. If any of the errors identified above occur during the processing, the structured field is rejected. (See Table B-1 on page B-2.)
2. Any explicitly created partitions are destroyed by BIND, UNBIND, ACTLU, DACTLU, ACTPU and DACTPU.
3. If at the end of a transmission, the display is in SEND or CONTENTION state and all partitions have zero extent windows, the structured field is rejected.

Destroy Partition

Function

To destroy a named partition and, if this is the only existing partition, to create default implicit partition zero. *Destroy* means to:

1. Remove the PID from further accessibility
2. Prohibit any further display from the character buffer that had been assigned to the PID until that buffer space is erased or overwritten
3. Clear or overwrite the displayed viewport associated with the PID.

If the identified partition does not exist, Destroy Partition has no effect.

If the named partition is the only existing partition, then Destroy Partition:

1. Destroys the named partition
2. Resets the display to implicit partition state and creates the default Implicit Partition zero
3. Resets any Input Inhibit conditions
4. Resets INOP to Read Modified and INPID to zero.

If the named partition is not the only existing partition:

1. If the Destroy Partition is active, activates the next partition with a window of nonzero extent. If all other partitions have zero-extent windows, Destroy Partition activates the next partition, that is:
   a. The cursor is displayed at the current cursor position of that partition if it has a nonzero-extent window.

      If it has a zero extent window, the cursor is displayed in the right hand end of the operator information area.
   b. Any saved PWAIT or SYSTEM LOCK conditions for that partition are raised, and the appropriate indicators are displayed.
   c. If the specified PID = INPID, then INPID is reset to zero, and INOP is reset to Read Modified.
2. Destroy Partition destroys the named partition.

   Note: If at the end of a transmission, the SLU is in SEND or CONTENTION state and all partitions have zero-extent windows, an error situation exists.

Format

Byte	Content	Content Description
0-1	X'0004'	Length of this structure
2	X'0D'	Destroy Partition
3	PID	The partition identifier X'00' through X'7E'

Erase/Reset

Function

Resets the device to implicit partition state destroying all existing (implicit or explicit) partitions. Function creates an implicit partition zero with default partition characteristics and of default size if IPZ = B'0' or of alternate size if IPZ = B'1'.

This structured field is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Format

Byte	Bit	Content	Content Description
0-1		X'0004'	Length of this structure
2		X'03'	Erase/Reset
3		Flags	
	0	IPZ	Implicit partition size:
		B'0'	Default size
		B'1'	Alternate size
	1-7	RES	Reserved

Load Color Table

Function

Defines a loadable color table.

Several loadable color tables can be defined within one Load Color Table structured field. If the same entry is referred to more than once, the last definition is the one that is used. Also, the whole table can be reinitialized with a copy of the non-loadable color table.

Format

The format is shown in the table below.

Byte	Bit	Content	Content Description
0 - 1		L	Length of structured field
2 - 3		X'0F05'	Structured Field ID (Load Color Table)
4 - N		COMMAND	Command

Note: For information regarding the format and operation of the COMMAND parameter, refer to related graphics documentation.

Load Line Type

Function

Specifies the controlling environment defined-line types.

The controlling environment defined-line types are identified by code points greater than X'40' in the Set Line Type, and Push and Set Line Type orders. There can be up to 190 different controlling environment defined line types. The controlling environment defined line types are specified in self-defining parameters. These specify the format of the line type. Several line type definitions can be provided within a single structured field.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structured field
2-3		X'0F07'	Structured Field ID (Load Line Type)
4-n		COMMAND	Command

Note: For the definition of the format and operation of the COMMAND parameter refer to the appropriate graphics or image product publications.

Load Programmed Symbols (Load PS)

Function

Loads symbol and character definition information into the device.

This structured field causes characters to be loaded into contiguously addressable slots in program storage. The storage area in the device is called Read/Write storage (RWS) and, when supported, is identified in the device's Character Set Query Reply. This storage is made up of 191 contiguously addressable slots, associated with positions (X'40' to X'FE') in the data stream. Remember that the slot associated with X'40' cannot be loaded and contains a blank.

If this structured field does not contain data, the operation executes anyway. It executes as the parameters specify and no data is loaded. The *characteristics* of a character set may be changed without altering the *contents* of the character set. For example, a character set previously defined as requiring skip suppression may have this parameter changed to require no skip suppression.

Format

The format is shown in the table below.

Byte	Bit	Content	Content Description
0-1		L	Length of this structured field.
2		X'06'	Load Programmed Symbols (Load PS)

Byte	Bit	Content	Content Description
3		**Flags**	
	0	BASIC/EXT	Basic or extended form:
		B'0'	Basic form: only bytes 0 through 6 present
		B'1'	Extended form: length determined by byte 7
	1	CLEAR	Clear loadable character set:
		B'0'	Do not clear loadable character set
		B'1'	Clear all character slots not loaded
	2	SKIP	Skip suppress (Note 1):
		B'0'	Suppression not required
		B'1'	Suppression required
	3-7	TYPE	Data format type:
		B'00000'	Reserved
		B'00001' (Type 1)	18-byte form, the first 2 bytes contain a 16-bit vertical slice, the following 16 bytes contain 8-bit horizontal slices. For a 9 x 12 character matrix the last 4 bytes contain binary zero.
		B'00010' (Type 2)	Type 1 compressed
		B'00011' (Type 3)	Row loading (from top to bottom)
		B'00100' (Type 4)	Type 3 compressed
		B'00101' (Type 5)	Column loading (from left to right)
		B'00110' (Type 6)	Type 5 compressed
		B'01000' (Type 8)	Vector
		Others	Reserved
4		LCID	Local character set ID: Use X'40' to X'EF' Use X'FF' to indicate that RWS associated with this LCID is free (not assigned) Other values are reserved
5		CHAR	Beginning code point X'41' through X'FE'
6		RWS	Loadable Character Set RWS Number

Bytes 7 to 12 are parameters for the extended form of Load PS.

Byte	Bit	Content	Content Description
7		P LENGTH	Length of parameters for extended form, including the length parameter. The parameters defined below may be progressively included by specifying the appropriate length. Omitted parameters are assumed to have the defined default indicated by the zero value for that parameter.
8	0	APA B'0' B'1'	All points addressable (APA): APA Not APA
	1	CB B'0' B'1'	LCID compare bit: LCID compare No LCID compare
	2	OB B'0' B'1'	Operator selectable by PS key: Operator selectable Not operator selectable
	3	MULTID B'0' B'1'	Multiple LCID: Release any other symbol set with the same LCID Release any other symbol set with the same LCID unless it was loaded with MULTID=B'1'
	4	USET B'0' B'1'	Use Symbol Envelope Table Do not use S.E.T. information Use S.E.T. information
	5-6	RES	Reserved
	7	SDPP B'0' B'1'	Self-Defining Parameters Present No self-defining parameters present. Self-defining parameters follow.
9		LW	Number of X-units in character cell (width of character matrixes)
10		LH	Number of Y-units in character cell (depth of character matrixes)
11		SUBSN X'00' X'41'- X'FE' Other	Subsection ID One-byte codes Subsection ID for 2-byte coded data Reserved
12		COLOR X'00' X'01' X'02' X'04' X'nn'	Color planes: All planes Blue plane Red plane Green plane Color attribute value

Byte	Bit	Content	Content Description
13		ST. SUBS X'00' (default) X'41'- X'FE' Others	Starting Subsection Identifier Starting Subsection defined by QUERY REPLY—CHARACTER SETS. Starting Subsection Identifier for 2-byte coded character set specified by LCID in this LOAD PS. Reserved
14		ECHAR	Ending code point
15		NW	Number of width pairs
16		NH	Number of height pairs
17		RES	Reserved - must be zero

Notes:

1. Values for parameters other than those shown above are rejected; invalid parameter values are also rejected. (See Table B-1 on page B-2.)
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	Content Description
M to N		Data	Character matrices

Additional Content Description

- **BASIC/EXT** – The extended-form bit in byte 3 indicates whether this structured field is of the extended form. The extended form contains additional information associated with copy operations, character matrix size, and color. If the device does not support the extended form and this bit is set to 1, the data stream is rejected with a sense code of X'1005' or X'1001'.
- **SKIP** – If skip suppression is on, any row in 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 characters on the current row. If the skip suppression flag is not set, no skip suppression is required for characters from this character set.
- **CLEAR** – If the CLEAR flag is set on, all slots in the specified RWS are cleared. The character definitions in the data portion of the structured field, interpreted according to the data format type (byte 3), are then loaded into contiguously addressable slots in the PS RWS, starting at the position defined by CHAR.

  If the CLEAR flag is set but no character matrix data is included in the structured field, the addressed character set is cleared. Subsequent data stream references to this character set will produce blank characters. For a dynamically allocated symbol store, this means that the storage is freed. For a preallocated symbol store, this means that every symbol slot is set to binary zero.

- **TYPE** – If the data format type specifies *compressed* data (type 2, 4, or 6), the data is first decompressed. (See "The Compression Function" on page 5-26.)

  For data of type 1 or type 2 (decompressed), each set of 18 contiguous bytes defines a character in a 9-by-16 character cell.

  For data of type 3 (decompressed), each set of LW contiguous bits defines one row of a character matrix. The number of LH contiguous rows defines a character as a LW-by-LH matrix of dots, where LW is the width of the character matrix and LH is the height of the character matrix. If LW and LH are not supplied in the extended form of the LOAD PS structured field, or if they are supplied and set to zero, the values of LW and LH are determined by the device. Each character definition is extended with bits that can be ignored to a multiple of 8 bits. (In other words, the next definition will commence on a byte boundary.)

  For data of type 5 (decompressed), each set of LH contiguous bits defines one column of a character cell. The number of LW contiguous columns defines a character as an LW-by-LH matrix of dots. The bits are extended in the same fashion described for type 3 data.

  Loading of character definitions continues until either of the following is true:

  - 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 -RSP (with a sense code of X'1001' or X'1005') is returned.
  - The slot corresponding to X'FE' is loaded. Excess data is ignored and a -RSP (with a sense code of X'1001' or X'1005') is returned.

- **LCID** – The LCID specified in this structured field is released from any previously associated PS RWS number if the MULTID is equal to B'0' in this structured field, or if the MULTID is equal to B'1' in this structured field and was equal to B'0' in the structured field that previously loaded that LCID. There may be several programmed symbol sets, all with the same LCID co-residing in different slot sizes or with different types. Each would be loaded with MULTID equal to B'0'. The PS set RWS number released will be assigned an LCID = X'FF'; this will not be done until the LCID specified in this structured field is assigned to the associated RWS.

  The PS RWS 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. The LCID and PS RWS number specified in this structured field are associated for any subsequent data stream processing.

  An LCID of X'FF' indicates that this PS RWS is free (not assigned). A PS RWS with an LCID = X'FF' cannot be selected by the operator.

  On a Load PS, when an error condition is detected before the contents of the associated RWS are altered, an error sense code is sent but the LCID is not changed. Where the error condition occurs during the update of the RWS, the error condition is sent and the LCID is updated equal the one specified in Load PS. The above applies to a sequence of Load PSs sent after a WSF. Also, any sequence of Load PSs following the failing Load PS is not executed.

- **RWS** – Byte 6 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 is outside this range, the data stream is rejected (invalid parameter, sense code X'1005' or X'1001').

- **CHAR** – The content of this byte specifies the code point at which loading of symbol definitions and/or symbol envelope table values is to commence. The code point is incremented by one for each pair of a symbol definition and a symbol envelope table parameter (if both are provided in this structured field). Otherwise the code point is incremented by one for each symbol definition or symbol envelope table parameter (if only one category of information is provided in this structured field). Code points whose symbol definitions or symbol envelope table values are not updated by this command are unaltered (subject to the CLEAR parameter).

  Any particular Load PS Set command may have more or less symbol envelope table values than symbol definitions. Only values or definitions explicitly provided cause any update to either the symbol envelope table values or to the symbol definitions. In other words, symbol definitions without corresponding symbol envelope table values update only the symbol definition and leave any existing symbol envelope table values unaltered and vice versa. On invocation, any code point that has not explicitly had a symbol definition and/or symbol envelope table values is treated as a space and/or having symbol envelope table values of all zeros.

  Note that the above rule permits the symbol envelope table values and the symbol definitions to be loaded independently (before or after).

- Byte 8

  - **APA** – When set to B'1' (not APA), this bit implies that fewer than all points may be printed to allow a performance gain for specific devices. For example, 3287 *not* APA attempts to print all characters in one head sweep across the print line.

  - **CB** – When this bit is set to B'0' (Compare), the LCID of this character set is compared with character set LCIDs in the printer to determine whether there is a match. If the LCIDs match and the CB bits are both zero, the copy operation is performed using the corresponding LCID in the printer. If not, characters from the nonloadable character set of the printer are used.

    When 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.

  - **OB** – When set to B'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.

    The SDPP indicator specifies whether or not there are self-defining parameters provided prior to the symbol definitions in this structured field.

- **LW/LH** – If LW and LH for the character matrix dimensions are specified and are nonzero, then the character matrixes defined by LW and LH will be loaded into the specified PS RWS. (LW must be not more than the character slot width of the referenced character set. LH must be not more than the character slot height of the referenced character set.) They are loaded so that the first bit aligns with the upper left-hand corner of the character slot.

If either LW or LH exceeds the size of the character slot for the referenced character set, this structured field will be rejected.

If LW and LH are not specified or are set to zeros, the device assumes that the size of the character matrices transmitted in the LOAD PS structured field is the same as the device default character slot size.

If data of types 1 or 2 has been specified, then only the values LW = 9 and LH = 16 are valid. Any other values will be rejected.

- **COLOR** – 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 primary 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 bits are omitted, or if the COLOR field is B'000', then the character-set data is loaded into the specified character slots in all three planes.

  Additionally, a symbol can be constructed using the color attribute values supported by the device as reported in the Color Query Reply. Since a device must support a power-of-two number of colors, the color attribute values for the complete character slot resolves into color constituent parts (known as the primaries) for that code point. If some further load for that code point uses a different color attribute value then that too resolves into color constituent parts. This may destroy a previous loading of that color constituent part. When that symbol is invoked, the color constituent parts are ORed for each pel of the character slot (to reconstitute the color attribute value of that pel).

  For a single-plane character set, if the COLOR field is B'001', B'010' or B'000', the data stream is rejected.

- **ECHAR.** – This is the EBCDIC code point of the last symbol definition or Symbol Envelope Table entry contained in this Load Programmed Symbol Set structured field. If there are fewer symbol definitions or Symbol Envelope Table entries provided than "ECHAR - CHAR + 1," then the symbol definitions and the Symbol Envelope Table entries of the unspecified code points are unaffected. If there are more symbol definitions or Symbol Envelope Table entries provided than "ECHAR - CHAR + 1," then the structured field is rejected and the condition of the store is as it was prior to this Load Programmed Symbol Set structured field. The condition, however, is subject to the CLEAR parameter and with the symbol definitions or Symbol Envelope Table entries in the range CHAR to ECHAR (inclusive) updated. This parameter is relevant only to symbol sets that specify that the ending code point must be specified in any Load Programmed Symbol Set structured field addressed to them.

- **NW** – This is the number of pairs of width indentation for this symbol set. Besides setting this number for the whole symbol set, this parameter also prescribes the format of the Symbol Envelope Table self-defining parameter.

- **NH** – This is the number of pairs of height indentation for this symbol set. Besides setting this number for the whole symbol set, this parameter also prescribes the format of the Symbol Envelope Table self-defining parameter.

Symbol Envelope Table Self-Defining Parameter

Function

Provides the information pertinent to a single symbol. Therefore, there will be as many of these self-defining parameters as necessary. The first of these is the information for the symbol identified by the CHAR parameter. Note that every symbol of the set has the same numbers of indentation pairs. If this command creates a Symbol Envelope Table for this symbol store, then any symbols not assigned explicit values assume all zero values. Subsequent Load Programmed Symbol Set commands only alter the pair values (subject to the CLEAR parameter) for code points whose values are explicitly provided.

Format

Byte	Content	Content Description
0	L	Length of parameter
1	X'01'	Symbol envelope table
2 to 1 + 2NW	WPAIRS	Pairs of width indentation values.
2+2NW to 2NW +2NH+1	HPAIRS	Pairs of height indentation values.

Additional Content Description

1. **WPAIRS** - A list of pairs of 1-byte indentation values. Each pair is an indentation value from the left and right edges of the definition cell respectively and is expressed in the same units as the definition cell. There can be as many pairs of values as necessary to correctly envelope the symbol. If there is only one pair, then their values are assumed for the entire cell. Otherwise, the first pair of values applies at the bottom of the cell, and the last pair of values applies at the top of the cell. The sum of the values of any pair must not exceed the width of the definition cell.

2. **HPAIRS** - A list of pairs of 1-byte indentation values. Each pair is an indentation value from the bottom and top edges of the definition cell respectively and is expressed in the same units as the definition cell. There can be as many pairs of values as necessary to correctly envelope the symbol. If there is only one pair, then their values are assumed for the entire cell. Otherwise, the first pair of values applies at the left of the cell, and the last pair of values applies at the right of the cell. The sum of the values of any pair must not exceed the height of the definition cell.

Terminator Self-Defining Parameter

Function

Indicates the cessation of self-defining parameters. Any data following in this structured field is interpreted as symbol definitions.

Byte	Content	Content Description
X to Y	X'02FF'	Terminates self-defining parameters

The Compression Function

Symbol definition bit strings can be transmitted by the Load PS structured field function in uncompressed or compressed form. When the TYPE = 2, 4, or 6, the compression function proceeds in four parts:

- First, the cell is divided into appropriate pieces called *slices* and the slices are further subdivided into 4-bit *elements*.
- Second, the elements are compared to determine pattern matching.
- Third, the elements are grouped into *bunches*.
- Fourth, the element comparisons are encoded as *compressed character matrixes*.

An uncompressed symbol definition requires either 18 bytes of data (display) or 10 bytes of data (printer) to be transmitted. *Compression*, as described here, is a method for reducing the number of bytes transmitted.

An uncompressed symbol definition is created by dividing the character cell within which a symbol is formed into bytes (slices), as shown under the next heading, "Character Cell Division." The symbol is defined by encoding the bits (dots) in each byte (slice) as B'1' if the dot is to be *on*, or B'0' if the dot is to be *off*. The dot pattern representing the symbol is thus formed.

Byte (slice) 1 is understood to represent the leftmost upper vertical 8 dots in the display matrix or the leftmost vertical 8 dots in the printer matrix. The string of 144 bits (display) or 80 bits (printer) thus encoded represents the uncompressed symbol definition. A comparison process, comparing elements (4 bits) in the uncompressed bit string with reference elements selected from the same bit string, is used to compress the data.

Character Cell Division

The character cell for a display or printer character position is divided into slices as shown in Figure 5-2 and Figure 5-3. A slice corresponds to a byte, and the bits to the pels of the character cell.

In the case of the packed vertical slice, type 6, the slices correspond to columns of bits from the character matrix. Thus the slice length is the same as the character matrix height (LH), while the number of slices corresponds to the matrix width (LW). Therefore, the leftmost column of the character matrix image corresponds to slice 1 and so forth.

The packed horizontal slice, type 4, is similar though rotated 90 degrees. The slices correspond to the rows of the character matrix and the slice length is the same as the matrix width (LW), while the number of slices is the width height (LH). The topmost row becomes the first slice, as shown in Figure 5-2.

Once the character cell has been sliced in an appropriate manner, the slices can be thought of as forming a data string, beginning with slice 1, the 0 bit in each slice at the left.

Slice 1	Slice 2	0 1 2 3 4 5 6 7	....	Slice 10 or 18

As noted, each group of 4 bits is termed an *element*. Each slice is extended as necessary with zero bits so that the slice length is a multiple of 4 bits. The bit string forming the symbol definition is compressed by comparing each element to the corresponding element in another slice (or 0), and encoding the successful comparisons or matches. The compressed bit string is generated according to the matches and mismatches that occur in the comparison process.

The Compression Process

In creating a type 2 (display) or type 6 (printer) compressed bit string for an individual symbol, an algorithm based on one of four comparison rules is used. A header (of 1 to 4 bits) is used at the start of each symbol definition to signal which of the four comparison rules was used in the compression.

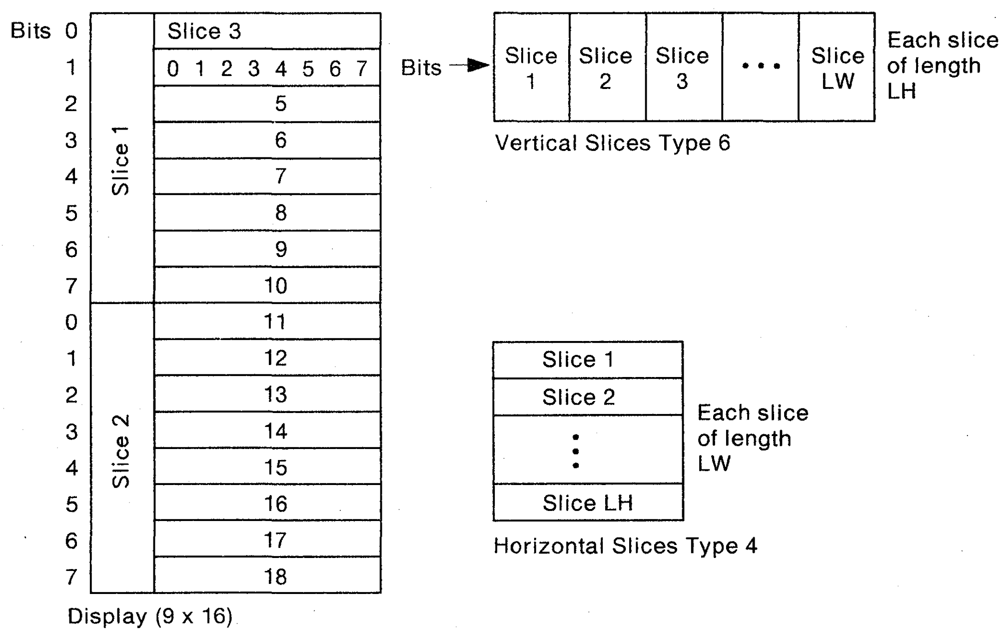


Figure 5-2. Vertical and Horizontal Slicing of a Character Cell

The compressed bit strings for all the symbols being defined are concatenated without regard for the byte boundaries, and then terminator bits are added to make the total bit string fit into an integral number of bytes. Type 2 or Type 6 data defining a full set of symbols in a Load PS structured field function is as follows:

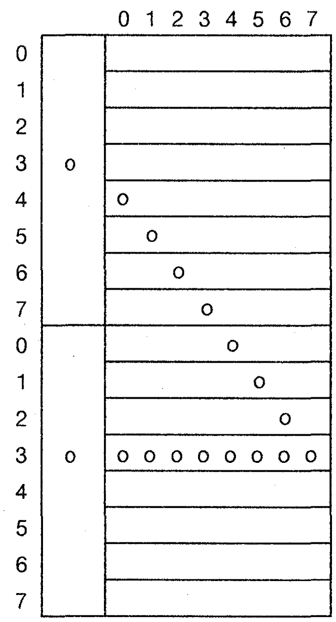
H	SD	H	SD	H	SD		H	SD	H	SD	T
Symbol 1		Symbol 2		Symbol 3		....	Symbol 189		Symbol 190		

H = header bits

SD = symbol definition bits

T = terminator bits

The following text describes the comparison rules and header bits, the creating of the compressed bit string, the terminator bits, and examples of compressing the symbol definition bit strings for the symbol in Figure 5-3.



In Type 1 data format, the dot pattern for this example symbol would be transmitted in 18 bytes having the following values:

Slice 1: B'00010000' = X'10'
Slice 2: B'00010000' = X'10'
Slice 3: B'00000000' = X'00'
Slice 4: B'00000000' = X'00'
Slice 5: B'00000000' = X'00'
Slice 6: B'00000000' = X'00'
Slice 7: B'10000000' = X'80'
Slice 8: B'01000000' = X'40'
Slice 9: B'00100000' = X'20'
Slice 10: B'00010000' = X'10'
Slice 11: B'00001000' = X'08'
Slice 12: B'00000100' = X'04'
Slice 13: B'00000010' = X'02'
Slice 14: B'11111111' = X'FF'
Slice 15: B'00000000' = X'00'
Slice 16: B'00000000' = X'00'
Slice 17: B'00000000' = X'00'
Slice 18: B'00000000' = X'00'

Figure 5-3. Type 1 Data Format - An Example Dot Pattern Encoded

The Comparison Rules and Header Bits

The four comparison rules that follow are used in creating a compressed symbol definition bit string from a Type 1 (display) or Type 5 (printer) uncompressed symbol definition bit string. Encoding the results of the comparisons is discussed in "Creating the Compressed Bit String" on page 5-29.

- Comparison Rule 1 (Header bit = B'0')

  Each element is compared with an element consisting of all 0 bits.

- Comparison Rule 2 (Header bits = B'1')

  Each element is compared with the corresponding element in the previous slice; for example, the first element of slice 2 is compared with the first element of slice 1, the second element of slice 2 with the second element of slice 1, the first element of slice 3 with the first element of slice 2, and so on. Since slice 1 has no previous slice, compare each element of slice 1 with a 0 element.

- Comparison Rule 3 (Header bits = B'110')

  Each element is compared with the corresponding element in the next-to-previous slice; for example, the first element of slice 3 is compared with the first element of slice 1, the second element of slice 3 with the second element of slice 1, the first element of slice 4 with the first element of slice 2, and so on. Since slices 1 and 2 have no next-to-previous slice, compare each element of slices 1 and 2 with a 1 element.

- Comparison Rule 4 (Header bits = B'1110') The symbol definition consisted of 0 bits only and no comparison is required. It is a blank symbol.

  If all of the header bits contain anything other than the bit combinations shown in the list above, the data stream will be rejected.

Creating the Compressed Bit String

The element comparisons are encoded by taking the elements four at a time (two slices — the slice-pair referred to in the following discussions) and comparing them with their corresponding elements in a reference slice-pair that has been created (following the comparison rules). Because the elements are compared four at a time, it is convenient to regard the 18 slices of the Type 1 data string or the 10 slices of a Type 5 data string symbol definition as being made up of nine or five slice-pairs. The elements of the slice-pairs are compressed as follows:

- Step 1
  Compare the first Type 1 or 5 slice-pair with the reference slice-pair.

  - Step 1A
    When the two slice-pairs are identical, put a 0 bit in the symbol definition bit string and repeat step 1 for the next slice-pair.

  - Step 1B
    When the two slice-pairs are not identical, put a 1 bit in the symbol definition bit string, and proceed to step 2.

- Step 2
  Compare, in turn, each element in the Type 1 or 5 slice-pair with the corresponding element in the reference slice-pair.

  - Step 2A
    For each element that matches (that is, the elements being compared are the same), put a 0 bit in the symbol definition bit string.

  - Step 2B
    For each element that does not match (that is, the elements being compared are not the same), put a 1 bit in the symbol definition bit string followed by a copy of the 4 bits of the nonmatching element from the Type 1 or 5 slice-pair.

- Step 3
  Repeat steps 1 and 2 in a similar manner through to the ninth slice-pair of a Type 1 string or the fifth slice-pair of a Type 5 string.

Terminator Bits

When the bit strings for all symbols have been created and concatenated, the Type 2 or Type 6 data string is completed with 1 bits to make up an integral number of bytes. There must be at least 4 of these terminator 1 bits, even if they spill over into a further byte. The number of 1 bits required thus ranges from 4 (minimum) to 11 (maximum).

Examples of the Compression Algorithm in Use

The following three examples show how a Type 1 data string for a particular symbol is compressed into a Type 2 data string. In these examples, the symbol whose Type 1 data string is being compressed is the one shown in Figure 5-3. Here is the data string for that symbol presented as nine slice-pairs:

Slice-pair 1: X'1010'
Slice-pair 2: X'0000'
Slice-pair 3: X'0000'
Slice-pair 4: X'8040'
Slice-pair 5: X'2010'
Slice-pair 6: X'0804'
Slice-pair 7: X'12FF'
Slice-pair 8: X'0000'
Slice-pair 9: X'0000'

For the particular symbol used in these examples, comparison rule 1 yields the shortest bit string; for any other symbol, however, the comparison rule that yields the shortest bit string depends on the symbol's particular dot pattern.

Example of Algorithm Using Comparison Rule 1

Slice-Pairs Being Compared according to Rule 1		Step 1: Compare slice-pairs. For a match, generate B'0'. For a mismatch, generate B'1' and do step 2.	Step 2: Compare elements. For a match, generate B'0'. For a mismatch, generate B'1' followed by a copy of the bits in the nonmatching Type 1 element.			
Reference Slice-Pair	Type 1 Slice-Pair		Element 1	Element 2	Element 3	Element 4
X'0000'	X'1010'	1	1 0001	0	1 0001	0
X'0000'	X'0000'	0				
X'0000'	X'0000'	0				
X'0000'	X'8040'	1	1 1000	0	1 0100	0
X'0000'	X'2010'	1	1 0010	0	1 0001	0
X'0000'	X'0804'	1	0	1 1000	0	1 0100
X'0000'	X'02FF'	1	0	1 0010	1 1111	1 1111
X'0000'	X'0000'	0				
X'0000'	X'0000'	0				

Figure 5-4. Example of Compression Algorithm Using Comparison Rule 1

With comparison rule 1, the header is B'0' and the symbol-definition bit string is created by comparing each Type 1 slice-pair with an all-zeros reference slice-pair as shown in Figure 5-4. The resultant bit string, including the header, is:

0110 0010 1000 1000 1110 0001 0100 0110 0100 1000 1010 1100
0010 1001 0100 1011 1111 1111 00

Note that the original type 1 bit string of 144 bits is compressed into 74 bits.

Example of Algorithm Using Comparison Rule 2

Slice-Pairs Being Compared according to Rule 2

Step 1: Compare slice-pairs.
For a match, generate B'0'.
For a mismatch, generate B'1' and do step 2.

Step 2: Compare elements.
For a match, generate B'0'.
For a mismatch, generate B'1' followed by a copy of the bits in the nonmatching Type 1 element.

Reference Slice-Pair	Type 1 Slice-Pair	Step 1	Element 1	Element 2	Element 3	Element 4
X'0010'	X'1010'	1	1 0001	0	0	0
X'1000'	X'0000'	1	1 0000	0	0	
X'0000'	X'0000'	0				
X'0080'	X'8040'	1	1 1000	0	1 0100	0
X'4020'	X'2010'	1	1 0010	0	1 0001	0
X'1008'	X'0804'	1	1 0000	1 1000	0	1 0100
X'0402'	X'02FF'	1	0	1 0010	1 1111	1 1111
X'FF00'	X'0000'	1	1 0000	1 0000	0	0
X'0000'	X'0000'	0				

Figure 5-5. Example of Compression Algorithm Using Comparison Rule 2

With comparison rule 2, the header is B'10' and the symbol-definition bit string is created by comparing each Type 1 slice-pair with a reference slice-pair composed of the previous slices, as shown in Figure 5-5. The resultant bit string including the header is:

1011 0001 0001 1000 0000 0111 0000 1010 0011 0010 0100 0101
1000 0110 0001 0100 1010 0101 1111 1111 1110 0001 0000 000

Note that the original Type 1 bit string of 144 bits is compressed to 95 bits.

Example of Algorithm Using Comparison Rule 3

Slice-Pairs Being Compared according to Rule 3		Step 1: Compare slice-pairs. For a match, generate B'0'. For a mismatch, generate B'1' and do step 2.	Step 2: Compare elements. For a match, generate B'0'. For a mismatch, generate B'1' followed by a copy of the bits in the nonmatching Type 1 element.			
Reference Slice-Pair	Type 1 Slice-Pair		Element 1	Element 2	Element 3	Element 4
X'0000'	X'1010'	1	1 0001	0	1 0001	0
X'1010'	X'0000'	1	1 0000	0	1 0000	0
X'0000'	X'0000'	0				
X'0000'	X'8040'	1	1 1000	0	1 0100	0
X'8040'	X'2010'	1	1 0010	0	1 0001	0
X'2010'	X'0804'	1	1 0000	1 1000	1 0000	1 0100
X'0804'	X'02FF'	1	0	1 0010	1 1111	1 1111
X'02FF'	X'0000'	1	0	1 0000	1 0000	1 0000
X'0000'	X'0000'	0				

Figure 5-6. Example of Compression Algorithm Using Comparison Rule 3

With comparison rule 3, the header is B'110' and the symbol-definition bit string is created by comparing each Type 1 slice-pair with a reference slice-pair composed of the next-to-previous slices, as shown in Figure 5-6. The resultant bit string, including the header, is:

1101 1000 1010 0010 1100 0001 0000 0011 1000 0101 0001 1001
0010 0010 1100 0011 0001 0000 1010 0101 0010 1111 1111 1110
1000 0100 0010 0000

Note that the original Type 1 bit string of 144 bits has been compressed to 112 bits.

Modify Partition

Function

Varies the partition parameters such as protection, viewport position, or window origin.

It has a similar syntax to Create Partition, where the detailed meaning of each parameter is explained.

- The structured field is checked for validity. If any of the following conditions exists, the structured field is rejected:
  1. The named partition does not exist.
  2. A reserved bit in a flag byte is not zero.
  3. The viewport overlaps another viewport.
  4. The viewport cannot be contained within the usable area.
  5. The window is not wholly contained in the presentation space.
  6. Bytes 8-11 are not X'00000000'.
- Working backward from byte 31 through byte 6, parameters may be omitted progressively by specifying the appropriate length. If a parameter is omitted in this way, its value is not altered. The specified values replace those currently in effect.
- If the viewport size is changed, the window size is changed accordingly. If as a result, the cursor is outside the window, the window is moved to contain it. Vertically, the window is moved by the minimum number of rows. Horizontally, the window is moved so that the cursor is in its center or the window abuts the edge of the presentation space, whichever occurs first. (If the window width is an even number of columns, the cursor will be at one of the two center positions.)
- If the window origin is changed and the cursor is outside the new window, the cursor is moved by the minimum number of rows and/or columns so that it is within the new window.
- If the size of the window is varied to zero and this partition is Active, the next partition (if any) with a nonzero window is activated. If every partition has a zero extent window, this partition remains active but the operator cannot type data into it, and the screen cursor is placed at the right-hand end of the indicator row.

  Note that if at the end of a transmission the display is in SEND or CONTENTION state and the active partition has a zero extent window, an error situation exists.
- If this is the only partition with a window of nonzero extent, then it is Activated.
- Note that some reserved fields in Modify Partition correspond to parameters in Create Partition. These partition parameters (for example, A-MODE) cannot be modified.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structure
2-3		X'0F0A'	Modify Partition
4		RES	Reserved
5		PID	Partition ID
6		RES	Reserved
7		Flags	
	0	RES	Reserved
	1	PROT	
		B'0'	Unprotected partition
		B'1'	Protected partition
	2-7	RES	Reserved
8-9		RES	Reserved
10-11		RES	Reserved
12-13		RV	Row origin of viewport
14-15		CV	Column origin of viewport
16-17		HV	Height of viewport
18-19		WV	Width of viewport
20-21		RW	Row offset of window origin
22-23		CW	Column offset of window origin
24-25		RS	Vertical scroll units
26-27		RES	Reserved
28-29		PW	Number of horizontal points in a character cell
30-31		PH	Number of vertical points in a character cell

Outbound Text Header

Function

Establishes initial conditions for a text partition.

The structured field is checked for validity. If any of the following conditions occurs, the structured field is rejected:

1. The partition A-MODE is not X'2'.
2. The named partition does not exist.
3. The OP-TYPE does not specify Write or Erase-Write.
4. Reserved flag bits are not B'0'.
5. LHDR < 2.
6. The HDR contains any graphics or unsupported controls.
7. The HDR contains formatting controls that are not self-consistent (for example, LM > width).
8. A parameter error occurs in any of the HDR controls.
9. CC > W-1.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structure
2-3		X'0F71'	Outbound Text Header
4		PID	Partition ID
5		OP-TYPE X'F1' X'F5'	Operation type: Write Erase/Write
6	 0-4 5 6 7	WCC RES B'1' B'1' B'1'	Reserved Sound alarm Keyboard restore Reset MDT
7		RES	Reserved
8		RES	Reserved
9		LVL	Cursor level
10-11		CRO	Cursor row offset
12-13		CC	Cursor column offset
14-15		LHDR	Header length includes itself (minimum value is 2)
16-n		HDR	Initial format controls

Additional Content Description

- **OP-TYPE**
  - If Erase/Write is specified:
    1. The current buffer address is reset to zero, so that subsequent text data will be loaded at the start of the buffer.
    2. The text buffer is cleared.
    3. The modified data tag (MDT) is reset.
    4. The window origin (row,column) is reset to (1,1).
    5. If Write is specified, the current buffer address is unchanged, so that subsequent text data will be appended to existing data.
  - **WCC**- If the WCC specifies reset MDT, the MDT is reset.
  - **LVL** - If LVL is X'FF', the LVL value is ignored. Otherwise, LVL identifies the element at which the cursor is to be placed (the first element in any position is at level 1).

    If LVL is X'00', level 1 is assumed.

    If there is no element at the specified level, the highest level element at that (row,column) position is assumed.
  - **CRO** - If CRO is X'FFFF', the cursor row is not changed. Otherwise, the cursor is placed in the specified row, (CRO+1), of presentation space.
  - **CC** - If CC is X'FFFF', the cursor column is not changed. Otherwise, the cursor is placed in the specified column, (CC+1), of presentation space.
  - **HDR** - If the HDR data is present, this is processed; otherwise the existing environment is unchanged.

    The HDR data is *not* stored in the text buffer and is *not* transmitted inbound.

Note that a subsequent Type 1 Text structured field may contain an insert cursor control. This will then override the cursor position defined by Outbound Text Header.

Formatting of the text buffer is started at the end of the next Outbound Text structured field or at end of the RU chain, whichever occurs first. When formatting is complete, the cursor position will be calculated and the WCC Keyboard Restore and Alarm bits take effect.

If the cursor is outside the window, the window is moved to contain it. Vertically, the window is moved by the minimum number of rows. Horizontally, the window is moved so that the cursor is in its center or the window abuts the edge of the presentation space, whichever occurs first. (If the window width is an even number of columns, the cursor will be at one of the two center positions.)

Outbound Type 1 Text Data Stream

Function

Transmits data to a text partition.

The structured field is checked for validity. If any of the following conditions exist, the structured field is rejected:

1. The partition A-MODE is not X'2'.
2. The named partition does not exist.

Format

Byte	Content	Content Description
0-1	L	Length of structure
2-3	X'0FC1'	Type 1 Text
4	PID	Partition ID
5-6	RES	Reserved
7-n	DATA	Text data

Additional Content Description

- **DATA**
  - The text data is interpreted and placed in the text buffer at the current buffer address, that is, following any previous data in the buffer.

    If any of the following errors occur during this process, the structured field is rejected:

    1. An unsupported X'2B' control is detected.
    2. A parameter error occurs in a X'2B' control.
    3. An unsupported 1-byte control is detected.
    4. The text data cannot be contained in the text buffer.
  - During this interpretation, nulls (X'00') are treated as no-op. They are not placed in the text buffer.
  - As the text data is interpreted, the current buffer address is incremented by 1 for each byte placed in the buffer. Note that the current buffer address is updated both by host writes, and by operator interaction (that is, it is always the address of the first empty byte in the buffer).
  - The buffer is formatted to the presentation space.
  - If DATA contains an insert cursor control, the screen cursor will be placed at the presentation position of the element immediately following this insert cursor control. If there are no elements in DATA following the insert cursor, the screen cursor will be placed at the presentation position immediately following that of the last element of DATA.

The window and cursor positions are then resolved as follows:

1. If the resulting cursor position is inside the window, the window position is unchanged.
2. If the cursor position is outside the window, the window is moved to contain it. Vertically, the window is moved by the minimum number of rows. Horizontally, the window is moved so that the cursor is in its center or the window abuts the edge of the presentation space, whichever occurs first. (If the window width is an even number of columns, the cursor will be at one of the two center positions.)

Note that the host can use the Modify Partition structured field to set the window origin.

Outbound 3270DS

Function

Directs 3270 data stream orders and data to a named partition.

This structured field is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Format

Byte	Content	Content Description
0-1	L	Length of this structure
2	X'40'	Outbound 3270DS
3	PID	Partition identifier (X'00' through X'7E')
4	CMND	Partition command:
	X'F1'	Write - The associated data is written to the buffers of the specified partition.
	X'F5'	Erase/Write - The viewport and buffers of the specified partition are cleared, and the data is written to the buffers as in a Write.
	X'7E'	Erase/Write Alternate - Same action as Erase/Write.
	X'6F'	Erase All Unprotected - All the unprotected fields in the named partition are set to nulls.
	X'F7'	BSC Copy
5	WCC CCC	3270 WCC byte for X'F1', X'F5', and X'7E' CCC for BSC copy (X'F7')
6-7	DATA	3270 Orders and data for X'F1', X'F5', and X'7E'
	FROM ADDRESS	BSC Copy (X'F7')
8-n	DATA	3270 Orders and data for X'F1', X'F5', and X'7E'

Note: Erase/Write and Erase/Write Alternate do not change the screen size. Use the Erase/Reset structured field to do this.

Additional Content Description

- **PID** — specifies to which partition the data in this structured field is to be directed. The specified partition must be defined or a data stream error occurs. (See Table B-1 on page B-2.) On a device that doesn't support multiple partitions, partition 0 must always be specified.
- **CMND** — contains control information for the partition. Note that, when CMND = X'6F', no WCC appears in the structured field. CMND values not shown are reserved and are rejected (sense code X'1003' or X'1001').

If the CMND byte specifies BSC Copy, the following requirements must be met for the copy to be valid. The copy is valid only when the "from" device and the "to" device are in implicit partition state.

1. The communications must be remote BSC.
2. The Outbound 3270DS structured field carrying the BSC Copy command must be the last structured field in the transmission.

An error sense code will be returned if these conditions are not met. See Table B-1 on page B-2 for other error conditions and the sense codes sense codes that are returned.

- **WCC** – contains the write control character as defined for the 3270 data stream in "Write Commands." When CMND = X'F1', bits 0 to 3 of the WCC byte are ignored for displays and bits 0 and 1 are ignored for printers. If PID is not equal to 0, only the indicated partition is reset.

  If no WCC byte is defined following an Erase/Write or Erase/Write Alternate command, then no erasing or resetting occurs. The structured field is treated as a negative reply to a trigger, but it does not acknowledge any outstanding read or entry.

Present Absolute Format

Function

Invokes a "named format" and presents it to a device.

To present a format that does not require relocation, the Present Absolute Format structured field can be used. This structured field will invoke the "named format" and present it to a display or printer. The content of the Format Presentation Command subfield specifies the type of command (Write, Erase/Write, or Erase/Write Alternate) used to present the format. The write control character used in the presentation is specified in the WCC subfield. The orders and data used to generate the format will be retrieved from storage in the control unit. The retrieval will be based on the Format Name specified in the Name subfield as qualified by the Format Group Name that is currently selected for the specified partition. If a Present Format request is made prior to selecting a format group for the partition, a "Format Group Not Selected" error condition will be reported, and the Present Format request will be rejected.

When the outboard node is unable to retrieve the specified format, the "Format Not Found" situation will be handled as an error condition.

Format

Byte	Content	Content Description
0-1	L	Length of structure
2	X'4B'	Present Absolute Format
3	PID	Partition ID (X'00' - X'7E')
4	FPC	Format Presentation Command:
	X'F1'	The data associated with the named format is written to the buffer of the specified partition (Write).
	X'F5'	The viewport and buffer of the specified partition are cleared, and the data associated with the named format is written to the buffer of the specified partition (Erase/Write).
	X'7E'	Same action as an Erase/Write.
5	WCC	3270 WCC byte
6-n	Name	Name of the format being invoked

Notes:

1. If PID is a text partition, an error occurs.
2. Erase/Write and Erase/Write Alternate do not change the screen size. Use the Erase/Reset structured field to do this.

Additional Content Description

Values of the Format Presentation Command that are -/tablewn are reserved. If transmitted, they are rejected. WCC bits 0,2 and 3 are ignored if transmitted to a display. If bit 1 of the WCC = 1, a reset is performed. If PID is not equal to zero, only the indicated partition is reset.

To invoke a format that does not require relocation, merge it with variable data transmitted in the data stream, and present it to a device, a sequence of Present Absolute Format structured field and Outbound 3270DS structured fields can be used. The variable data is transmitted in the Outbound 3270DS structured field.

```
Present Absolute Format Structured Field

┌──────┬──────┬──────┬──────┬──────┬──────┐
│ L    │ TYP  │ PID  │ FPC  │ WCC  │ Name ├──────┐
│ XXXX │ 4B   │ XX   │ XX   │ XX   │ 1-n  │      │
└──────┴──────┴──────┴──────┴──────┴──────┘      │
                                                 │
┌────────────────────────────────────────────────┘
│
│       Outbound 3270DS Structured Field
│
│  ┌──────┬─────┬─────┬──────────────────────────────────┐
└─►│ L    │ TYP │ PID │ 3270 data stream of variable data │
   │ XXXX │ 40  │ XX  │ W,WCC,orders and data             │
   └──────┴─────┴─────┴──────────────────────────────────┘
```

Data Stream

The relationship between the Format Presentation Command and the 3270 structured field command is as follows:

- The first operation performed is that specified by the Present Absolute Format structured field. The command is taken from the FPC subfield, the WCC is taken from the WCC subfield, and the orders and data used are retrieved from storage in the control unit based on the format name specified in the Name subfield.
- The second operation performed is that specified by the Outbound 3270DS structured field. The command is taken from byte 4 of the structured field. The WCC is taken from byte 5 and the orders and data from bytes 6 through n.

  Example 1

  To erase the screen, present a format, and merge variable data, an Erase/Write command should be specified as the Format Presentation Command in the Present Absolute Format structured field. A Write command should be specified in byte 4 of the Outbound 3270DS structured field to write the variable data to the screen.

Example 2

To present a format without erasing the screen and then merge variable data, a Write command should be specified as the Format Presentation Command in the Present Absolute Format structured field. A Write command should be specified in byte 4 of the Outbound 3270DS structured field to write the variable data to the screen.

Present Relative Format

Function

Invokes a "named format" and increments the buffer addresses contained in the format data stream by the 16-bit binary offset value, specified in the Format Offset Value subfield, prior to processing the format data for presentation to a device.

To present a format that requires relocation, the Present Relative Format structured field must be used. This structured field functions similar to the Present Absolute Format structured field, except that the buffer addresses contained in the named format are incremented by the 16-bit binary value, prior to processing the format data stream for presentation to a device. If a Present Format request is made prior to selecting a format group for the partition, a "Format Group Not Selected" error condition will be reported and the Present Format request will be rejected.

When the outboard node is unable to retrieve the specified format, the "Format Not Found" situation will be handled as an error condition.

Format

Byte	Content	Content Description
0-1	L	Length of structure
2	X'4C'	Present Relative Format
3	PID	Partition ID (X'00'-X'7E')
4-5	FOV	Format offset Value: 16-bit binary offset value that is added to the buffer addresses contained in the named format prior to the data stream processing.
6	FPC	Format Presentation Command:
	X'F1'	Write - The data associated with the named format is written to the buffer of the specified partition.
	X'F5'	Erase/Write - The viewport and buffer of the specified partition are cleared, and the data associated with the named format is written to the buffer of the specified partition.
	X'7E'	Same action as an Erase/Write.
7	WCC	3270 WCC Byte
8-n	Name	Name of the format being invoked.

Note: If PID is a text partition, an error occurs.

Additional Content Description

- **FPC** — Values of the Format Presentation Command not shown are reserved. If transmitted, they are rejected.
- **WCC** — WCC bits 0, 2 and 3 are ignored if transmitted to a display. If WCC bit 1 = 1, a reset is performed. If PID is not equal to zero, only the indicated partition is reset.

To merge variable data with a relocated format, a sequence of Present Relative Format and Outbound 3270DS structured fields must be used.

```
   Present Relative Format Structured Field

┌──────┬──────┬──────┬───────┬──────┬──────┬────────┐
│ L    │ TYP  │ PID  │ FOV   │ FPC  │ WCC  │ Name   ├──┐
│ XXXX │ 4C   │ XX   │ XXXX  │ XX   │ XX   │ 1-n    │  │
└──────┴──────┴──────┴───────┴──────┴──────┴────────┘  │
                                                       │
                                                       │
  ┌────────────────────────────────────────────────────┘
  │
  │          Outbound 3270DS Structured Field
  │
  │   ┌──────┬──────┬──────┬────────────────────────────────┐
  └──►│ L    │ TYP  │ PID  │ 3270 data stream of variable data │
      │ XXXX │ 40   │ XX   │ W,WCC,orders and data          │
      └──────┴──────┴──────┴────────────────────────────────┘
```

The structured fields are processed in the sequence in which they are transmitted in the data stream. Since structured fields can be utilized in various combinations, each structured field is processed as a separate entity. Using sequences of structured fields in the data stream allows multiple formats to be invoked and merged with multiple streams of variable data within one WSF transmission. Note that the Format Offset value has no effect on the Outbound 3270DS structured field.

Read Partition

Function

Reads data from a specific partition or queries the device.

This structured field is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The Read Partition structured field provides a means for the host to explicitly read a partition. In addition, the Read Partition structured field can be used by the host to request a 3270 data stream device or work station to report functions supported by the device or work station. This information is returned to the host in the form of Query Reply structured fields.

The inbound data stream in reply to Read Partition (Query) is:

X'88'	Query Reply	Query Reply

where the reply is sent as a sequence of Query Reply structured fields.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2		X'01'	Read partition
3		PID	Partition identifier: X'00' - X'7E' (read operations), or X'FF' (query operations)
4		TYPE X'02' X'03' X'6E' X'F2' X'F6'	The type of operation to be performed: Query Query List Read Modified All (RMA) Read Buffer (RB) Read Modified (RM)
5	0-1 2-7	REQTYP B'00' B'01' B'10' B'11'	Request Type - present only for TYPE = X '03' (QUERY LIST) QCODE List Equivalent + QCODE List All Reserved Reserved
6-n		QCODE	Query Reply Codes for TYPE X'03'

Additional Content Description

- **TYPE**
  - TYPE = X '02' - the structured field ends after byte 4.
  - TYPE = X '03' - byte 5 is a flag byte. Bytes 6-n contain the QCODEs of the Query Replies being requested.

  Note: Read type codes not shown here are reserved for future extension, and will be rejected.

- **REQTYP** - present only if TYPE = X '03'.
  - B'00' - (QCODE List) indicates the only Query Replys being requested are those specified in bytes 6-n. If the value is B '00' but no list is present (count field is valid), a Null Query Reply is returned.
  - B '01' - (Equivalent + QCODE List) indicates all the Query Replies that would be sent in response to a Query are sent in addition to those QCODEs (if any) that are specified in the QCODE list (bytes 6-n). No duplicate Query Replies are sent. If the QCODE list requests a Query Reply that would be sent anyway, because of the B'01' flag, the Query Reply is sent only once.
  - B '10' - (All) indicates that *all* the Query Replies that are supported are sent. If a QCODE list is present (bytes 6-n), the All flag overrides the list, that is, the list is ignored.

- **QCODES** – contains the Query Reply codes requested when the value in byte 4 is X'03'.

The same QCODE may appear more than once in the list (bytes 6-n). However, only one Query Reply will be returned for a particular QCODE value regardless of how many times it appears in the list.

All QCODE values in the list are valid. Those QCODEs not supported are ignored. However, if none of the QCODEs in the list are supported, a null Query Reply is returned.

If any of the following conditions exists, Read Partition is rejected (see Table B-1 on page B-2):

- The device is in RETRY state (sense code X'0871').
- The terminal is not in receive (RCV) or contention (CONT) state.
- Read Partition is not the last structured field in the RU chain.
- The RU chain does not specify CD.
- The operation type is Query and the PID is not X'FF'.
- The operation type is Read Modified, Read Modified All, or Read Buffer, and PID does not exist.
- The operation type is Read Modified All or Read Buffer, and PID specifies a text operation.
- The operation type is Read Modified All or Read Buffer, and PID specifies a text partition.

Otherwise, the following steps are performed:

1. The enter-inhibit condition is raised.
2. If the PID specified in Read Partition is not X'FF', INPID is set to the specified PID.
3. INOP is set to the operation specified in Read Partition.
4. For a read-modified, read-modified-all, or read-buffer operation, the AID is set to X'61'.
5. The data is transmitted inbound.
6. The display is placed in RCV state.
7. The device is placed in retry-read state.
8. For read operations with type code X'F6', X'6E', or X'F2', the data is transmitted inbound in the format specified by the operation and the inbound reply mode with an AID of X'61'.
9. For a query operation (type code X'02'), a set of query replies is transmitted inbound. These replies describe the features on that device.

Request Recovery Data

Function

Requests recovery data for Print Job Restart when sent from PLU to SLU.

Format

The format of this structured field is shown below:

Byte	Content	Content Description
0 - 1	X'0005'	Length of structured field
2 - 3	X'1030'	Request recovery data
4		Reserved

This structured field must flow to enable the SLU to send the recovery data to the PLU.

Reset Partition

Function

Resets definable partition characteristics.

The partition characteristics for the specified partition (PID) are reset to their default values:

Partition Characteristic	Default
Inbound Reply mode	Field
Format Group Selection	No format group selected
Set Partition Characteristics	None defined

Reset partition has no effect if the partition A-MODE is text.

Format

Byte	Content	Content Description
0-1	X'0004'	Length of this structure
2	X'00'	Reset Partition
3	PID	Partition identifier (X'00' - X'7E')

Restart

Function

Indicates 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. This structured field is sent from the PLU to the SLU.

Format

The table below shows the format of this structured field.

Byte	Content	Content Description
0 - 1	L	Length of structured field
2 - 3	X'1033'	Restart
4	RES	Reserved
5 - 6	Start page	Number of pages to skip on restart
7 - 8	Start line	Number of lines to skip on page for restart
9 - N	SCS data	SCS data (noncompressed and noncompacted) to set up for restart

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. There are three other data fields:

- 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 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 is sent 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

Intermixes SNA Character String (SCS) data with other structured fields. The SCS data portion of the SCS Data structured field is treated the same as SCS data not delimited by an SCS Data structured field.

Format

Byte	Content	Content Description
0-1	L	Length
2	X'41'	SCS Data Structured Field
3	PID	Partition ID ('X'00' - X'7E')
4-n	Data	SCS data

Select Color Table

Function

Selects either the nonloadable or the loadable color table.

Format

The format of this structured field is shown in the table below.

Byte	Content	Content Description
0 - 1	X'0006'	Length of structured field
2 - 3	X'0F04'	Select color table
4 - 5	COMMAND	Command

Note: For information regarding the format and operation of the COMMAND parameter, refer to related graphics documentation.

Select Format Group

Function

Selects the format group that is to be the source of formats in subsequent operations.

The Select Format Group structured field defines, for the specified partition, a single format group that is to be the current source of formats for subsequent Present Absolute and Present Relative Format requests for that partition. The last group selected remains current until it is reset or replaced by a subsequent Select Format Group for that partition. Subsequent Present Absolute and Present Relative Format requests specifying a particular partition are qualified by the "current" format group for that partition.

If a Present Absolute or Present Relative Format request is made prior to selecting a format group, it will be rejected.

Note: If PID is a text partition the structured field is rejected.

Format

Byte	Content	Content Description
0-1	L	Length of structure
2	X'4A'	Select Format Group
3	PID	Partition ID (X'00' - X'7E')
4-n	GRP	Format group

Set Checkpoint Interval

Function

Passes from the host to the terminal the number of pages that are to be in the interval between checkpoints. It contains the recovery data needed by the host to recover from the error.

This structured field resets all previous checkpointed information. The checkpoint counters begin with the first Function Management data byte following the structured field.

Format

Byte	Content	Content Description
0-1	X'0007'	Length of this structure
2-3	X'1032'	Request Recovery Data
4	Reserved	
5-6	INTERVAL	Checkpoint interval - specifies the number of pages in the interval between terminal checkpoints. This number enables checkpointing of jobs that use the short forms without taking an excessive number of checkpoints. A zero value indicates that checkpoints are not to be taken.

Set MSR Control

Function

Allows the application program associated with the host to set the device input states and their associated indicators. In addition, this structured field allows the application program to set user indicators.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structured field
2-3		X'0F01'	Type code for Set MSR
4		PID	Partition identifier
5		TYPE	MSR type = X'01' All other values reserved
6		STATE MASK	Bit i = B'1' selects states to be modified by corresponding bit in byte 7:
	0	USER	User Mode
	1	LOCK	Locked /Unlocked
	2	AUTO	Auto Enter
	3	AI1S	Audible Ind 1 Suppress
	4	AI2S	Audible Ind 2 Suppress
	5-7	RES	Reserved
7	0-7	STATE VALUE	Bit i = B'0' sets state off Bit i = B'1' sets state on
8		IND. MASK	Bit j = B'1' selects corresponding indicator
9		IND. VALUE	Bit j = B'0' sets indicator off Bit j = B'1' sets indicator on

Additional Content Description

This structured field is sent by the application to control the following device states and indicators.

- State Mask Byte 6 (bits 0 – 4)

  Identifies the state of the device. When these bits are set to B'1' the corresponding bit in byte 7 determines whether the state is on or off.

  - USER: This state of the device is set by the application, and the meaning is application determined.
  - LOCK: When this state is set, the device is disabled for input.
  - AUTO ENTER: When this state is set, an input operation on the device will cause an Enter operation from the device, regardless of the type of card read. When this state is reset, an input operation on the device will not cause an Enter operation from the device unless a secure card is read.
  - AUDIBLE IND 1 SUPPRESS: When this state is set, the internally generated audible indication 1 is suppressed.
  - AUDIBLE IND 2 SUPPRESS: When this state is set, the internally generated audible indication 2 is suppressed.

- STATE VALUE Byte 7

  Determines whether the state is on or off. If the corresponding bit in byte 6 is set to B'1', apply the bit in byte 7.

- Byte 8 and 9

  These indicators are set by the application program to convey information to the operator of the device. For the Set MSR Control structured field TYPE X'01', magnetic reader indicators 0–4 are defined. These indicators are reset by the next read to type the correct partition. The device states set by the Set MSR Control structured field are device related, while the magnetic reader indicators are partition related. Therefore, the magnetic reader indicators are saved on a partition basis and only affect the device when the partition is active.

Set Printer Characteristics

Function

Allows the application to control the setting and resetting of certain functions and modes within a control unit terminal (CUT) printer.

The Set Printer Characteristics structured field is valid only in non-SNA environments.

Format

Byte	Bit	Content	Meaning
0−1		L	Length of structure
2−3		X'0F84'	Set Printer Characteristics
4		**Flags**	
	0	RSTALL	Reset All Characteristics
		B'0'	Off (No reset all)
		B'1'	On (Reset all)
	1−7	Reserved	Reserved
5		**Flags**	Reserved

The base may be sent without any self-defining parameters appended when a Reset of all previously set functions is desired without setting any new ones (RSTALL=ON). If the base is sent without any self-defining parameters appended with No Reset All indicated (RSTALL = OFF), it will be treated as a no-op by the printer and a normal response will be returned.

Within a WSF transmission, the SPC must be sent as the first structured field in the transmission. However, if the SPC structured field is used with the Data Chain structured field, it must be sent immediately following a Data Chain (Begin or Only) structured field. If used following a Data Chain (Begin) structured field, and a recoverable error occurs prior to completion of the end-of-chain, the unit of recovery is the chain. An example of recovery would be to retry the chain from the beginning.

The Early Print Complete (EPC) self-defining parameter is used to set an early print complete mode of operation on or off. This parameter provides for enabling/disabling of the printer operator control. (EPC mode may be on or off depending on the setting of the operator control.) The default is printer operator control enabled.

If the RSTALL flag in the base part of the structured field equals Reset, the default is set. However, when the Early Print Complete self-defining parameter is present, the SREPC flag overrides the RSTALL flag.

For example, if RSTALL = Reset and SREPC = EPC on (operator control disabled), the EPC mode will be on and operator control will be disabled.

Early Print Complete Self-Defining Parameter

Function

A self-defining parameter that can be appended to the base portion of this structured field. EPC allows the loading of the printer buffer and the printing operation to overlap, thereby improving throughput to high-speed printers. EPC is supported only in non-SNA environments on 3174 Models 1L, 1R, 51R, and 81R.

Format

Byte	Bit	Content	Meaning
0		'03'	Length of parameter
1		X'01'	Early Print Complete
2		SREPC	Set/Reset Early Print Complete
	0−1	B'00'	Enable operator control
		B'01'	EPC Off − disable operator control
		B'10'	EPC On − disable operator control
		B'11'	Reserved
	2−7	Reserved	Reserved

Prior to use of the SPC structured field, an application should determine the characteristics of the printer by issuing a Read Partition Query List structured field. The Query List structured field must either contain a "Q" code of X'A9' or indicate "ALL."

When EPC is on, the printer will return a positive completion status for the print information received after it is verified but before it is printed. If an error occurs while this information is printing, the error will be reported and associated with print information received later. Because the error is not properly synchronized with print information it pertains to, error handling may be affected.

If an SPC structured field is detected out of sequence, it will be rejected with an Op Check sense code sent to the host.

All set printer characteristics will be reset to default by any of the following:

- Receipt of an Erase Write (EW) or Erase Write Alternate (EWA) command with WCC = Reset
- Power on reset (POR)
- RSTALL=X'1'B.

If a system performs its error handling by reprinting the job, EPC should be turned on at the beginning of the job and turned off with the last piece of print information for the job. This requires the code that manages EPC to recognize job boundaries. In the SNA environment, LU1 may be used to overlap the loading of a printer's buffer with the print operation if the serial load and print required by LU3 results in performance problems.

Set Reply Mode

Function

Defines the format of any subsequent inbound data streams from an alphanumeric partition by defining the values required by the host application.

Subsequent inbound data streams from the named partition sent inbound in response to a read command are in the format defined by the Set Reply Mode structured field. This reply mode remains in effect until set by another Set Reply Mode or until reset by a reset action.

The inbound data stream from a Text partition is not controlled by a Reply Mode. Hence a Set Reply Mode structured field sent to a text partition will be rejected.

Format

Byte	Content	Content Description
0-1	L	Length of this structure
2	X'09'	Set Reply Mode
3	PID	Partition identifier (X'00' through X'7F')
4	MODE X'00' X'01' X'02' Other	 Field mode Extended field mode Character mode Reserved
5-n	ATTR LIST (Present only if mode is X'02')	The type codes for the attribute types that: • Will be transmitted using SA orders if mode X'02' is specified. X'02') • The operator may select from the keyboard. (The operator may select the attribute value to associate with the keyed data.)

Additional Content Description

The values that may appear in each mode are listed below:

- MODE
  - X'00' Field mode:
    1. 3270 orders (SBA, SF)
    2. Characters
    3. Field attributes
    4. Graphic escape
  - X'01' Extended field mode:
    1. 3270 orders (SBA, SF, SFE)
    2. Characters
    3. Field attributes
    4. Extended field attributes
    5. Graphic escape

    In Field mode or Extended Field mode, attribute selection by the operator is not allowed. Any operator selection allowed by a previous Set Reply Mode structured field is reset to the default value.

- X‘02’ Character mode:

  1. 3270 orders (SBA, SF, SFE, SA)
  2. Characters
  3. Field attributes
  4. Extended field attributes
  5. Character attributes listed in Set Reply Mode structured field
  6. Graphic escape

  In Character mode, the listed attribute types may have values selected by the operator. Any operator selection current when Set Reply Mode is interpreted is not reset. Attribute selection by the operator for attribute types not explicitly listed is not allowed.

  Values for Mode not shown here are reserved and will be rejected. Attribute types other than those defined will be rejected.

Set Window Origin

Function

Changes the position of the window within the presentation space of the specified partition.

If the named partition does not exist, an error situation arises. (See Table B-1 on page B-2.)

The window of the specified partition is repositioned to the new origin. If the following condition occurs, the window is not moved and a sense code is returned:

$RW + HW > H$
$CW + WW > W$

where:

HW = Height of window (viewport)
H = Height of presentation space
WW = Width of window (viewport)
W = Width of presentation space

The current cursor position for the named partition is not changed unless the new window origin results in the current cursor position's being outside the window. If outside the window, the current cursor position is moved by the minimum number of row and columns needed to place it in the nearest row or column of the window.

Note: The Modify Partition structured field may also be used to change the window position.

Format

Byte	Content	Content Description
0-1	X'0008'	Length of this structure
2	X'0B'	Set Window Origin
3	PID	Partition identifier (X'00' through X'7E')
4-5	RW	Row position of the window origin
6-7	CW	Column offset of the window origin

Outbound/Inbound Structured Fields

The structured fields described in this section are structured fields that can be transmitted either from the application to the device or from the device to the application. The structured field format is the same both inbound and outbound.

Data Chain

Function

Provides a data-chaining function for non-SNA environments. It identifies the individual transmissions as being part of a single message.

This structured field utilizes the structured field grouping function described in "Structured Field Grouping" on page 5-3 to provide a "chaining" function. Items such as structured fields and control code sequences contained in the message may span transmissions that are related through Data Chain structured fields. This allows a message to be divided into transmissions without any examination/manipulation of the message content. The Data Chain structured field was defined for use in the non-SNA environment. Its use in the SNA environment is unnecessary because of the chaining function provided in SNA.

Format

Byte	Bit	Content	Content Description
0-1		X'0006'	Length of Structure
2-3		X'0F21'	Data Chain
4	0	RES	Reserved
	1-2	GROUP	
		B'00'	Continue
		B'01'	End
		B'10'	Begin
		B'11'	Only
	3-4	INCTRL	Inbound Control
		B'00'	No change
		B'01'	Enable Inbound Data Chaining
		B'10'	Disable Inbound Data Chaining
		B'11'	Reserved
	5-7	RES	Reserved
5		**Flags**	Reserved

Additional Content Description

- **GROUP** - Establishes whether the data following the Data Chain is the beginning, continuation, or the end of a chain of grouped transmissions. The value B'11' (only) indicates the entire message is contained in the transmission that follows the Data Chain structured field.

- **INCTRL** - This parameter is only set in the outbound Data Chain structured field to control how the inbound data is sent. It is not set in the inbound Data Chain structured field and should be zeros.

  An outbound Data Chain structured field with INCTRL = B'01' (Enable Inbound Data Chaining) allows the device to use the Data Chain structured field for inbound transmissions. If inbound data chaining is already enabled, it will remain enabled.

  An outbound Data Chain structured field with INCTRL = B'10' (Disable Inbound Data Chaining) prevents the device from using the Data Chain structured field for inbound transmissions. If inbound data chaining is already disabled, it will

remain disabled until subsequently enabled by an outbound Data Chain structured field with
INCTRL = B'01' (Enable Inbound Data Chaining).

An outbound Data Chain structure field with INCTRL = B'00' (no change) has no effect on the enable/disabled state.

Destination/Origin

Function

The function of this structured field is to identify the destination/origin of structured field in a single session multi-device (workstation) implementation.

Outbound (from the "host") the ID identifies the destination of the structured fields that follow. Inbound (to the "host") the ID identifies the origin of the structured fields that follow.

At the beginning of a transmission the destination/origin is the default (primary display).

Once a Destination/Origin structured field establishes the destination/origin, it applies until either another Destination/Origin structured field establishes a new destination/origin or a new transmission starts.

Format

Byte	Bit	Content	Content Description
0-1		X'0008'	Length of this structure
2-3		X'0F02'	Destination/Origin
4		Flags	
	0-1	INCTRL	Input Control:
		B'00'	Enable input
		B'01'	No change
		B'10'	Disable input
		B'11'	Reserved
	2-7	RES	Reserved
5		Flags	Reserved
6-7		ID	Identifies destination or origin of the structured fields which follow in the data stream

Additional Content Description

- **INCTRL** - Applies only on outbound (to the auxiliary device); on inbound the INCTRL flag is ignored.
  - B'00' - Allows the auxiliary device to send data. If the auxiliary device is already enabled, it will remain enabled.
  - B'01' - A change does not occur in the enabled/disabled status.
  - B'11' - The auxiliary device is not permitted to send data until subsequently enabled by a Destination/Origin structured field with INCTRL flag = B'00' - If the auxiliary device is already disabled, the INCTRL flag = B'10' will cause no change.

  If a Destination/Origin structured field is directed to the base display (ID = X'0000'), the INCTRL flag applies on a "global" basis: all the auxiliary devices supported are enabled/disabled/unchanged as a group.

  Note: There is one exception where an auxiliary device may send input without being enabled. An Exception Condition structured field, reporting unavailability of the auxiliary device, may be sent in reply to a Destination/Origin structured field sequence attempting to use the auxiliary device.

- **DOID** - The valid values for the DOID are:
  - X'0000' (permanently assigned to the primary display).
  - All ID values returned in the Query Replies for Auxiliary devices.

  All other values are invalid and must be rejected.

Object Control

Function

Carries the Object Data between the device and the host.

It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the Object Data.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structured field
2-3		X'0F11'	Object Control
4		PID	Partition identifier
5		Flags	
	0-1	SPANF	Spanning flag (outbound)
		B'00'	Not first and not last (middle)
		B'01'	Last but not first
		B'10'	First but not last
		B'11'	First and last (only)
	2-3	MODE	Control interpretation mode (outbound)
		B'00'	Immediate mode
		B'10'	Store mode
		OTHER	Reserved
	4-7	RES	Reserved
6		OBJTYP	Object type
		X'00'	Graphics
		X'01'	Image
7-n		DATA	Data appropriate to the object type. See the appropriate object publications.

Additional Content Description

- **SPANF** - Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.
- **MODE** - Specifies how the Object Data contained in the DATA parameter (byte 7 – n) are to be interpreted in the device. This flag is applicable to outbound transmissions only.
  - *B'00'* - Immediate mode: for graphics, data units contained within the DATA parameter will be stored if the device has this capability; otherwise they are rejected.
  - *B'10'* - Store mode: for graphics, data units contained within the DATA parameter will be stored if the device has this capability; otherwise it will be rejected.
- **OBJTYP** - Contains the object type being carried in this structured field.
- **DATA** - Contains the self-describing Object Data commands. There can be more than one entity of an object type specified. Entities of different object

types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

Note: For the format and contents of the DATA parameter, refer to the appropriate graphics or image product publications.

Object Data

Function

Carries the object data between the device and the host.

It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the object data.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structured field
2-3		X'0F0F'	Object Data
4		PID	Partition identifier
5		Flags	
	0-1	SPANF	Spanning flag (outbound)
		B'00'	Not first and not last (middle)
		B'01'	Last but not first
		B'10'	First but not last
		B'11'	First and last (only)
	2-3	MODE	Control interpretation mode (outbound)
		B'00'	Immediate mode
		B'10'	Store mode
		OTHER	Reserved
	4-7	RES	Reserved
6		OBJTYP	Object type
		X'00'	Graphics
		X'01'	Image
7-n		DATA	Data appropriate to the object type. See the appropriate object publications.

Additional Content Description

- **SPANF** - Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.
- **MODE** - Specifies how the object data contained in the DATA parameter (byte 7 – n) are to be interpreted in the device. This flag is applicable to outbound transmissions only.
  - *B'00'* - Immediate mode: for graphics, data units contained within the DATA parameter will be stored if the device has this capability, otherwise they are rejected.
  - *B'10'* - Store mode: for graphics, data units contained within the DATA parameter will be stored if the device has this capability, otherwise it will be rejected.
- **OBJTYP** - Contains the object type being carried in this structured field.
- **DATA** - Contains the self-describing object data commands. There can be more than one entity of an object type specified. Entities of different object

types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

Note: For the format and content of the DATA parameter, refer to the appropriate graphics or image product publications.

Object Picture

Function

Carries the object pictures between the device and the host.

It is used both inbound and outbound. When used outbound, it indicates the mode of interpretation for the object pictures.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structured field
2-3		X'0F10'	Object Picture
4		PID	Partition identifier
5		Flags	
	0-1	SPANF	Spanning flag (outbound)
		B'00'	Not first and not last (middle)
		B'01'	Last but not first
		B'10'	First but not last
		B'11'	First and last (only)
	2-3	MODE	Interpretation mode (outbound)
		B'00'	Immediate mode
		B'10'	Store mode
		B'01'	Reserved
		B'11'	Store and draw mode.
	4-7	RES	Reserved
6		OBJTYP	Object type
		X'00'	Graphics
		X'01'	Image
7-n		DATA	Data appropriate to the object type. See the appropriate object publications.

Additional Content Description

- **SPANF** - Permits the segmentation of a structured field into several related structured fields of the same type. The rule for spanning is that the DATA parameters of structured fields of that type are logically contiguous. These indicators are applicable to both inbound and outbound transmissions.
- **MODE** - Specifies how the object pictures contained in the DATA parameter (byte 7 − n) are to be interpreted in the device. This flag is applicable to outbound transmissions only.
  - *B'00'* - Immediate mode: for graphics, each chained segment will be executed immediately as it is received and not stored.

    Non-chained segments and extended drawing routines will be stored if the device has this capability, otherwise they are rejected.
  - *B'10'* - Store Mode: each object picture contained within the DATA parameter will be stored if the device has this capability, otherwise it will be rejected. No execution will occur.

- *B'11'* - Store and draw mode: for graphics, each segment and extended drawing routine contained within the DATA parameter will be stored if the device has this capability, otherwise it will be rejected. Additionally, chained segments will be executed.

- **OBJTYP** - Contains the object type being carried in this structured field.
- **DATA** - Contains the self-describing object picture commands. There can be more than one entity of an object type specified. Entities of different object types must not be mixed in this parameter, since it is interpreted by the object type and therefore correct interpretation of mixed objects cannot be guaranteed.

  Note: For the format and content of the DATA parameter, refer to the appropriate graphics or image product publications.

OEM Data

Function

Carries the non-IBM-defined data (or a "value added" version) to or from another equipment manufacturer's (OEM) auxiliary device. An OEM device is defined here as a device that is manufactured outside of IBM. The device may carry either an outside manufacturer logo or an IBM logo.

The data to or from an auxiliary device on a 3270 data stream workstation must be in structured field form. The OEM Data structured field provides a means of carrying the manufacturer's defined data (or a "value added" version) to or from an OEM auxiliary device.

Optional Parameters

All parameters through byte 6 must be present. Although there will usually be data in bytes 7 – n, an OEM Data structured field of length 7 is valid, for example, it could be used to end a group when there is no more data available to send.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of Structure
2-3		X'0F1F'	OEM Data
4		RES	Reserved
5	0-1	GROUP	
	B'00'		Continue
	B'01'		End
	B'10'		Begin
	B'11'		Only
	2-7	RES	Reserved for flags
6		Flags	Reserved
7-n		DATA	Data

Additional Content Description

- **GROUP** - Provides a "spanning" function; that is, it allows data to be divided into two or more OEM Data structured fields without regard to control sequence boundaries.
- **DATA** - The OEM Aux Device query reply returned for a particular OEM auxiliary device will indicate whether the data in the OEM Data structured field contains the manufacturer-defined data stream or a registered "value added" version of the manufacturer-defined data stream.

Save/Restore Format

Function

Used by the display to send the current format parameter control block in reply to a Save/Restore Format Outbound structured field specifying primary save.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2-3		X'1034'	Save/Restore Format
4		Flags	
	0	SAVE/RES	
		B'0'	Save the current format parameter control block (FPCB).
		B'1'	Restore saved FPCB.
	1	SEC/PRI	
		B'0'	FPCB saved at secondary
		B'1'	Format parameter control block saved at primary
	2-7	RES	Reserved
5-n		FPCB	Contents of the format parameter control block that is to be saved or restored

Additional Content Description

- Flags

  The Save/Restore flag (byte 4, bit 0) must be set to B'0' and the Secondary/Primary flag (byte 4, bit 1) must be set to B'1'. Other values are invalid.

- FPCB

  The Format parameter control block is a device-dependent control block that contains formatting parameters established by power-on defaults or subsequently received SNA Character String controls such as:

  - Set Horizontal Format (SHF)
  - Set Vertical Format (SVF)
  - Set Line Density (SLD)
  - Set Print Density (SPD).

  The display has a current format parameter control block and one format parameter control block save area. There are no constraints on the number of save areas in the primary.

The format parameter control block must contain all of the information necessary to save/restore all format parameters which the display supports. After a save/restore operation of any given FPCB, either at the display or the host, the format parameters in effect will be precisely those in effect before the save/restore operation. The save/restore operation itself causes no movement of the current position in the presentation space — that is, the presentation space position is not saved/restored.

At power-on time, the display will initialize the save area to the default values of the device.

Select IPDS Mode

Function

Selects the Intelligent Printer Data Stream (IPDS) mode when IPDS is supported in 3270 non-SNA.

Refer to your product publications for information on how the data is processed when IPDS is selected.

Format

Byte	Content	Content Description
0-1	X'0006'	Length of Structure
2-3	X'0F83'	Select IPDS Mode
4-5	Flags	Reserved

Chapter 6. Inbound Structured Fields

Introduction

This chapter contains the inbound structured fields, including all of the query reply structured fields. These Inbound structured fields are structured fields that are only transmitted one way. That is, they are sent from the device to the host application. Inbound structured fields are preceded by an AID of X'88'.

(AID) X'88'	Structured Field

The structured fields in this chapter are discussed in alphabetical order with all of the query replies also arranged alphabetically after the query reply structured field.

Inbound Structured Fields

ID	Name
80	Inbound 3270DS
81	Query Reply

84 Query Reply (Alpha Partitions)
9D Query Reply (Anomaly)
99 Query Reply (AUXDA)
9F Query Reply (Begin/End of File)
85 Query Reply (Character Sets)
86 Query Reply (Color)
AB Query Reply (Cooperative Processing Requestor)
98 Query Reply (Data Chaining)
A2 Query Reply (Data Streams)
A0 Query Reply (Device Characteristics)
97 Query Reply (Document Interchange Architecture (DIA)
91 Query Reply (DBCS-Asia)
95 Query Reply (DDM)
B5 Query Reply (Extended Drawing Routine)
8A Query Reply (Field Validation)
8C Query Reply (Field Outlining)
90 Query Reply (Format Presentation)
B4 Query Reply (Graphic Color)
B6 Query Reply (Graphic Symbol Sets)
87 Query Reply (Highlight)
82 Query Reply (Image)
A6 Query Reply (Implicit Partitions)
AA Query Reply (IOCA Auxiliary Device)
B2 Query Reply (Line Type)
8B Query Reply (MSR Control)
FF Query Reply (Null)
8F Query Reply (OEM Auxiliary Device)
A7 Query Reply (Paper Feed Techniques)

B3 Query Reply (Port)
9C Query Reply (Product Defined Data Stream)
B1 Query Reply (Procedure)
88 Query Reply (Reply Modes)
A1 Query Reply (RPQ Names)
92 Query Reply (Save/Restore Format)
B0 Query Reply (Segment)
A9 Query Reply (Settable Printer Characteristics)
96 Query Reply (Storage Pools)
80 Query Reply (Summary)
83 Query Reply (Text Partitions)
A8 Query Reply (Transparency)
81 Query Reply (Usable Area)
9A Query Reply (3270IPDS)

0F22 Exception/Status
0FB1 Inbound Text Header
0FC1 Type 1 Text Inbound
1031 Recovery Data

Structured-field type codes not listed are rejected. Unless specifically stated to the contrary, any bits/fields classified as reserved must be checked for zero value; nonzero values are rejected.

Exception/Status

Function

Allows the reporting of exception/status information at the application level.

All parameters shown for the base structured field must be present. The structured field is allowed to carry only one exception/status condition. **Either**, but not both, of the self-defining parameters must also be present.

A length value of X'0000' is invalid.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of structured field
2-3		X'0F22'	Exception/status
4		PID	Partition identifier
5		Flags	
	0	AVAILSTAT	Availability status:
		B'0'	Status not provided
		B'1'	Status provided
	1-7	RES	Reserved
6		Flags	Reserved

Additional Content Description

- **PID** - Provides the partition identifier (PID) of the partition associated with the reported exception condition or status. When used for reporting an exception condition or status for a direct accessed auxiliary device, the Exception/Status structured field must be preceded by a Destination/Origin structured field. Also, when reporting an exception condition or status of a direct accessed auxiliary device, the PID parameter in the Exception/Status structured field has no meaning and should be ignored. The PID should be set to X'FF'.

- **AVAILSTAT** - Indicates whether the exception condition being reported requires an availability status when the reported exception clears. This is set to B'0' when the exception code definition does not require a subsequent availability status. It is set to B'1' when the definition of the exception code being carried by the Exception/Status structured field requires an availability status when the condition clears.

Self-Defining Parameters

Auxiliary Device Exception Condition

Function

Reports exception conditions associated with a direct accessed auxiliary device. Although the error codes may have the same value as existing SNA defined codes, the meanings are not necessarily the same.

Format

Byte	Bit	Content	Content Description
0		X'06'	Length of parameter
1		X'01'	Auxiliary Device Exception Condition
2-3		Reserved	Reserved
4-5		EXCODE	The specific exception code. It defines the direct accessed auxiliary exception condition.

Auxiliary Device Status

Function

Reports a change of status. The sending of certain exception codes that indicate unavailability requires that an available status be sent when the device becomes available.

Format

Byte	Bit	Content	Content Description
0		X'04'	Length of parameter
1		X'02'	Auxiliary Device Status
2-3		STATCODE	Specific Status Code

Inbound Text Header

Function

Transmitted inbound from a text partition. It may be followed by a Type 1 Text Inbound structured field.

A single Inbound Text Header structured field is transmitted inbound from a text partition (A-MODE = X'2'):

1. In reply to a Read Modified command when the device is in a retry state and INPID identifies the text partition
2. For an operator enter action, when the active partition is a text partition.

Format

Byte	Content	Content Description
0-1	X'0016'	Length of structure
2-3	X'0FB1'	Inbound Text Header
4	PID	Partition ID
5	AID	Attention identifier
6	RES	Reserved
7	RES	Reserved
8	RES	Reserved
9	LVL	Cursor level
10-11	CRO	Cursor row offset
12-13	CC	Cursor column offset
14-15	RW	Row offset of window origin
16-17	CW	Column offset of window origin
18-19	HW	Window height
20-21	WW	Window width

Additional Content Description

- **LVL** - The LVL parameter specifies the level of the element at the presentation space position (CRO,CC) that is identified. (The first element is at level 1.)

  When the element level can be determined by the device, it should return a value for LVL in the range 1 through 254. If the device does not identify the level explicitly, it will return value X'FF' for LVL, meaning all elements at position (CRO,CC).

  It is not required that a device have the capability of distinguishing between multiple elements at the same position. That is, a device could *always* return LVL = X'FF' for a non-empty presentation position.

- **CRO, CC** - The (CRO,CC) parameters specify the presentation space position of the cursor. Note that these values can range from 0 through 65534 (that is, the value X'FFFF' will not be returned inbound).

  If position (CRO,CC) is empty, LVL will be X'00'.

- **RW, CW** - RW and CW give the origin of the window on presentation space.
- **HW, WW** - HW and WW give the height and width of this window. Note that, in a text partition, the operator may change the size of this window from the initial values established by the Create Partition structured field.

Inbound 3270DS

Function

This structured field is used to transmit 3270 orders and data inbound from a named alphanumeric partition.

This structured field is transmitted inbound from an alphanumeric partition (Create Partition Structured Field — A-MODE = B'0000' or B'0001'):

1. In reply to a Read Modified, Read Modified All (SNA only), or Read Buffer command, when the inbound partition has a nonzero PID and the device is in Retry state
2. In reply to a Read Partition structured field, type X'F6', X'6E' or X'F2', addressed to a partition with a nonzero PID.
3. For an operator Enter action, when the Active partition has a nonzero PID

Orders and data from partition zero (whether implicitly or explicitly created) are transmitted in 3270 format, not in this structured field.

Format

Byte	Content	Content Description
0-1	L	Length of structure
2	X'80'	Inbound 3270DS
3	PID	Partition identifier
4	AID	Attention identifier
5-6	CURSOR	Cursor address
7-n	DATA	Orders and data

Recovery Data

Function

This structured field is sent from the SLU to the 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

The following table shows the format of this structured field.

Byte	Bit	Content	Content Description
0 - 1		X'0031'	Length of this structured field
2 - 3		X'1031'	Recovery data
4		RES	Reserved
5		**Flags**	
	0	B'0'	Vertical format not needed on Restart
		B'1'	Vertical format needed on Restart
	1	B'0'	Horizontal format not needed on Restart
		B'1'	Horizontal format needed on Restart
	2 - 7	RES	Reserved
6		SLD	SLD - (Set line density) parameter in effect at the checkpoint.
7		Char. set	Character set parameter of Set Attribute control in effect at the checkpoint.
8 - 11		Vertical	Byte offset from Checkpoint interval structured field to the Set Vertical Format control in effect for the checkpoint.
12 - 13		V-offset	Byte offset within the string control byte string or the Set Vertical Format character
14 - 15		V-sequence	RU sequence number
16 - 17		V-length	Length of the Set Vertical Format character string required for restart.
18 - 19		SPD	Set Primary Density parameter in effect at the checkpoint
20 - 23		Horizon	Same as vertical for Set Horizontal Format (SHF)
24 - 25		H-offset	Same as V-offset for SHF
26 - 27		H-sequence	Same as V-sequence for SHF
28 - 29		H-length	Same as V-length for SHF
30		Color	Color parameter of Set Attribute in effect at the checkpoint.
31		Hilite	Highlight parameter of Set Attribute in effect at the checkpoint.
32 - 33		Pages	Number of pages printed since the checkpoint.

Byte	Bit	Content	Content Description
34 - 35		Lines	Number of lines printed on the page with the error.
36 - 39		Checkpoint	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.
40 - 41		C-offset	Byte offset within the String Control Byte string or structured field of the checkpointed character
42 - 43		C-sequence	RU sequence number of the RU containing the checkpoint 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.
48		Prime	Prime compression character

Additional Content Description

- **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 W-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 Set Vertical Field (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 Set Horizontal Field (SHF).

- **SLD** – Set Line Density is a 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 changed since the checkpoint. The default value (X'00") indicates the parameter is not to be reinitialized at restart.

- **Char. Set.** – Character Set 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. The default value (X'00') indicates the parameter is not to be reinitialized at restart.

- **Vertical** – Vertical is a 4-byte 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. 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).

- **W-offset** – W-offset is a 2-byte 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 if the vertical field points to the actual SWF code or if the SVF character is not within a SCB string or structured field.

- **W-sequence** – This contains the RU sequence number of the RU that contains the SVF control code in effect at the checkpoint.

- **W-length** – This 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.
- **SPD** – Set Print Density 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. The default value (X'0000') indicates the parameter is not to be reinitialized at restart.
- **Horizon** – This is a 4-byte binary counter. It indicates that the byte offset from the Set Checkpoint Interval structured field to the SHF code point is in effect for this checkpoint. If no SHF has been received, this field is set to zero. 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 if the Horizontal Field points to the actual SHF code or if the SHF character is not within an SCB string or structured field.
- **H-sequence** – H-sequence contains the RU sequence number of the RU that contains the SHF control code in effect at the checkpoint.
- **H-length** – H-length 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.
- **Color** – Color 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. The default value (X'00') indicates the parameter is not to be initialized at restart.
- **Hilite** – This is the 1-byte parameter field of the SA value for highlighting in effect at the checkpoint. The default SA value (Xosq.00') is not supported or has not changed since the checkpoint. The default value (X'00') indicates the parameter is not to be initialized at restart.
- **Pages** – Pages indicates the number of pages that the SLU has printed since the checkpoint that is being sent to the PLU was taken.
- **Lines** – Lines indicates the number of lines that the SLU has printed since the beginning of the page on which the error had been detected.
- **Checkpoint** – Checkpoint is a 4-byte binary counter that indicates a certain number of FM data bytes. These bytes are from the Set Checkpoint Interval structured field to the first code point after the one causing 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. If the character is within a parameterized SCS control code, this is the offset to the SCS control

code. This field is zero if the checkpoint field points to the exact checkpointed character or if the checkpointer is not within an SCB string or structured field.

- **C-seq** — C-seq contains the RU sequence number of the RU containing the checkpointed character.
- **C-seqoff** — C-seqoff contains the byte offset within the RU of the checkpointed character.
- **C-SCSoff** — This 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 if the checkpointed character is not within a parameterized SCS code.
- **Prime** — Prime is the prime compression character in effect at the time of checkpoint.

Type 1 Text Inbound

Function

Transmits data from a text partition.

The Type 1 Text Inbound structured field is transmitted inbound from a text partition (A-MODE = X'2') following an Inbound Text Header structured field if:

1. The AID corresponds to a read modified operation, and
2. The MDT is set because the operator modified data.

Note that the Type 1 Text Inbound structured field is *not* transmitted:

1. For a "short read" (PA key or Clear Partition), or
2. If the MDT is not set.

Format

Byte	Content	Content Description
0-1	L	Length of structure
2-3	X'0FC1'	Type 1 Text Inbound
4	PID	Partition ID
5-6	RES	Reserved
7-n	DATA	Text data

Additional Content Description

- **DATA** - consists of the entire contents of the text buffer, with an embedded insert cursor control to identify the cursor position. The data stream position of insert cursor should be interpreted in combination with the LVL parameter in the Inbound Text Header, as explained below.

  1. If LVL is X'01' through X'FE', the Insert cursor control immediately precedes the element at position [CRO, CC, LVL].
  2. If LVL = X'FF' (that is, all elements at [CRO,CC]), the insert cursor control immediately precedes the first element in the data stream that is at a (row, column) at or beyond (CRO,CC)..
  3. If LVL = X'00' (that is, [CRO,CC] is empty), the insert cursor control immediately precedes the first element in the data stream (if any) that is at a (row, column) beyond (CRO,CC). If there is no such element, insert cursor is at the end of the data stream.

Query Replies

The Query Reply structured field is used to report to the host application the functions supported by the 3270 data stream devices and information about these functions.

All Query Reply structured fields have a value of X'81' in byte 2 that identifies the structured field as a Query Reply. The QCODE parameter (byte 3) is a unique value assigned to each Query Reply.

The Read Partition structured is used for soliciting query replies when the TYPE parameter has a value of either X'02' (Query) or X'03' (Query List). The PID parameter (byte 3) has a value of X'FF' (PID not applicable). If the TYPE parameter value is X'03' (Query List), the Read Partition structured field contains an additional parameter, REQTYP (Request Type), bits 0-1 of byte 5 and optionally a list of QCODES starting at byte 6. The Request Type parameter is used to indicate the type of Query List operation:

- B'00' = QCODE List (QCODEs are listed for the Query Replies requested)
- B'01' = Equivalent (with or without a QCODE List)
- B'10' = All.

The query replies are returned in response to a Query or Query List as indicated in the Query Response Table.

Query

A Read Partition structured field with a TYPE parameter (byte 4) value of X'02' is defined as a Query.

Refer to the Query Response Table for the query replies that a 3270 data stream device or workstation supports that must be returned in reply to a Query. Those not indicated are not returned in reply to a Query.

Query List

A Read Partition structured field with a TYPE parameter (byte 5) value of X'03' is defined as a Query List.

There are three versions of Query List. The version is designated by the Request Type parameter (bits 0-1, byte 5) value:

- **QCODE List** (B'00') - The Query List = (QCODE List) contains a list of one or more Query Reply QCODES. The 3270 data stream device or workstation returns all the requested Query Replies (QCODES listed) that are supported. If none of the requested Query Replies are supported, a Null query reply is returned. There is no requirement as to the order of the QCODES in the Read Partition (Query List) structured field or in the order that the requested query replies are returned by the 3270 data stream device or workstation.

  It is not invalid for a particular QCODE to appear more than once in the list. However, regardless of how many times it appears, the 3270 device or workstation does not return duplicate query replies.

- **Equivalent (B'01')** - Query List = (Equivalent) requests the 3270 device or workstation to return the same Query Replies that would be returned in reply to a Query. Optionally a list (of QCODES) may also be included. Requests for the same QCODE are made within the list, or a QCODE within the list would be returned for the Query Equivalent, no duplicate Query Replies will be returned by the 3270 data stream device or workstation.

- **All (B'10')** - Query List = (All) requests the 3270 data stream device or workstation to return all the Query Replies supported. The Query List = (All) may contain a QCODE list. However, the QCODE list is ignored by the 3270 data stream device or workstation.

Therefore, although duplicate Query Replies may be requested by the Query List = (QCODE List, Equivalent, or All) the 3270 data stream device or workstation will not return duplicate Query Replies. Note that there are cases where a single request (for example, QCODE in Query List) can result in multiple Query Replies being returned. For example, a 3270 data stream workstation could support two other equipment manufacture (OEM) auxiliary devices — plotters, for instance. A Query List request for the OEM AUX Device query reply would result in two OEM AUX Device query replies being returned, one for each device. However, although the two Query Replies are of the same type (OEM AUX Device), the contents of the Query Replies differ (at least in the DOID parameter) and hence are not "duplicate" Query Replies.

All 3270 data stream devices or workstations must support the Summary query reply, QCODE = X'80'. This query reply is returned in reply to a Query or Query List (QCODE List, Equivalent, or All). The Summary query reply provides a list of the QCODEs of *all* the query replies supported by the 3270 data stream device or workstation. (The QCODE for the Summary query reply itself is also included in the list.) The Summary query reply provides the host with the only indication of support of functions where the associated query reply is returned in reply to a Query List = (QCODE List or All).

Query Response Table

	Returned in Response to a Read Partition:			
	Query	Query List		
Query Reply		List	Equivalent	All
Alphanumeric Partitions	Yes	X'84'	Yes	Yes
Auxiliary Devices	Yes	X'99'	Yes	Yes
Begin/End of File	Yes	X'9F'	Yes	Yes
Character Sets	Yes	X'85'	Yes	Yes
Color	Yes	X'86'	Yes	Yes
Cooperative Proc. Requestor	Yes	X'AB'	Yes	Yes
Data Chaining	Yes	X'98'	Yes	Yes
Data Streams	Yes	X'A2'	Yes	Yes
DBCS Asia	Yes	X'91'	Yes	Yes
DDM	Yes	X'95'	Yes	Yes
Device Characteristics	Yes	X'A0'	Yes	Yes
DIA	Yes	X'97'	Yes	Yes
Extended Drawing Routing	No	X'B5'	No	Yes
Field Outlining	Yes	X'8C'	Yes	Yes
Field Validation	Yes	X'8A'	Yes	Yes
Format Presentation	Yes	X'90'	Yes	Yes
Graphic Color	No	X'B4'	No	Yes
Graphic Symbol Sets	No	X'B6'	No	Yes
Highlighting	Yes	X'87'	Yes	Yes
Implicit Partition	Yes	X'A6'	Yes	Yes
IOCA Auxiliary Device	No	X'AA'	No	Yes
Line Type	No	X'B2'	No	Yes
MSR Control	Yes	X'8B'	Yes	Yes
Null	No	X'FF'	No	No
OEM AUX Devices	No	X'8F'	No	Yes
Paper Feed Techniques	Yes	X'A7'	Yes	Yes
Partition Characteristics	No	X'BE'	No	Yes
Port	No	X'B3'	No	Yes
Procedure	No	X'B1'	No	Yes
Product Defined Data Stream	No	X'9C'	No	Yes
Reply Modes	Yes	X'88'	Yes	Yes
Settable Printer Characteristics	No	X'A9'	No	Yes
RPQ Names	Yes	X'A1'	Yes	Yes
Save/Restore Format	No	X'92'	No	Yes
Segment	No	X'B0'	No	Yes

	Returned in Response to a Read Partition:			
	Query	Query List		
Query Reply		**List**	**Equivalent**	**All**
Storage Pools	No	X'96'	No	Yes
Summary	Yes	X'80'	Yes	Yes
Text Partitions	Yes	X'83'	Yes	Yes
Transparency	No	X'A8'	No	Yes
Usable Area	Yes	X'81'	Yes	Yes
3270IPDS	No	X'9A'	Yes	Yes

Query Reply

Function

Query requests are contained in the Read Partition structured field.

In reply to a Query, the device transmits inbound a set of structured fields that describe the device features. Only the query replies that reflect the functions and features supported by the device are transmitted inbound.

Each query reply has an ID of X'81nn', where nn is the QCODE. When a query reply is used in the 3270 data stream, it is preceded by an AID of X'88'. If the structured field is one of a set of Query Reply structured fields, only the first is preceded by an AID of X'88'.

Format

Byte	Content	Content Description
0-1	L	Length of this structure
2	X'81'	Query Reply
3	QCODE	The code identifying the feature being described.
4-n	PLIST	A variable length parameter list. See the specific query reply in this chapter for the parameter list for each QCODE.

Query Reply (Alphanumeric Partitions)

Function

Transmits the maximum number of partitions that can be supported on this device and the device's ability to support partition-related functions.

When this function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'84'	Alphanumeric Partitions
4		NA	Max number of alphanumeric partitions
5-6		M	Total available partition storage
7		Flags	
	0	SCROLL	
		B'0'	Vertical scrolling *not* supported.
		B'1'	Vertical scrolling supported.
	1-2	RES	Reserved.
	3	AP	
		B'0'	All Points addressability *not* supported.
		B'1'	All Points addressability supported.
	4	PROT	
		B'0'	Partition protection *not* supported.
		B'1'	Partition protection supported.
	5	COPY	
		B'0'	Presentation space local copy *not* supported.
		B'1'	Presentation space local copy supported.
	6	MODIFY	
		B'0'	Modify Partition *not* supported.
		B'1'	Modify Partition supported.
	7	RES	Reserved.

Additional Content Description

- **NA** - specifies the number of alphanumeric partitions supported.

  If NA > 0, the alphanumeric partitions may be named 0 through N-1, where N is the total number of partitions supported.

  If NA = 0, one and only one alphanumeric partition may be created via Create Partition. The identifier of this partition must be 0. All outbound and inbound operations to this partition must be via 3270 commands (that is EW, EWA, EAU, RM, RMA [SNA only], RB).

- **M** - defines the total number of bytes of storage available for creating alphanumeric partitions. The value of X'FFFF' is reserved.

  Refer to CO, RO, CM, and FO for allocation of M.

- **FLAGS** - The flag bits specify device support of partition related functions, and are associated with parameters in the Create Partition structured field. A zero value indicates that the associated function is not supported.

  - **SCROLL** - indicates whether the device supports vertical scrolling. Support of scrolling implies:

    1. Support for Create Partition with presentation space height greater than viewport height
    2. Support for Set window Origin with a change of the column (row)
    3. Possible support of local window movement by the operator; thus the operator may move the window from the host-specified position.

  - **AP** - indicates that the Create Partition structured field parameters can be specified in terms of addressable points.

  - **MODIFY** - indicates whether the device supports the Modify Partition structured field. If the vertical scroll flag is also set, Modify Partition may be used to change the window row.

Buffer Allocation Self-Defining Parameter

The self-defining parameters must be included as part of the Alphanumeric Partitions Query Reply unless bytes 3-6 (RO, CO, FO) are all zeros and byte 2 (CM) = 1. The default value for parameters not sent is zero.

A partition size, as defined by (HxW) of the Create Partition structured field, requires storage of:

CM(H x W) + (RO x H) + (CO x W) + FO bytes.

The total storage for all created partitions must not exceed M. A Create Partition structured field that causes the total storage used to exceed M will be rejected.

Format

Byte	Content	Content Description
0	L	Length of this structure
1	X'02'	Buffer allocation parameters
2	CM	Character multiplier - number of storage bytes required for each position defined by (HxW) of the Create Partition structured field
3	RO	Row overhead - bytes of storage required per partition row
4	CO	Column overhead - bytes of storage required per partition column
5-6	FO	Fixed overhead - fixed storage bytes required per partition in addition to RO and CO

Query Reply (Auxiliary Device)

Function

This query reply indicates direct access support of one or more auxiliary devices.

When a 3270 Data Stream Work station function is supported (that is, support of one or more auxiliary devices) this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'99', Equivalent, or All) or to a Query.

This query reply indicates support of:

- Destination/Origin structured field
- Read Partition structured field specifying Query List
- One or more auxiliary devices.

Format

Byte	Content	Content Description
0-1	X'0006'	Length of this structure
2	X'81'	Query Reply
3	X'99'	AUXDA
4-5	FLAGS	Reserved

Query Reply (Character Sets)

Function

Transmits information about:

1. Each character set supported
2. The ability of the device to support the Load Programmed Symbols (Load PS) structured field and the Graphic Escape (GE) character.

This query reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

This form of the query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List=X'85', Equivalent, or All).

This query reply consists of at least two parts: the base and a descriptor. Each is discussed below.

Character Sets Query Reply Base

Function

This portion of the Character Sets Query Reply tells the host application which loadable and/or nonloadable character sets are available at the device. The base is always required and has requirements of its own.

Requirements

The GF flag in the base must be set to B'1'. The CGCSGID field must be present. The first 3 bytes of the descriptor must be present.

Format

The table below shows the format of these parameters.

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'85'	Character Sets Query Reply
4-N		**Flags**	
	0	ALT	
		B'0'	Graphic Escape not supported
		B'1'	Graphic Escape supported
	1	MULTID	
		B'0'	Multiple LCIDs are not supported
		B'1'	Multiple LCIDs are supported
	2	LOADABLE	
		B'0'	LOAD PS are not supported
		B'1'	LOAD PS are supported

Byte	Bit	Content	Content Description
4 cont	3	EXT	
		B'0'	LOAD PS EXTENDED is not supported
		B'1'	LOAD PS EXTENDED is supported
	4	MS	
		B'0'	Only one character slot size is supported
		B'1'	More than one size of character slot are supported
	5	CH2	
		B'0'	2-byte coded character sets are not supported
		B'1'	2-byte coded character sets are supported
	6	GF	
		B'0'	CGCSGID is not present
		B'1'	CGCSGID is present
	7	RES	Reserved
5		**Flags**	
	0	RES	Reserved
	1	PSCS	Programmed Symbols Character Slot
		B'0'	Load PS slot size match required
		B'1'	Load PS slot size match not required
	2	RES	Reserved
	3-7	RES	Reserved
6		SDW	Default character slot width
7		SDH	Default character slot height
8-11		FORM	Supported LOAD PS format types, bit encoded. Bit i = 1 means type i is supported.
12		DL	Length of each descriptor

Additional Content Description

- **ALT** - Indicates that the device can support an alternate character set through the use of the GE control character.
- **MULTID** - Indicates that the device supports multiple graphic symbol sets with the same LCID (see the MULTID parameter of the Load PS structured field on page 5-19).
- **LOADABLE** - Indicates that the Load Programmed Symbol Set structured field is supported. In some implementations, the Load PS Set structured field may be supported while one or more of the attached devices do not support any loadable character sets (for example, a control unit with multiple devices attached). Consequently, the Character Sets Query Reply associated with a particular device may indicate support of the Load PS Set structured field, but only nonloadable character sets are supported. Therefore, to indicate support of loadable character sets, the Character Sets Query Reply for a device must indicate the support of both the Load PS Set structured field *and* support of one or more loadable character sets.

- **EXT** - Indicates that the device supports the Load PS extensions. Thus EXT can be set only if LOADABLE is set.

- **MS** - Indicates whether the device supports a character set related character slot size or not. This parameter can take one of two values:

  - If MS = B'0', the character slot size for *all* character sets is given by the parameters SDW and SDH. If any values are specified in the character set descriptors for SW or SH, they are ignored.
  - If MS = B'1', each character set has its own character slot size, given by the parameters SW and SH in the descriptor for that character set. The parameters SW and SH are present in all descriptors. However, if the value of either SW or SH for a given character set is zero, the values given by SDW and SDH will be used.

  If the device supports non-matrix characters (as indicated in byte 5, bit 1 of the Usable Area Query Reply), bytes 6 and 7 (SDW and SDH) are not applicable and must be set to zero.

- **CH2** - Indicates whether or not 2-byte character sets are supported. In a 2-byte character set, the characters are represented by 2-byte codes in the data stream.

  - If CH = B'0', 2-byte character sets are not supported and the SUBSN parameters are not present in the descriptors.

  - If CH = B'1', 2-byte character sets are supported and the SUBSN parameters are present in the descriptors.

- **GF** - Indicates that the device returns Coded Graphic Character Set Global identifiers in this query reply. This parameter remains to accommodate some existing implementations that support the Character Sets Query Reply but not CGCSGID. All new implementations must have the CGCSGID present and have GF set to B'1'.

- **PSCS** - Indicates whether or not the character size (LW/LH) specified in the Load PS Sets structured field must exactly match the character slot size (SDW/SDH or SW/SH) specified in the Character Sets Query Reply.

  - If PSCS = B'0', a Load PS Sets structured field specifying other than an exact match will be rejected.

  - If PSCS = B'1', a Load PS Sets structured field specifying LW equal to or less than SW/SDW and an LH equal to or less than SH/SDH will be accepted.

  Regardless of the PSCS flag setting, if either or both the width or height of the character size specified in the Load PS Sets structured field exceeds the corresponding dimension of the character slot specified in the Character Sets Query Reply, the Load PS Sets structured field will be rejected. Support of a Load PS Sets character size smaller than the character slot size (bit = B'1') is an implementation/requirement option. The PSCS flag has no meaning for an implementation that does not support Load PS Sets/Loadable Symbol Sets. The flag is set to B'0'.

  When supported, a valid Load PS Sets structured field specifying a character size smaller (in either or both dimensions) will load the character starting at the top left of the designated character slot. If the width (LW) is less than the width of the character slot, then the remaining dots in each row of the character slot are cleared to zero. The height (LH) is processed in a similar manner when it is less than the height of the character slot. The effect of this processing is as if the character slot was cleared to zero immediately before the character that was transmitted (in the Load PS Sets structured field) was loaded into the character slot.

- **FORM** - In this 32-bit field, each bit corresponds to a format type that can be specified in the Load PS Set structured field. Thus, if bit i is set, the device supports format type i.

  Currently, Load PS Set format types 1 - 6 and 8 are defined. The remaining values are reserved. This means that bit 0 and bits 7 through 32 (with the exception of bit 8) are reserved. The remaining bits will be set depending on the formats supported.

- **DL** - Defines the length of each descriptor.

Character Set Descriptors

Function

Defines the characteristics of the character sets. Descriptors follow the base portion of a Character Sets Query Reply. At least one descriptor is required for each Character Sets Query Reply. Each descriptor's length is determined by the DL parameter; the number of bytes assigned to this parameter is the length of the descriptor.

Format

The table below shows the format of the character set descriptors.

Byte	Bit	Content	Content Description
1		SET	Device specific Character Set ID (PS store No.)
2		**Flags**	
	0	LOAD	
		B'0'	Non-loadable character set
		B'1'	Loadable character set
	1	TRIPLE	
		B'0'	Single-plane character set
		B'1'	Triple-plane character set
	2	CHAR	
		B'0'	Single-Byte coded character set
		B'1'	Double-Byte coded character set
	3	CB	
		B'0'	LCID compare
		B'1'	No LCID compare
	4-7	RES	Reserved
3		LCID	Local character set ID (alias)
4		SW	Width of the character slots in this character set. Present only if MS flag set to B'1'.
5		SH	Height of the character slots in this character set. Present only if MS flag set to B'1'.
6		SUBSN[1]	Starting subsection. Present only if CH2 flag = 1

Byte	Bit	Content	Content Description
7		SUBSN[1]	Ending subsection. Present only if CH2 flag = 1
8-11		CGCSGID[1]	Coded Graphic Character Set Identifier. Present only if GF flag = 1.

[1] The position of the CGCSGID and SUBSN parameters may vary. For example, if SUBSNs and SW/SH were not present, the CGCSGID parameter would start at byte 4.

Additional Content Description

- **SET** - Defines the device-specific character set ID. For a loadable character set, this value is specified in the RWS parameter of Load PS Set structured field. It also defines the key that can be used by the operator to select this character set. The character set with device specific ID equal to X'01' is defined as the alternate character page.

  If multiple base character sets are supported, then the device will return multiple descriptors with SET = X'00' (one descriptor for each supported base character set). The first such descriptor defines the device default character set.

  The BASE parameter in the Create Partition structured field is used to select the appropriate base character set for a partition by indexing down the base character set descriptors in the order returned in this query reply.

  The device default character set is the one associated with the implicit partition and with the keyboard.

- **LOAD** - Indicates whether or not this character set is loadable.

- **TRIPLE** - Indicates whether or not this device set has three planes: red, blue, and green.

- **CHAR** - Indicates whether the character set is a 1-byte or a 2-byte character set. It can take one of two values:

  - If CHAR = B'0' - the characters are represented by 1-byte codes in the data stream.

    The character slot maps into a display cell of the size equal to AW x AH (AW/AH are defined in the Usable Area query reply).

  - If CHAR = B'1' - the characters are represented by 2-byte codes in the data stream.

    The character slot maps into a display cell of size equal to 2AW x AH.

- **CB** - Indicates whether this character set can be compared for copy. See the section "Load Programmed Symbols (Load PS)" on page 5-18.

- **LCID** - Identifies the Local ID currently connected to this character set; it does not necessarily uniquely identify the contents. A value of X'FF' indicates that this character set is "free." (It cannot be accessed using an LCID in an SA, SFE, or MF order, and it cannot be selected by the operator using the PS [A-F] selection keys.) If the set is loadable, X'FF' also means that the set is available for local copy. The Load PS Set structured field allows LCID (alias) to be connected to a loadable character set. This is the data stream value used in the SA, SFE, or MF orders to refer to characters from this set. Nonloadable character sets will have default LCID assigned to them (X'F0' to X'FE'), where:

X'F0' to X'F7' = 1-byte character sets
X'F8' to X'FE' = 2-byte character sets.

- **SW and SH** - Define the size of all the character slots in the character set. The parameters are present only if character sets with different slot sizes are supported (that is, when the parameter MS = B'1'). If SW and SH are present but are set to zero in any descriptor, the character slot size for that character set is given by SDW and SDH, respectively.

  When a character is presented in the usable area, a dot matrix SW by SH will be placed in the top left of the appropriate cell.

  In a data type that is addressed in cells (for example, alphanumeric), the cell size, (X by Y) is defined when the partition is created, or it defaults to the values associated with the usable area (namely AW by AH). Thus, if SW < X, the remaining pels in each row will become background and similarly for SH < Y. If SW > X, each row of the dot matrix is truncated on the right and similarly for SH > Y.

  The rules for interpreting the Load PS Sets structured field data stream are summarized as follows:

  For FORMAT types 3 through 6:

  1. If LW and LH are present in the Load PS Sets structured field extension, then LW and LH will be used.
  2. Else, if SW and SH are present in the Character Set Descriptor, and are non-zero, SW and SH will be used.
  3. Else, SDW and SDH will be used.

  This operation is illustrated in the following figure.

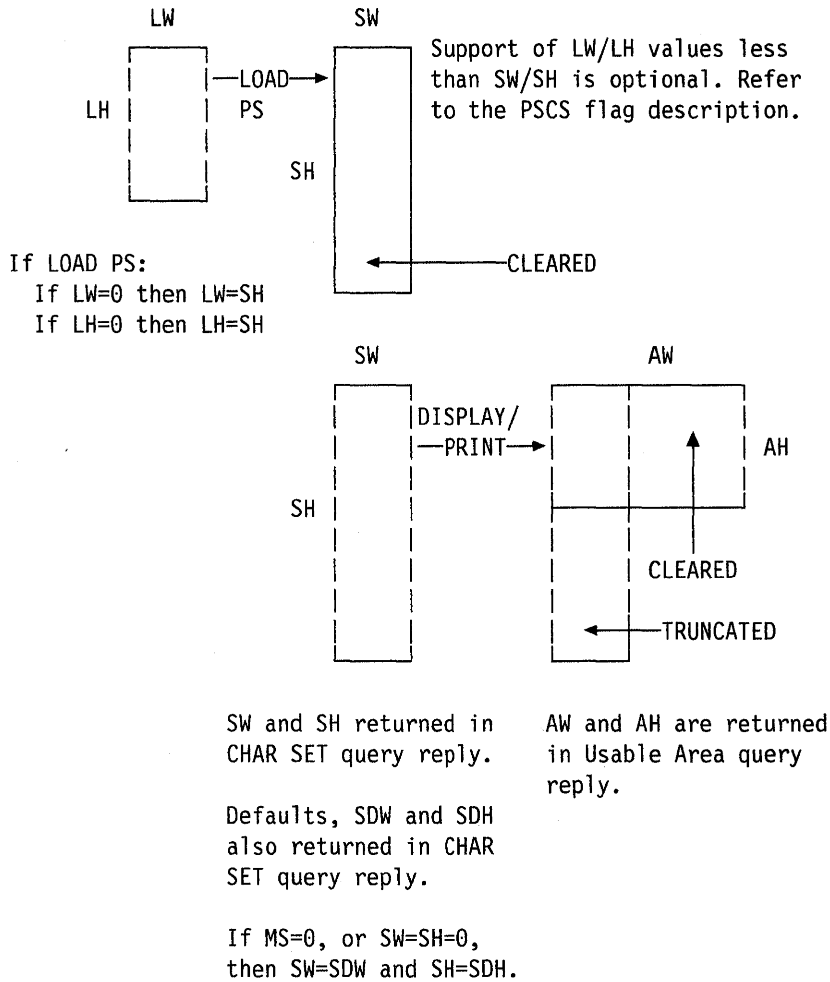


Figure 6-1. Diagram of steps used in interpreting Load PS Sets data stream.

Bytes 6 and 7 are not present when bit 2 of byte 2 is B'0' (1-byte coded character set). A 2-byte coded character set may consist of a nonloadable portion or both a nonloadable and a loadable portion. If a character set consists of both a nonloadable and loadable portion, then bit 0 of byte 0 (LOAD flag) is set to B'1' in the descriptor for that character set.

- SUBSN (bytes 6 and 7 of each descriptor) - Define the starting subsection and ending subsection identifiers supported by the device for the Programmed Symbol store being described.

  The starting subsection (byte 6 of each descriptor) defines the starting or lowest subsection identifier allowed for the 2-byte coded character set being described. The default value for a loadable 2-byte character set is X'41', but it may be set by the ST.SUBS of Load PS.

  For a nonloadable 2-byte character set, the starting subsection will be set to the lowest subsection identifier accepted by the device for that device character set. The ending subsection identifier will be set to the highest subsection identifier accepted by the device. Supported subsection identifiers must be consecutive within the range X'41' to X'FE'. For a loadable 2-byte character set, the starting subsection may be set by the ST.SUBSN parameter of LOAD PS.

The ending subsection will be set by the device based on two factors. The first is the starting subsection identifier; the second is the number of physical subsections supported (indicating the range of consecutive subsection identifiers accepted by the device for this loadable device set). The difference between the ending and starting subsection identifiers is one less than the number of subsections supported for that device set. Supported subsection identifiers must be within the range X'41' to X'FE'.

The starting and ending subsection identifiers for loadable 2-byte character sets are reset to their device default values by power-on and test mode. The device default values are X'41' for the starting subsection identifier and the appropriate value for the ending subsection identifier (indicating the number of consecutive subsections supported by the device set).

Subsection parameters (bytes 6 and 7) are present in all descriptors if and only if flag CH2 is set to B'1'. If they are present but the specified device set represents a 1-byte coded character set (CHAR = B'0'), then both SUBSN parameters are set to X'00'.

- CGCSGID - The Coded Graphic Character Set Global Identifier (CGCSGID) consists of a 2-byte character set identifier followed by a 2-byte code page identifier. The CGCSGID is included for each descriptor in the Character Set Query Reply. If the character set being defined in the descriptor has no associated CGCSGID (for instance, Programmable Symbol Sets, which are user defined), the 4 bytes will be set to zeros.

Note: The CGCSGID is made up of the 2-byte character set number and the 2-byte code page number. For more information about CGCSGID values, see the *IBM 3174 Subsystem Control Unit Character Set Reference*, GA27-3831.

Query Reply (Color)

Function

Transmits information about the color features of the device. The structured field is sent if the device is capable of interpreting 3270 field attribute as color attributes or accepts some subset of the color attribute values.

When this function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'86', Equivalent, or All).

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'86'	Color
4		Flags	
	0	RES	Reserved
	1		Printer Ribbon:
		B'0'	Printer only - black ribbon is *not* loaded
		B'1'	Printer only - black ribbon is loaded
	2-7	RES	Reserved
5		NP	Length of color attribute list (NP = number of CAV/COLOR pairs)
n		CAV(n)	Color attribute value accepted by the device
	n+1	CI(n)	Color identifier of the color displayed/printed for CAV(n)

Additional Content Description

- **CAV(n)** - The parameters CAV(n) are all those color attribute values that are accepted by the device without causing a negative response. Bytes n and n+1 are repeated for each of the data stream values accepted by the device.
- **CI(n)** - The parameters CI(n) identify the colors that are displayed or printed by the device for each of the accepted color attribute values. Note that the device must either display the color whose color identifier is the same as the color attribute value or display the device default color.

The color associated with the CAV(n) value of X'F7' defines the default color that is displayed or printed when a single plane character set is referenced; the associated CI(n) value may be any of the values in CAV(n), including X'00'.

The CAV(n) value of X'00' may have an associated CI(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', CI1 = value associated with the device default color, as the first entry in the CAV/CI pairs list.

Self-Identifying Background Color Subfield

Format

Byte	Content	Content Description
0	L	Length of subfield
1	BCOL X'02'	Type of subfield background color
2	CAVDEF X'00'	Default color attribute value
3	CIDEF	Default background color identifier

Additional Content Description

- **BCOL** - is a self-identifying subfield describing the support of background color. The subfield has an identifier of X'02' and will be present only if the device supports background color. There will only be one of this type of subfield. All color attribute values supported by (foreground) color are supported by background color with the same color identifiers. Therefore, only the background color default is identified by the CAVDEF and CIDEF parameters.

Query Reply (Color) Example

	CI(n) Color Identifier			
CAV(n) Attribute Value	**Color Display**	**Mono Display**	**Color Printer**	**3290 Display**
X'00'	X'F4'	X'F4'	X'F7'	X'F8'
X'F1'	X'F1'	X'00'	X'F1'	X'00'
X'F2'	X'F2'	X'00'	X'F2'	X'00'
X'F3'	X'F3'	X'00'	X'00'	X'00'
X'F4'	X'F4'	X'00'	X'F4'	X'00'
X'F5'	X'F5'	X'00'	X'00'	X'00'
X'F6'	X'F6'	X'00'	X'00'	X'00'
X'F7'	X'F7'	X'00'	X'00'	X'00'

Query Reply (Cooperative Processing Requestor)

Function

Indicates that Cooperative Processing Requestor (CPR) functions are supported.

Format

The table below shows the format of this query reply.

Byte	Content	Meaning
0-1	L	Length of structure
2	X'81'	Query Reply
3	X'AB'	Cooperative Processing Requestor
4-5	**Flags**	Reserved
6-7	LIMIN	Maximum CPR bytes/transmission allowed inbound
8-9	LIMOUT	Maximum CPR bytes/transmission allowed outbound
10	FEATL	Length (in bytes) of feature information that follows
11-12	FEATS	CPR length and feature flags
13 to (N*2) + 12	FEATSs	Additional flags

Self-Defining Parameters

Byte	Content	Meaning
0	X'04'	Parameter Length
1	X'01'	Direct Access ID
2-3	DOID	Destination/Origin Identification

When this function is supported, the query reply is transmitted inbound in reply to a Query or Query List (QCODE List=X'AB', Equivalent, All).

1. CPR Query Reply

   - **LIMIN** – Certain implementations may have a limit on the number of bytes of CPR information that are allowed inbound following an AID88. The number of bytes in an inbound CPR transmission must be equal to or less than the LIMIN value. A LIMIN value of X'0000' indicates no implementation limit on CPR data inbound.

   - **LIMOUT** – The sum of bytes contained in all the CPR structured fields following a WSF command must be equal to or less that the LIMOUT value. If this limit is exceeded, the transmission will be rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on CPR data outbound.

     The LIMOUT parameter applies only to the CPR structured fields. For example, if LIMOUT = 400 bytes, a WSF followed by a CPR SF (100 bytes), 3270 SF (500 bytes) and a CPR SF (300 bytes) would be accepted.

- **FEATL** — Indicates the number of bytes of feature information that follow. The minimum value is X'02'.

- **FEATS** — The content of the feature bytes is defined in related publications.

2. Direct Access ID Self-Defining Parameter

   - **DOID** — The presence of the Direct Access ID self-defining parameter indicates that the CPR device may be accessed directly. The value given in the DOID field is used in the Destination/Origin structured field to indicate that the destination/origin of the data following it is the CPR device. When more than one CPR device is supported, each one will have a separate CPR Query Reply and a separate ID.

     For the Query Reply to be valid, this self-defining parameter must be present.

Query Reply (Data Chaining)

Function

Indicates that data chaining is supported in the non-SNA environment.

When the function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE=X'98', Equivalent, or All).

Format

The table below shows the format of this query reply.

Byte	Bit	Content	Content Description
0−1		X'0006'	Length of structure
2		X'81'	Query reply
3		X'98'	Data chaining
4	0-1	DIR	Indicates which direction can use the Data Chain structured field.
		B'00'	Both
		B'01'	From device only
		B'10'	To device only
	2-7	B'11'	Reserved
5		**Flags**	Reserved

Query Reply (Data Streams)

Function

Indicates which data streams are supported by the device. The positional relationship of the data within the structured field indicates the default data stream.

The currently defined values for the list of data stream identifiers are as follows:

- X'00' - SCS Base Data Stream with extensions as specified in the BIND Request and Device Characteristics Query Reply structured field
- X'01' - Document Content Architecture Level 2
- X'02' - IPDS as defined in related documentation.

All other values are reserved. As noted above, the first identifier to appear in the subfield (for example, at byte 4 of the structured field) will be assumed to be the default data stream.

When the function is supported, this query reply is transmitted inbound in response to a Read Partition structured field specifying Query, or Query List (QCODE List = X'A2', Equivalent, or All).

Format

The table below shows the format of this query reply.

Byte	Content	Content Description
0-1	L	Length of this structure
2	X'81'	Query Reply
3	X'A2'	Data Streams
4	X'nn...'	List of (L-4) data stream identifiers, the first of which is the default data stream

Query Reply (DBCS-Asia)

Function

This Query Reply indicates the support of the DBCS-Asia node as defined in Chapter 12, "Double-Byte Coded Character Set (DBCS) Asia."

When this function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'91', Equivalent, or All).

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'91'	DBCS-Asia
4		**Flags**	Reserved

Self-defining parameters follow.

SO/SI Self-Defining Parameter

The presence of the SO/SI self-defining parameter indicates that the host may send SO/SI to the device. Also, if previously written by the host, SO/SI may be read back. However, unless indicated in the Input Control self-defining parameter, SO/SI may not be generated by the operator.

Format

Byte	Bit	Content	Content Description
0		X'03'	Parameter length
1		X'01'	SO/SI
2		SOSET	Set ID of the Shift Out (SO) character set

- **SOSET** - designates the Set ID of the character set associated with SO. The LCID that the implementation assigned to the character set designated in SOSET is reported in the Character Sets query reply.

Input Control Self-Defining Parameter

The presence of the Input Control self-defining parameter indicates support of the extended field attribute type, Input Control.

Format

Byte	Bit	Content	Content Description
0		X'03'	Parameter length
1		X'02'	Input Control
2		FUNC	Functions supported
	0-6	RES	Reserved
	7	CREATE	Operator creation of SO/SI
		B'0'	SO/SI Creation not supported
		B'1'	SO/SI Creation supported

- **CREATE** - indicates whether or not the Input Control may be used to enable/disable the operator creation of SO/SI.
  - B'0' - indicates creation of SO/SI by the operator is not supported.
  - B'1' - indicates creation of SO/SI by the operator is supported and may be enabled/disabled using the Input Control extended attribute. The default (for example,, POR) is disabled.

Note: When CREATE = B'1' (operator creation of SO/SI supported), the SO/SI self-defining parameter must be present.

Query Reply (DDM)

Function

This Query Reply indicates the Distributed Data Management (DDM) subsets supported.

When this function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'95', Equivalent, or All).

Format

Byte	Content	Content Description
0-1	L	Length of structure
2	X'81'	Query Reply
3	X'95'	DDM
4-5	**Flags**	Reserved
6-7	LIMIN	Maximum DDM bytes/transmission allowed inbound
8-9	LIMOUT	Maximum DDM bytes/transmission allowed outbound
10	NSS	Number of subsets supported
N	DDMSS	DDM subset identifier

Additional Content Description

- **LIMIN** - Certain implementations may have a limit on the number of bytes of DDM information that are allowed inbound following an AID88. The number of bytes in an inbound DDM transmission must be equal to or less than the LIMIN value. A LIMIN value of X'0000' indicates no implementation limit on DDM data inbound.

- **LIMOUT** - The sum of bytes contained in all the DDM structured fields following a WSF command must be equal to or less than the LIMOUT value. If this limit is exceeded, the transmission will be rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on DDM data outbound.

  The LIMOUT parameter applies only to the DDM structured fields. For example, if LIMOUT = 400 bytes, a WSF followed by a DDM structured field (100 bytes), 3270 structured field (500 bytes) and a DDM structured field (300 bytes) would be accepted.

- **NSS** - Indicates the number of different subsets that are supported; that is, the number of DDMSS parameters present.
- **DDMSS** - Indicates the DDM subset that is supported. Valid values are:
  - X'01' = DDM Copy Subset 1
  - All other values reserved.

Direct Access ID, Self-Defining Parameter

Format

Byte	Content	Content Description
0	X'04'	Parameter Length
1	X'01'	Direct Access ID
2-3	DOID	Destination/Origin identification

Additional Content Description

- **DOID** - The presence of the Direct Access ID self-defining parameter indicates the DDM "device" may be accessed directly. The value given in the DOID field is used in the Destination/Origin structured field to indicate the destination/origin of the following data is the DDM "device."

Query Reply (Device Characteristics)

Function

Transmits the device's ability to support SNA Character String (SCS) functions.

When the function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query, or Query List (QCODE List = X'A0', Equivalent, or All). The function descriptor is included in the Device Characteristics query reply to indicate how an SCS function is supported.

As a general rule, any of the SCS control codes reported in the Device Characteristics query reply, which are received with invalid parameters, will be rejected. However, there are some situations where there is a relationship between parameters in which a normally valid parameter value becomes invalid in combination with another parameter value. For example, a device supporting character densities of 10 and 15 cpi could support a maximum MPP of 160 only when the character density was 15 cpi. The descriptors describe the action taken for the sets of parameters that are interrelated.

Format

Byte	Content	Content Description
0-1	L	Length of this structure
2	X'81'	Query Reply
3	X'A0'	Device Characteristics

Function Descriptors

The base part is followed by one or more function descriptors, each defining the characteristics of one SCS control. The SCS controls having descriptors are defined below.

Set Print Density (SPD) Descriptor

Function

The Set Print Density descriptor is included for historical purposes. The Horizontal Dimensional Parameters descriptor is to be used in its place.

Format

Byte	Content	Content Description
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'	Character density parameter length
10	X'60'	The character density is present sometimes. It only takes discrete values, and it appears only one time.
11	X'mm'	Number of discrete values supported by the device
12-n		The discrete character density values supported.

Horizontal Dimensional Parameters Descriptor

Function

When using Horizontal Dimensional Parameters, character density (CD) and maximum print position (MPP) occur as 2-byte pairs, one for each density. The character density is the number of characters per inch, rounded down to the next integer. The first CD/MPP pair is the device default setting. The other CD/MPP pairs must appear in the ascending order of MPP values.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2-3		X'FF01'	Horizontal Dimensional parameters
4		**Flags**	
	0	B'0'	SPD not supported
		B'1'	SPD supported
	1-7	0	Reserved
4+2n-1		CD(n)	Character Density
4+2n		MPP(n)	Maximum Print Position for CD.

Additional Content Description

- **CD/MPP** - CD is given preference, that is, it will take effect, regardless of whether or not the maximum MPP in effect is valid for that CD. If the maximum MPP in effect becomes invalid when a new CD is set, the maximum MPP becomes the highest value valid for the new CD. If a new MPP is set using the SHF control that is invalid for the CD in effect, the MPP will be rejected.

Vertical Dimensional Parameters Descriptor

Function

When using Vertical Dimensional Parameters, typographic points and maximum page length (MPL) occur as 2-byte pairs, one for each density. The typographic points are expressed in 1/72nds of an inch and are rounded down to the next integer. The first typographic point/MPL pair is the device default setting. The other typographic point/MPL pairs must appear in the ascending order of MPL values.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2-3		X'FF02'	Vertical Dimensional parameters
4		**Flags**	
	0	B'0'	Set Line Density not supported.
		B'1'	Set Line Density (SLD) supported.
	1-7	RES	Reserved
4+2n-1		PNTS(n)	Typographic points
4+2n		MPL(n)	Maximum page length for PNTS

Additional Content Description

- **PNTS/MPL** - Treated the same as CD/MPP, the line density PNTS is given the preference, and a valid PNTS will be accepted whether or not the MPL in effect is valid for that PNTS. Also, if a new MPL is set via the SVF control that is invalid for the PNTS in effect, the MPL will be rejected.

Page Presentation Media (PPM) Descriptor

Function

The PPM Descriptor indicates whether the device provides support for such functions as forms control, paper source and destination drawers, print quality level, and duplex printing.

On certain devices, Programmable Symbol sets will have only the Quality levels inherent in the characteristics in which they were loaded into the device. On these devices, quality may not be changed on PS sets. Other devices have the capability of changing quality for both PS sets as well as nonloadable character sets by not requiring a downstream load to change the quality level. Devices will set the Quality flag byte, bit 1, to designate their capability to change quality on PS sets.

Note: On certain devices, changes in print quality may inherently result in a change in cell geometry. Following a change in the Quality parameter, the Primary should issue a Query List with the Usable Area Query Reply specified to determine the effects of the quality change upon the cell geometry.

For some devices, certain combinations of supported quality and character density are invalid. The existence of this condition will be indicated by setting bit 3 of byte m+1 to B'1' and including a count of the number of invalid Q,CD pairs. This is followed by a list of the invalid Q,CD combinations (two bytes per pair). Devices that can support all combinations of supported quality and character density will set bit 3 of byte (m+1) to B'0' and will not include the above fields.

Format

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2-3		X'FF03'	Page Presentation Media descriptor
4		Flags	
		B'0'	PPM not supported
	0	B'1'	PPM supported
			Reserved
	1-7		
5		FC	Parameter Forms Control (FC)
	0	B'0'	FC not supported
		B'1'	FC supported
	1-7		Reserved
6		FCNUMB	
		X'nn'	Number of discrete Forms Control (FC) parameter values supported by the device.
7 to j			The discrete FC values supported. The first value is the default value. (1 byte per value)
j+1		SD	
	0		Parameter Source Drawer (SD)
		B'0'	SD not supported
		B'1'	SD supported
	1	B'0'	This parameter is not operator-selectable at the device.
		B'1'	This parameter is operator-selectable at the device. Its current value may be invoked by the Primary setting SD = X'FF'.
	2-7	RES	Reserved
j+2		SDNUMB	
		X'nn'	Number of discrete Source Drawer (SD) parameter values supported by the device.
j+3 to k		SDPARM	The discrete SD values supported. The first value is also the default value (1 byte per value).
k+1		DDO	Parameter Destination Drawer Offset (DDO)
	0	B'0'	DDO not supported
		B'1'	DDO supported
	1-7		Reserved

Byte	Bit	Content	Content Description
k+2		DD	Parameter Destination Drawer (DD)
	0	B'0'	DD not supported
		B'1'	DD supported
	1	B'0'	This parameter is not operator-selectable at the device.
		B'1'	This parameter is operator-selectable at the device. Its current value may be invoked by the Primary setting DD=X'FF'.
	2-7		Reserved
k+3		DDNUMB X'nn'	Number of discrete Destination Drawer (DD) parameter values supported by the device.
k+4 to m			The discrete values supported. The first value is also the default value.
m+1		QUAL	Parameter Quality (Q)
	0	B'0'	Q not supported
		B'1'	Q supported
	1	B'0'	This parameter is not operator-selectable at the device.
		B'1'	This parameter is operator-selectable at the device. Its current value may be invoked by the Primary specifying Q=X'FF'.
	2	B'0'	Q is not applicable to PS sets.
		B'1'	Q is applicable to PS sets.
	3	B'0'	All supported quality levels are allowed at all supported character densities.
		B'1'	Some combinations of supported quality levels/character densities are not valid.
	4-7		Reserved
m+2		X'nn'	Number of discrete Quality (Q) parameter values supported by the device.
m+3 to n		QPARMS	The discrete Q values supported. The first value is also the default value (1 byte per value).
n+1		X'PP'	Number of discrete Quality (Q)/Character Density (CD) pair parameter values that are not allowed by the device.
n+2 to p		Q/CD(P)	Q/CD pairs of valid values that are not allowed as a combination. Two bytes per pair.

Byte	Bit	Content	Content Description
p+1		DUP	Parameter Duplex (D)
	0	B'0' B'1'	Duplex is not supported. Duplex is supported.
	1	1	This parameter is not operator-selectable at the device.
		B'1'	This parameter is operator-selectable at the device. Its current value may be invoked by the Primary setting D = X'FF'.
	2-7		Reserved

Additional Content Description

- **CD/Q** - CD is given preference. A valid CD will take effect whether or not the quality (Q) level in effect is valid for that CD. When a new CD selection makes the current Q level invalid, the Q level will default to the next highest valid Q level. If a new valid Q level is set via the PPM that is invalid because of the CD in effect, it will be rejected.
- **FCPARM** - Valid values for the Forms Control mechanisms are:
  - X'01' - Paper Source Drawer
  - X'02' - Envelope Source Drawer
  - X'03' - Manual Paper Drawer
  - X'04' - Manual Envelope Drawer
- **SDPARM** - Valid values for the Paper Source Drawers are:
  - X'01' - Select paper from bin 1
  - X'02' - Select paper from bin 2
  - X'03' - Select paper from bin 3
  - X'04' to 'FE' - Select paper from respective drawer
- **DDPARM** - Valid values for the Paper Destination Drawers are:
  - X'01' - Primary destination drawer
  - X'02' to 'FE' - Secondary destination drawers
- QPARM - The number of print quality levels supported are listed numerically in monotonically increasing order (in other words, the first number is the lowest and the last number is the best print quality.

Note that for interrelated parameters (for example, CD/MPP), when both parameters are changed in the same transmission, the results obtained can be different, depending on which parameter appears first.

Set Text Orientation (STO) Descriptor

Function

Indicates that the Set Text Orientation SCS control code is supported by the device.

Format

Byte	Bit	Content	Content Description
0 - 1	X'0005'	L	Length of this structure
2 - 3		X'FF04'	Set Text Orientation Descriptor
4		Flags	
	0	B'0'	STO not supported
		B'1'	STO supported
	1 - 7		Reserved

Query Reply (Document Interchange Architecture (DIA))

Function

Indicates the Document Interchange Architecture (DIA) function sets supported.

When the function is supported, this query reply is transmitted inbound in reply to a Read Partition structured field specifying Query, or Query List (QCODE List =X'97', Equivalent, or All).

Format

Byte	Content	Content Description
0-1	L	Length of structure
2	X'81'	Query Reply
3	X'97'	DIA
4-5	**Flags**	Reserved
6-7	LIMIN	Maximum DIA bytes/transmission allowed inbound
8-9	LIMOUT	Maximum DIA bytes/transmission allowed inbound
10	NFS	Number of three byte function set identifiers that follow
11-13	DIAFS	DIA function set identifier
N-N=2	DIAFSs	Additional DIA function set identifiers

Additional Content Description

- **LIMIN** - Certain implementations may have a limit on the number of bytes of DIA information that are allowed inbound following an AID X'88'. The number of bytes in an inbound DIA transmission must be equal to or less than the LIMIN value. A LIMIN value of X'0000' indicates no implementation limit on DIA data inbound.
- **LIMOUT** - The sum of bytes contained in all the DIA structured fields following a WSF command must be equal to or less than the LIMOUT value. If this limit is exceeded, the transmission will be rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on DIA data outbound.

  The LIMOUT parameter applies only to the DIA structured fields. For example, if LIMOUT=400 bytes, a WSF followed by a DIA structured field (100 bytes), an Outbound 3270DS structured field (500 bytes), and a DIA structured field (300 bytes) would be accepted.
- **NFS** - The number of different function sets that are supported, that is, the number of 3-byte DIAFS parameters present. A valid query reply must have at least one DIAFS.
  - X'01' = File Server
  - X'02' = File Requestor
  - X'03' = Both File Server and File Requestor. All other values reserved.

  The second and third bytes give the function set number in hexadecimal. For example, to indicate the role of the File Server with support of function set 11, the value of bytes 11 through 13 would be X'01000B'.

For a description of the Document Interchange Architecture functions refer to:

1. *Document Interchange Architecture: Technical Reference*, SC23-0781.
2. *Document Interchange Architecture: Document Profile Reference*, SC23-0764.
3. *Document Interchange Architecture: Transaction Programmer's Guide*, SC23-0763.

Self-Defining Parameters

Direct Access ID Self-Defining Parameter

For the query reply to be valid, this Self-Defining Parameter must be present.

Byte	Content	Content Description
0	X'04'	Parameter length
1	X'01'	Direct Access ID
2-3	DOID	Destination/Origin identification.

Additional Content Description

- **DOID** - The presence of the Direct Access ID self-defining parameter indicates that the DIA "device" may be accessed directly. The value given in the DOID field is used in the Destination/Origin structured field to indicate that the destination/origin of the following data is the DIA "device." When more than one DIA "device" is supported, each one will have a separate DIA Query Reply and a separate ID.

Query Reply (Field Outlining)

Function

Specifies the details of the field outlining supported by the device.

When this function is supported, the query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'8C', Equivalent, or All).

Format

Byte	Bit	Content	Content Description
0-1		X'000A'	Length of this structure
2		X'81'	Query Reply
3		X'8C'	Field Outlining
4	0-7	Flag	Reserved
5	0	SEP	Support of separation of underlining and overlining:
		B'0'	Separation *not* supported
		B'1'	Separation supported
	1-7	RES	Reserved
6		VPOS	Location of vertical line
7		HPOS	Location of overline/underline
8		HPOS0	Location of overline in case of separation
9		HPOS1	Location of underline in case of separation

Additional Content Description

- **SEP** – indicates that the separation of under/overline is supported in printers. When the function is not supported (SEP = B'0'), an underline and the next overline are always printed as one line. When the function is supported (SEP = B'1'), an underline and the next overline are separated when specified.
- **VPOS** – indicates the horizontal location of the vertical line.
- **HPOS** – indicates the vertical location of the horizontal line. In the printer, the location of the horizontal line when the separation is not specified is indicated here. When the overline is drawn above the cell, the location of the underline, which is the same as the location of the overline in the next row, is used.

- **HPOS0** and **HPOS1** — indicate the vertical locations of the overline and the underline respectively for the case Separation is specified. In displays or printers where the device does not support the Separation, zero values are set. When the overline is drawn above the character cell, the location of the overline in the next row is used.

  Each of VPOS, HPOS, HPOS0, and HPOS1 is a 1-byte binary number, and they indicate the location of the Field Ruling lines against the top left corner of the character cell, where the value of the location is zero vertically and horizontally. These values are measured in the same unit as that in cell units of the Usable Area query reply, and measured when the Skip Suppression is specified.

Query Reply (Extended Drawing Routine)

Function

Indicates at which graphic subset level extended drawing routines are supported.

When this function is supported, it is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List= X'B5', Equivalent or All).

Format

Byte	Content	Content Description
0-1	X'09'	Length of this structured field
2	X'81'	Query Reply
3	X'B5'	Extended Drawing Routine
4-n	DATA	Data

Note: For information regarding the format and operation of the DATA parameter, refer to related graphics documentation.

Query Reply (Field Validation)

Function

Specifies that the device supports field validation and indicates the types of validation the device supports.

When this function is supported, the query reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'8A', Equivalent, or All).

Format

Byte	Bit	Content	Content Description
0-1		X'0005'	Length of this structure
2		X'81'	Query Reply
3		X'8A'	Field Validation
4		TYPES	Types supported:
	0-4	RES	Reserved
	5	B'0' B'1'	Mandatory fill *not* supported Mandatory fill supported
	6	B'0' B'1'	Mandatory entry *not* supported Mandatory entry supported
	7	B'0' B'1'	Trigger *not* supported Trigger supported

Query Reply (Format Presentation)

Function

Specifies that the device supports Format Presentation.

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'90', Equivalent, or All).

Format

Byte	Content	Content Description
0-1	X'0004'	Length of structure
2	X'81'	Query Reply
3	X'90'	Format Presentation

Query Reply (Graphic Color)

Function

Indicates the support for color available in graphics.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List =X'B4', or All).

Format

Byte	Content	Content Description
0-1	L	Length of structured field
2	X'81'	Query Reply
3	X'B4'	Graphic Color
4-n	DATA	Data

Note: For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

Query Reply (Graphic Symbol Sets)

Function

Reports all of the symbol set stores that are available for use in graphics.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List =X'B6', or All).

Format

Byte	Content	Content Description
0-1	L	Length of structured field
2	X'81'	Query Reply
3	X'B6'	Graphic Symbol Sets
4-n	DATA	Data

Note: For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

Query Reply (Highlight)

Function

Transmits the types of highlighting supported by the device.

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'87', Equivalent, or All).

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. The attribute values listed in the query reply are accepted by the device. The following attribute values are the only valid values:

- X'00' Default
- X'F0' Normal Highlight
- X'F1' Blink
- X'F2' Reverse Video
- X'F4' Underscore
- X'F8' Intensify.

This indicates that the device supports highlighting on an exclusive basis; that is, one and only one of the highlight values may be applied to a field or character location.

Format

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.

Byte	Content	Content Description
0-1	L	Length of this structure
2	X'81'	Query Reply
3	X'87'	Highlight
4	NP	Number of attribute-value/action pairs
n	Vi	Data stream attribute value accepted
n+1	Ai	Data stream action

The following example illustrates this Query Reply response.

Byte	Content	Content Description
0-1	X'000C'	Length
2	X'81'	Query Reply
3	X'87'	Highlighting
4	X'04'	Number of Pairs
5 6	X'00' X'F0'	Attribute Value (Default) (Pair 1) Action - Normal
7 8	X'F1' X'F1'	Attribute Value (Pair 2) Action - Blink
9 10	X'F2' X'F2'	Attribute Value (Pair 3) Action - Reverse Video
11 12	X'F4' X'F4'	Attribute Value (Pair 4) Action - Underscore

Query Reply (Image)

Function

This query reply provides specific information about the device support of image parameter sets and image order sets.

This query reply is returned to the host in reply to a Read Partition structured field specifying Query List (QCODE List=X'82', Equivalent or All).

Format

This query reply has the following format:

Byte	Content	Content Description
0-1	LL	Length of this structured field
2	X'81'	Query Reply
3	X'82'	IMAGE
4-n	COMMAND	Command

Note: For information requiring the format and operation of the COMMAND parameter, refer to related imaging documentation.

Query Reply (Implicit Partition)

Function

Defines unique implicit partition characteristics.

This query reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

Requirements

The Implicit Partition Query Reply must always be sent inbound in reply to a Read Partition structured field by any device supporting EBASE.

This query reply consists of three parts: the base, one of two self-defining parameters for devices, and a character cell self-defining parameter. The self-defining parameters are:

1. The Implicit Partition sizes for display devices self-defining parameter **or** the Implicit Partition sizes for printer devices self-defining parameter
2. If needed, the Implicit Partition sizes for character cell dimensions self-defining parameter.

The minimum support for the Implicit Partition Query Reply consists of items 1 and 2. Three is supported only as needed. All parts making up an Implicit Partition Query Reply are discussed later in this section.

Note: When the third parameter (on the list above) is **not** present, the character cell size for the Implicit Partition is specified in the Usable Area Query Reply.

Implicit Partition Query Reply Base

Function

Precedes any Implicit Partition Query Reply self-defining parameters. It is always required.

Format

The table below shows the format of the Implicit Partition Query Reply Base.

Byte	Content	Content Description
0-1	L	The length of this structure.
2	X'81'	Query Reply
3	QCODE X'A6'	 Implicit Partition Query Reply
4-5	**Flags** X'0000'	 Reserved

Implicit Partition Query Reply Self-Defining Parameters

This section discusses the self-defining parameters that make up at least one part of the Implicit Partition Query Reply.

Implicit Partition Screen Sizes for Display Devices

Function

Informs the host of the default and alternate screen sizes of the Implicit Partition. The sizes are specified in character cells. This parameter is required for all *display* devices.

When the default and alternate screen sizes are specified, two things must be true:

- Default and alternate values must be nonzero.
- If the device does not have an alternate screen size, the value for the alternate screen size must be that of the default screen size.

The Implicit Partition default and alternate screen sizes are established differently in SNA and non-SNA systems. In SNA systems, the default and alternate sizes returned in this reply are those established at BIND. In non-SNA systems, the default and alternate sizes returned in this reply are those in effect at the time the query reply is generated.

Format

The table below shows the format of these self-defining parameters.

Byte	Content	Content Description
0	X'0B'	Length of this self-defining parameter.
1	X'01'	Implicit Partition Sizes.
2	**Flags** X'00'	 Reserved
3-4	WD	Width of the Implicit Partition default screen size (in character cells).
5-6	HD	Height of the Implicit Partition default screen size.
7-8	WA	Width of the Implicit Partition alternate screen size.
9-10	HA	Height of the Implicit Partition alternate screen size.

Implicit Partition Sizes for Printer Devices

Function

Informs the host of the default and alternate printer buffer sizes. The sizes are specified in character cells. This parameter is required for all *printer* devices.

When the default and alternate printer buffer sizes are specified, two things must be true:

- Default and alternate values must be nonzero.
- If the device does not have an alternate print buffer size, the value for the alternate size must be that of the default size.

The Implicit Partition default and alternate printer buffer sizes are established differently between SNA and non-SNA systems. In SNA systems, the default and alternate sizes returned in this reply are those established at BIND. In non-SNA systems, the default and alternate sizes returned in this query reply are those in effect at the time that the query reply is generated.

The buffer size defines the following printer buffer restrictions:

1. The maximum linear character buffer address that can be explicitly specified in 3270 orders. (The maximum buffer address is one less than the buffer size in character cells.)
2. The wrapping point for the transmitted data.

   If the implied address for the data being loaded into the character buffer exceeds the maximum address allowed by the buffer size, then the implied address is reset to zero and loading continues from the first buffer location.

Format

The table below shows the format of these self-defining parameters.

Byte	Content	Content Description
0	X'0B'	Length of this self-defining parameter.
1	X'03'	Implicit Partition Sizes
2	**Flags** X'00'	 Reserved
3-6	DPBS	Default printer buffer size (in character cells)
7-10	APBS	Alternate printer buffer size

Implicit Partition Sizes for Character Cell Dimensions

Function

Informs the host system of the character cell sizes associated with the default and alternate Implicit Partition sizes. This parameter is **not** required for all Implicit Partition query replies.

Use this parameter only if:

- The cell size associated with either the default and/or alternate screen size for the Implicit Partition is **different** from the cell size that is reported in the Usable Area Query Reply parameters (AW/AH).
- The device supports the Load PS structured field.

Character Cells are measured in this Query Reply the same as in the Usable Area Query Reply parameters (UNITS, Xr, and Yr).

When this parameter is present, the default or alternate character cell sizes are determined by three factors:

- The Implicit Partition default screen size will use the default character cell size.
- The Implicit Partition alternate screen size will use the alternate character cell size.
- The default character cell size for the Explicit Partition is stated in the Usable Area Query Reply parameters (AW/AH). (There is no alternate size in Explicit Partition state.)

Format

The following table shows the format of this self-defining parameter.

Byte	Content	Content Description
0	X'0B'	Length of this self-defining parameter
1	X'02'	Implicit Partition Sizes
2	**Flags** X'00'	 Reserved
3-4	WCD	Width of character cell for the Implicit Partition default screen size
5-6	HCD	Height of character cell for the Implicit Partition default screen size
7-8	WCA	Width of character cell for the Implicit Partition alternate screen size
9-10	HCA	Height of character cell for the Implicit Partition alternate screen size

Query Reply (IOCA Auxiliary Device)

Function

This query reply indicates support of a 3270 workstation auxiliary device that uses the Image Object Content Architecture (IOCA) data stream.

This query reply is returned to the host as a result of receiving a Read Partition structured field specifying Query List (QCODE List= X'AA', Equivalent or All).

When a workstation supports multiple IOCA auxiliary devices, an IOCA Aux device Query Reply must be sent for each of them.

Format

This query reply has the following format:

Byte	Content	Content Description
0-1	LL	Length of this structured field
2	X'81'	Query Reply
3	X'AA'	IOCA Auxiliary Device
4-5	**Flags**	Reserved
6-7	LIMIN	Max IOCA bytes/inbound transmission
8-9	LIMOUT	Max IOCA bytes/outbound transmission
10	IOTYPE X'00' X'01' X'02'	Type of device Input device Output device

Note: For information requiring the format and operation of the COMMAND parameter, refer to related imaging documentation.

Additional Content Description

- **LIMIN** - The sum of bytes contained in all of the structured fields that are associated with the IOCA Aux device—following an AID 88— will be equal to or less than the value specified (in hexadecimal) in LIMIN. A LIMIN value of X'0000' indicates no implementation limit on IOCA data inbound.

- **LIMOUT** - The sum of bytes contained in all the IOCA structured fields associated with the IOCA auxiliary device— following a WSF command— must not exceed the LIMOUT value. If this limit is exceeded, the transmission must be rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on IOCA data outbound.
- **IOTYPE** - This specifies the type of device. There are three types of devices: Input, Output, and Input/Output.

Self-Defining Parameters

For information concerning self-defining parameters, refer to related image documentation.

Query Reply (Line Type)

Function

Indicates which line type attribute values are supported and the corresponding device action. Also, for loadable line types, it indicates which formats of line type definition are supported, as well as an indication of which, if any, code points are already loaded.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List =X'B2' or All).

Format

Byte	Content	Content Description
0-1	L	Length of structured field
2	X'81'	Query Reply
3	X'B2'	Line Type
4-n	DATA	Data

Note: For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

Query Reply (MSR Control)

Function

Defines what magnetic slot reader devices are attached and specifies the magnetic slot reader type that the application program can use to determine the control requirements of the device.

When the function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'8B', Equivalent, or All).

Format

Byte	Content	Content Description
0-1	X'0007'	Length of structured field
2	X'81'	Query Reply
3	X'8B'	MSR Control
4	**Flags**	Reserved
5	ND	Number of MSR device types
6		MSR type:
	X'01'	Magnetic reader
	Other	Reserved

Query Reply (Null)

Function

Informs the host that the device does not support any of the features or functions that the Read Partition structured field specifying Query List (QCODE List) concerns.

This Query Reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support." If the host queries the device and the device supports at least one feature or function that was queried about, then the Null Query Reply is **not** sent to the host. Case 1 is an example:

Case 1

- A device supports features A, B, and C.
- The host queries for features A, X, and Z.

In this case the device will not send the Null Query Reply. It will send the Query Reply, but only for feature A. From such a reply, the host knows that the device does not support features X and Z.

Case 2 provides an example of a Null Query Reply.

Case 2

- A device supports features A, B, and C.
- The host queries for features X, Y, and Z.

In this case the device will send the Null Query Reply because the device does not support *any* of the requested features.

Note: This Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query List (QCODE List=X'FF').

Format

The table below shows the format of this Query Reply.

Byte	Content	Content Description
0-1	X'0004'	Length of Structure
2	X'81'	Query Reply
3	X'FF'	Null

Query Reply (OEM Auxiliary Device)

Function

This Query Reply indicates support of an OEM auxiliary device (see Chapter 11, "Auxiliary Devices and Workstations"). An OEM device is defined here as a device that is manufactured outside of IBM and does not use an IBM defined data stream. The device may carry either an outside manufacturer logo or an IBM logo.

When the function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'8F', Equivalent, or All).

When a workstation supports multiple OEM devices, an OEM Aux Device Query Reply must be sent for each of the devices.

Optional Parameters:

All parameters shown in the "base" part of the Query Reply must be present.

At least one self-defining parameter must be present. Since only the Direct Access self-defining parameter is currently defined, it must be present.

Format

Byte	Content	Content Description
0-1	L	Length of Structure
2	X'81'	Query Reply
3	X'8F'	OEM Aux Device
4	Flags	Reserved
5	DSREF	Data stream reference identifier
6-13	DTYPE	Device type
14-21	UNAME	User assigned name

Additional Content Description

- **DTYPE** - This is an 8-byte character string used to send the device type to the "host" application. An understanding must exist between the workstation and the host on the relationship of the DTYPE value and the device characteristics (for example, what data the device will accept or send).

- **DSREF** - This parameter is used to indicate what data may be contained in an OEM Data structured field directed to/from the OEM auxiliary device.

  A value of X'00' indicates the data in the OEM Data structured field must be the data stream the device recognizes or sends as derived from the DTYPE parameter.

  A nonzero value indicates that the OEM Data structured field contains a "value added" data stream. That is, the OEM Data structured field contains controls and other data, in addition to the data recognized/sent by the device. Further workstation processing of this "value added" data stream is required before sending the data to the OEM auxiliary device. See your related product documentation for a description of the value added data stream associated with the DSREF parameter.

Valid values for the DSREF parameter (in addition to X'00') are:

- X'01' = Plotter Type 1
- All other values reserved.

- **UNAME** - This parameter contains an 8-byte character string name provided by the workstation. The purpose is to provide a "user friendly" name (for example, plotter1, plotter2) for use by the application. This name will not appear in the data stream between the host and the workstation. A value of all zeros indicates no name assigned. The value X'FF FF' is reserved.

Direct Access Self-Defining Parameter

This self-defining parameter provides the ID for use in the Destination/Origin structured field in the direct access of the OEM auxiliary device.

Format

Byte	Content	Content Description
0	X'04'	Parameter Length
1	X'01'	Direct Access
2-3	DOID	Destination/Origin ID

Additional Content Description

- **DOID** - The value in this byte is used in the ID field of the Destination/Origin structured field to identify the auxiliary device as the destination or origin of the data that follows.

Query Reply (Paper Feed Techniques)

Function

This Query reply transmits the currently installed and active paper feed technique (for example, continuous forms, cut sheets,). It transmits the size of the restricted print areas (if any) at the top and/or bottom of the presentation surface (form).

When the function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'A7', Equivalent, or All).

A device is required to support this Query Reply if any area of the presentation surface (form) is restricted. Otherwise a device has the option of whether this Query Reply is supported.

The top and bottom margin offsets define the areas at the top and bottom of the presentation surface (form) where printing should not occur. The top and bottom margin offsets can range from 0 (meaning there is no restricted area) to 65 535/1440ths of an inch (45.51 inches).

Format

Byte	Bit	Content	Content Description
0-1		X'0009'	Length of this structure
2		X'81'	Query Reply
3		X'A7'	Paper feed technique
4		**Flags**	
	0-2	B'000'	Unknown
		B'001'	Cut sheet manual
		B'010'	Continuous form
		B'011'	Cut sheet automatic
		B'100'	Document on demand
		Other	Values reserved
	3-7	B'00000'	Reserved
5-6		X'nnnn'	Top margin offset in 1/1440ths of an inch
7-8		X'mmmm'	Bottom margin offset in 1/1440ths of an inch

Additional Content Description

- **Flags**
  - *B'100'* - Document on demand is similar to Continuous forms, but has the characteristic that printing will stop automatically after the current page is ejected. This allows the operator to tear off the current form in a convenient manner. Printing resumes when the operator presses the Enable (Start) key.

- **Bytes 5 and 6** - The top margin offset is the top margin that must be used if the restricted area at the top of the presentation surface is to be avoided. The top margin offset is measured from the top edge of the presentation surface to the base line of the first allowable print line.

  Except when the printer is operating in the 3270 mode, the action taken when printing is attempted in the top restricted area is device dependent (that is, could be forced movement to first "safe" line, attempt to print, reject request, etc.). In effect, to obtain predictable results, the application must avoid attempting to print in the top restricted area.

Other than lines of nulls or lines of spaces, the application may use any of the means provided by the data stream in effect to move through the top restricted area without printing. For example:

- Set the TM parameter of SVF control to a value equal to or greater than the first allowable print line.
- Move from top of the presentation surface to the first allowable print line (or greater) via VTs, NLs or LFs.

When the printer is operating in the 3270 mode (LU3 or non-SNA) the printer must protect against printing in the top restricted area. An application attempt to print in the top restricted area will result in a forced move to the first "safe" print line. The number of lines moved will be the line equivalent of the distance given in the top offset field of the Paper Feed Technique QR; the line counter will be incremented for each line of "forced" movement.

- **Bytes 7 and 8** - The bottom margin offset is the bottom margin that must be used if the restricted area at the bottom of the presentation surface is to be avoided. The bottom margin offset is measured from the bottom edge of the presentation surface to the base line of the last allowable print line.

  The action taken when on an application attempt to print in the bottom restricted area is device dependent (that is, could be forced move to next form, attempt to print, request rejected). In effect, to obtain predictable results, an application must avoid printing in the bottom restricted area.

  Other than lines of nulls and lines of spaces, the application may use any of the means provided by the data stream in effect to avoid printing in the bottom restricted area. For example:

  - Use of an FF (form feed) prior to reaching the bottom restricted area
  - Use of NLs (new lines), LFs (line feeds), or VTs (vertical tabs) to move through the bottom restricted area.

  When operating in 3270 mode the means of avoiding the bottom restricted area is somewhat limited; no VTs or LFs and the FF/NL controls are not always valid. Therefore, the user should ensure that the paper loaded in the printer is long enough to accommodate the complete buffer contents.

Query Reply (Port)

Function

Defines which ports are supported.

For each port supported, an appropriate, separate Query Reply is returned in response to a Read Partition structured field specifying Query List (QCODE List =X'B3' or All).

Format

Byte	Content	Content Description
0-1	L	Length of structured field
2	X'81'	Query Reply
3	X'B3'	Port QCODE
4-n	DATA	Data

Note: For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

Query Reply (Procedure)

Function

Indicates at which graphic subset level graphic procedures are supported.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List =X'B1' or All).

Format

Byte	Content	Content Description
0-1	L	Length of structured field
2	X'81'	Query Reply
3	X'B1'	Procedure
4-n	DATA	Data

Note: For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

Query Reply (Reply Modes)

Function

Transmits the reply modes supported by the device.

The device sends this form of Query Reply when responding to a Read Partition structured field indicating Query.

For a description of reply modes, see the "Set Reply Mode" structured field.

Format

Byte	Content	Content Description
0-1	L	Length of this structure
2	X'81'	Query Reply
3	X'88'	Reply modes
4-n	MODES	Modes supported:
	X'00'	Field mode
	X'01'	Extended field mode
	X'02'	Character mode
other	RES	Reserved

Query Reply (RPQ NAMES)

Function

Tells the application which RPQs are initialized for use in the display. If appropriate, RPQ dependent information is supplied for each.

When this function is supported, this Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'A1', Equivalent, or All).

Format

Byte	Content	Content Description
0-1	L	Length of structured field
2	X'81'	Query Reply
3	X'A1'	RPQ Names
4-7	DEVICE	Device type identifier
8-11	MODEL	Model type identifier (X'00000000' = all models)
12	RPQL	RPQ length (length of RPQ Name + RPQ)
13-n	RPQID	RPQ name

RPQ Query Reply Example

8775 Supporting Only RPQ SU0183

Byte	Content	Content Description
0-1	X'0013'	Length of structured field
2	X'81'	Query Reply
3	X'A1'	RPQ Names
4-7	C'8775'	Device type identifier
8-11	AL4(0)	Model type identifier (X'00000000' = all models)
12	X'07'	Length of RPQ name including this byte
13-18	C'SU0183'	RPQ name

Query Reply (Save/Restore Format)

Function

Indicates that the secondary (device) supports the Save/Restore Format structured field.

When the function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'92', Equivalent, or All).

Format

Byte	Content	Content Description
0-1	X'0006'	Length of this structure
2	X'81'	Query Reply
3	X'92'	Save/restore format
4-5	FPCBL	Format parameter control block length: a binary count that designates the length of the Format Parameter Control Block (FPCB). The FPCB length is fixed for a given implementation, that is, length does not vary with the specific SCS control codes used within a session.

Query Reply (Segment)

Function

Indicates at which graphic subset level graphic segments are supported.

When the function is supported, this form of Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List =X'B0' or All).

Format

Byte	Content	Content Description
0-1	LL	Length of structured field
2	X'81'	Query Reply
3	X'B0'	Segment
4-n	DATA	Data

For the definition of the format and operation of the DATA parameter, refer to the appropriate graphics product publications.

Query Reply (Settable Printer Characteristics)

Function

Indicates support of one or more characteristics such as printer functions and modes that can be set or reset by use of the SPC structured field. Self-defining parameters are used to describe each characteristic supported. This structured field flows inbound in reply to a READ Partition – Query List ("Q" code = X'A9'. or "ALL").

Format

The following table shows the format of this Query Reply.

Byte	Content	Meaning
0–1	L	Length of structure
2	X'81'	Query reply
3	X'A9'	Settable printer characteristics
4–5	**Flags**	Reserved

Early Print Complete Self-defining Parameter

For the Query Reply to be valid, this self-defining parameter must be present.

Byte	Bit	Content	Meaning
0		X'03'	Parameter length
1		X'01'	Early Print Complete
2		**Flags**	
	0 - 1	POC	Printer operator control
		B'00'	No printer operator control
		B'01'	Printer operator control EPC set off
		B'10'	Printer operator control EPS set on
		B'11'	Reserved
	2 - 7	RES	Reserved

This self-defining parameter supports the early print complete mode of operation. When the early print complete mode is set, it allows a 3270 printer to indicate print complete (ready for more data) prior to actual completion of the printing. The overlapping of "load" and "print" can improve throughput for certain situations involving high-speed printers.

Additional Content Description

- **POC** - (printer operator control) indicates whether the implementation provides printer operator control of the set/reset of the EPC mode. It also indicates the operator selection when the query list was received.

Query Reply (Storage Pools)

Function

Identifies the storage pools in the device. For each storage pool, there is a self-defining parameter describing the characteristics of that storage pool. These characteristics are the total size of that storage pool when empty, the amount of that storage pool available for additional objects, and a list of identifiers of the types of objects housed in that storage pool.

When the function is supported, this form of Query Reply structured field is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List =X'96' or All).

Format

Byte	Content	Content Description
0-1	L	Length of structured field
2	X'81'	Query Reply
3	X'96'	Storage pool query code

Self-Defining Parameters

Storage Pool Characteristics

Format

Byte	Content	Content Description
0	L	Length of parameter
1	X'01'	Storage pool characteristics
2	SPID	Storage pool identity
3-6	SIZE	Size of this storage pool when empty
7-10	SPACE	Space available in this storage pool
11-n	OBJLIST	Identifiers of objects housed in this storage pool

Additional Content Description

- **SIZE** – Size is the total size of the storage pool in bytes, for example, the size when it contains no objects.
- **SPACE** – Space is the amount of storage in the pool remaining in bytes, that is, the amount available to house additional objects.
- **OBJLIST** – This is a list of the 2-byte identifiers of the objects that are housed in this storage pool. The following identifiers are from the registry of object identifiers:

Object	Identifier
Segment	X'0001'
Procedure	X'0002'
Extended drawing routine	X'0003'
Data unit	X'0004'
Temporary	X'0005'
Line type	X'0006'
Symbol set	X'0007'

Query Reply (Summary)

Function

Provides a list of the device-supported query replies (QCODES) that may be used by the host in a Read Partition Query List (QCODE List). All of the QCODES that are supported by the device are included in the Summary Query Reply (except the Null Query Reply QCODE).

This Query Reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The Summary Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query, or Query List (QCODE List =X'80', Equivalent, or All).

Format

The table below shows the format of this Query Reply.

Byte	Content	Content Description
0-1	L	The length of this structure
2	X'81'	Query Reply
3	QCODE X'80'	 Summary Query Reply
4-N	LIST	List of supported QCODES

Query Reply (Text Partitions)

Function

Defines the text partition support.

The "maximum partition size" is the guaranteed size of the text buffer. This means that if the host restricts an outbound text structured field to this size, it is guaranteed to fit in the buffer. However, if the outbound text structured field exceeds this size, it does *not* necessarily mean that the transmission will fail.

When the function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'83', Equivalent, or All).

Format

The following table shows the format of this Query Reply.

Byte	Bit	Content	Content Description
0-1		L	Length of structure
2		X'81'	Query Reply
3		X'83'	Text Partitions
4		NT	Maximum number of text partitions
5-6		M	Maximum partition size
7		**Flags**	
	0	B'0' B'1'	Vertical scrolling *not* supported Vertical scrolling supported
	1	RES	Reserved
	2	RES	Reserved
	3	B'0' B'1'	AP addressability *not* supported AP addressability supported
	4	B'0' B'1'	Partition protection *not* supported Partition protection supported
	5	RES	Reserved
	6	B'0' B'1'	Modify partition *not* supported Modify partition supported
	7	RES	Reserved
8		NTT	Number of text types supported
9-N		TLIST X'01'	List of types supported: Type 1 Text Others reserved

Additional Content Description

- **NT** – This specifies the number of text partitions supported. The range of text PIDs supported is 0 to N-1, where N is the total number of partitions supported. (See also "Alphanumeric Partitions Query Reply.")
- **Flags**
  - The scrolling flags indicate whether the device supports vertical scrolling. Support of scrolling implies:
    1. Support for the Set Window Origin structured field with a change of the row.
    2. Possible support of local window movement by the operator; thus the operator may move the window from the host-specified position.
  - The Modify Partition flag indicates whether the device supports the Modify Partition structured field. If the vertical scrolling flag is also set, the Modify Partition structured field can be used to change the window row.
- **TLIST** – is a list of 1-byte values showing which types of text are supported. It has NTT entries, where NTT is the number of text types supported.

Query Reply (Transparency)

Function

Identifies the type of transparency supported by the device.

When the function is supported, this form of the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List =X'A8', or All).

Format

The following table shows the format of this Query Reply.

Byte	Content	Content Description
0-1	L	Total length of this reply
2	X'81'	Query Reply
3	X'A8'	Transparency
4	NP	Number of pairs
5	V(i)	Data stream attribute value accepted.
6	A(i)	Associated action value
7-N		Additional V(i)/A(i) pairs as needed

Additional Content Description

- **NP** - The number of value/action pairs in this Query Reply.
- **V(i)** - The background transparency attribute values valid for this device. For each accepted value the Query Reply returns an action value that indicates the device action associated with that value.
- **A(i)** - Contains the valid action values accepted by the device. The action values currently defined are listed in the A(i) parameter. The device actions currently defined for each action value are as follows:

Action Value	Device Action
X'00'	Default
X'F0'	Normal (Background transparent) - OR
X'F1'	Background is transparent - XOR
X'FF'	Non-transparent (opaque)

An action value of X'00' indicates that the device action for the corresponding attribute value is the same as for the default attribute value.

Query Reply (Usable Area)

Function

Defines the size and other characteristics of the display surface that can be used for partition viewports independent of the data type. One example is the default size of a character cell.

This Query Reply is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix F, "Functions Required for Systems Application Architecture (SAA) Support."

The Usable Area is the portion of the display surface that can be used for partition viewports.

Notes:

1. Character Slot Size — The character slot size, specified in the Character Sets Query Reply, is mapped into the character cell size that is defined in the Usable Area Query Reply.
2. Variable Character Cell Size — If a device supports variable character cell sizes, an explicit partition's character cell size may be defined by the host through use of the Create Partition structured field. In this case, the host-specified value overrides the default character cell size described in the Usable Area Query Reply.
3. Printers — Printers use the fields in the Usable Area Query Reply differently than displays. When a printer uses a field differently than displays, there will be a note to define how the printer uses the field.

If a printer is a page printer, consider the following:

- Because print data is not immediately placed on the paper but resides in a volatile internal storage area, unexpected losses of power may cause loss of printed data that would otherwise have been printed on non-page printers.
- Page printers may have recovery resources that eliminate the need to resend print data after an intervention-required condition.

The Usable Area Query Reply must always be sent inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE List =X'81', Equivalent, or All).

The minimum support for the Usable Area Query Reply consists of the Usable Area Query Reply Base and the self-defining parameters that are appropriate to the device.

The Usable Area Query Reply has three self-defining parameters:

1. On pels Limit
2. Multiple Usable Area
3. Alternate Usable Area.

The inclusion of any of the self-defining parameters requires that the entire Usable Area Query Reply Base be transmitted.

Usable Area Query Reply Base

Format

The following table shows the format for this Query Reply.

Byte	Bit	Content	Content Description
0-1		L	Length of this structure
2		X'81'	Query Reply
3		X'81'	QCODE for the Usable Area Query Reply.
4		**Flags**	
	0	B'0'	Reserved
	1	B'0' B'1'	Non-page printer Page printer
	2	B'0'	Reserved[1]
	3	HC B'0' B'1'	Hard copy: Not a "hard copy" device. A "hard copy" device.
	4-7	ADDR X'0' X'1' X'2' X'3' X'F'	Allowable addressing modes for alphanumeric support of the 3270 data stream Reserved 12/14-bit addressing allowed Reserved 12/14/16- bit addressing allowed Unmapped (no explicit address)
5		**Flags**	
	0	VCP B'0' B'1'	Variable cells: Variable cells not supported Variable cells supported
	1	CHAR B'0' B'1'	Characters: Matrix character Non-matrix character
	2	CELLUNITS B'0' B'1'	Cell Units: Value in bytes 6 and 7, 8 and 9; cells Value in bytes 6 and 7, 8 and 9; pels
	3-7	B'00000'	Reserved
6-7		W	Width of usable area in cells/pels
8-9		H	Height of usable area in cells/pels
10		UNITS X'00' X'01'	Units of measure of pels / in. or mm. Inches Millimeters
11-14		Xr	Distance between points in X direction as a fraction, measured in UNITS, with 2-byte numerator and 2-byte denominator

Byte	Bit	Content	Content Description
15-18		Yr	Distance between points in Y direction as a fraction, measured in UNITS, with 2-byte numerator and 2-byte denominator
19		AW	Number of X units in default cell
20		AH	Number of Y units in default cell
21-22		BUFFSZ	Character buffer size (bytes)
23		XMIN	Minimum number of X units in variable cell (present if VCP flag set)
24		YMIN	Minimum number of Y units in variable cell (present if VCP flag set)
25		XMAX	Maximum number of X units in variable cell (present if VCP flag set)
26		YMAX	Maximum number of Y units in variable cell (present if VCP flag set)
[1] A few early products set this bit to B'1' to indicate that the Outbound 3270DS and Erase/Reset structured fields were not supported.			

Note: When omitting the BUFFSZ and/or the XMIN, YMIN, XMAX, YMAX parameters from the Usable Area Query Reply Base, the following rules must be observed:

- Bytes 0 through 20 must always be included in the Usable Area Query Reply Base.
- Bytes 21 through 26 (parameters BUFFSZ and XMIN, YMIN, XMAX, YMAX) must always be included when there is any self-defining parameter included with the Usable Area Query Reply Base. If the parameters are not applicable, then zero values are used for the parameters.
- If there are no self-defining parameters and no variable character cell parameters, then the XMIN, YMIN, XMAX, YMAX parameters may be omitted. The parameters XMIN, YMIN, XMAX, YMAX are treated as one parameter (that is, if you have a value for XMIN, then you have values for YMIN, XMAX, YMAX).
- If there are no self-defining parameters, no variable character cell parameters, and the BUFFSZ parameter is not applicable, then both the BUFFSZ and the XMIN, YMIN, XMAX, YMAX may be omitted.

Additional Content Description

- **HC** - Indicates that this device is a “hard copy” device (a printer).
- **ADDR** - Indicates the addressing modes that the device can support. The following lists the valid values and the description of each value. All other values are reserved.
  1. X'0' - Reserved. This value was used by the 8100 to indicate 12-bit addressing. All displays and printers must support 12/14-bit addressing mode.

2. X'1' - 12/14-bit addressing allowed. When the 12/14-bit addressing mode (outbound) is specified for a partition (Implicit Partition zero is always set to 12/14-bit addressing), bits 0 and 1 of the first address byte following the 3270 order are flag bits and have the following significance:

   - B'00' - 14-bit binary address follows.

     The next 14 bits (the remainder of this byte and the full 8 bits of the next byte) contain the buffer address in binary form. No address translation is necessary.

   - B'01' - 12-bit coded address follows.

     The next 14 bits are to be interpreted as a 3270 coded address (6 bits in each byte). The second byte will have the first 2 bits as B'01' or B'11'. Otherwise, the data stream will be rejected.

   - B'10' - Reserved.

     Receipt of a buffer address beginning with the flag bits of B'10' will cause the data stream to be rejected.

   - B'11' - 12-bit coded address follows.

     The next 14 bits are to be interpreted as a 3270 coded address (6 bits in each byte). The second byte will have the first 2 bits as B'01' or B'11'. Otherwise, the data stream will be rejected.

   The rules for an inbound data stream with the 12/14-bit addressing mode are a function of the partition size and are as follows:

   - If the partition size is greater than 4096 characters, then all of the addresses are in the 14-bit form.

   - If the partition size is equal to or less than 4096 characters, then all of the addresses are in the 12-bit form.

3. **X'2'** - Reserved. This value was used to indicate 16-bit addressing only.

4. **X'3'** - 12/14/16-bit addressing allowed. This value is used to support partitions that use the 16-bit addressing. The 16-bit addressing is interpreted as binary values.

   For 12/14-bit addressing see X'1' above.

5. **X'F'** - Unmapped (no explicit address). This value is used to indicate a non-3270 data stream addressing mode (for example, like SCS for printers).

- **VCP** - Indicates that the device can support different character cell sizes in each partition. Thus, VCP must be B'0' unless the device also supports the Create Partition structured field. If VCP is set to B'1', then a character cell size may be specified in the Create Partition structured field when each partition is created. The XMIN, YMIN, XMAX and YMAX parameters, of this Query Reply, provide the minimum and the maximum valid values for the character cell sizes.

- **CHAR** - Indicates whether matrix or non-matrix type characters are supported. (An example of a non-matrix type character would be found on a belt printer.) If this bit is set to B'1', bytes 10 through 20 are not applicable and must be set to zeros.

- **CELLUNITS** - Indicates the unit of measure (character cells or pels) of the W and H parameters:
  - If Byte 5, bit 2 (CELLUNITS) = B'0', the values in W and H indicate Usable Area width and height in units of character cells.
  - If Byte 5, bit 2 (CELLUNITS) = B'1', the values in W and H indicate Usable Area width and height in units of pels.
- **W and H** - The values in W and H define the size of the usable area for the base display surface.

  Thus, the size of the Usable Area in ***pels*** is (W x AW) by (H x AH) when Byte 5, bit 2 is B'0' (character cells), or W x H when Byte 5, bit 2 is B'1' (PELs).

  Thus, W x H defines the **maximum** number of characters that can be presented in the Usable Area, when characters from a nonloadable alphanumeric character set are written with the 3270 data stream.

  Note: For printers:

  - For an LU Type 1 printer, W and H are the Maximum Print Position (MPP) and the Maximum Print Line (MPL) respectively supported by the implementation. These values indicate the maximum MPP and MPL that the host may set using the SHF(MPP) and the SVF(MPL) SCS controls. The values reported in the W and H fields are not affected by operator set values on printers that provide operator control (for example, switches). A host selection using SHF or SVF overrides an operator selection.
  - For LU Type 3 and non-SNA 3270 printers, the W and H fields have no meaning and are ignored by the host. The W and H fields should be set to zeros.
- **UNITS** - indicates the unit of measure (inches or millimeters) of the Xr and Yr parameters.
- **Xr and Yr** - gives the distance between the "pel centers" in fractions of the units specified by the UNITS parameter (inches or millimeters). When determined, the numerator and denominator values for Xr and Yr are expressed in a 16-bit (2-byte), positive, hexadecimal number.

  If a value for Xr and/or Yr cannot be determined, then the Xr and Yr will be set to X'FFFFFFFF' (Xr = X'FFFF' and Yr = X'FFFF'). If the device supports the Load Programmed Symbol Set structured field, then the values for Xr and Yr must not be X'FFFFFFFF'.

  For example, if the device has 72.5 pels/inch horizontally, and 69 PELs/inch vertically, then the values returned would be:

  UNITS - X'00'

  Xr - X'00020091' (2/145 inch)

  Yr - X'00010045' (1/69 inch).
- **AW and AH** - Define the default character cell size for the device. If the device supports the Load Programmed Symbol Set structured field and the size of the character cell used for either the default or alternate size of Implicit Partition Zero is different than (AW, AH), then the Character Cell Dimensions parameter must be returned in the Implicit Partitions Query Reply. If a partition is created without specifying the character cell dimensions (PW, PH), the character cell used by the display must be of size (AW, AH).

- **BUFFSZ** - Defines the amount of character buffer available. It applies only to a device that does not support partitions. If the device supports partitions, the Alphanumeric Partitions Query Reply will define the buffer size, and the BUFFSZ parameter will be set to zero.

  Note: For printers:

  - For LU Type 1, the BUFFSZ parameter is not applicable and should be set to X'0000'.
  - For LU Type 3 and non-SNA 3270 Printers, the Implicit Partition Query Reply provides the buffer wrap points associated with the default and alternate buffer sizes. The BUFFSZ parameter is redundant. The value used in BUFFSZ is the same as that in the Implicit Partition Query Reply for Alternate Printer Buffer Size (APBS).

- **XMIN, YMIN, XMAX, YMAX** - Define the limits of character cell size that can be specified in the Create Partition structured field when the device supports different size cells in each partition.

On pels Limit Self-Defining Parameter

Function

Informs the host application of the percentage of pels that can be on at the same time.

If the device has no limit (100% of the pels on at the same time), then the device does not have to send this self-defining parameter.

The On pels Limit self-defining parameter is required when the device has a limit for how many pels may be "on" at the same time.

Format

Byte	Content	Content Description
0	X'03'	Length of self-defining parameter
1	X'01'	"On" pels limit
2	PERCENT	Percent of pels allowed on at a time

Addition Content Description

- **PERCENT** - Indicates the percentage of the available pels, reported in the Usable Area Query Reply Base, that may be on at the same time.

  The PERCENT parameter gives the percentage as a hexadecimal value. Valid values are X'01' through X'64' (1 through 100 percent).

  For example, if the limit is 31%, the value of the parameter is X'1F'.

  Exceeding the reported limit may result in the loss of displayed information.

Multiple Usable Area Self-Defining Parameter

Function

Informs the host application of the presence of multiple (two or more), separate and independent display surfaces and what data types will be viewed on each surface. If the device has only a single display surface, then the Multiple Usable Area self-defining parameter is not returned to the host application, and all of the data types supported by the device will be displayed on the single display surface.

When the Multiple Usable Area self-defining parameter is included in the Usable Area Query Reply, the following apply:

- The existence of multiple display surfaces has no effect on the host data stream.
- The Usable Area Query Reply Base will report the usable area for alphanumeric (A/N) data and every data type not reported in a Multiple Usable Area self-defining parameter.
- Each and every instance of the Multiple Usable Area self-defining parameter must have only one unique data type (DATATYPE) parameter.
- The routing of data to the multiple display surfaces is done by data type. Therefore, each data type can only occur once per logical terminal.
- When a partition is created in the Multiple Usable Area environment, the device creates the viewports for each of the display surfaces. The viewport for the Alphanumeric partition is defined (size and position) in the Create Partition structured field. The size and position of the viewports for the other display surfaces is determined by the device using the data in the Create Partition structured field and the data in the Usable Area Query Reply.

The Multiple Usable Area self-defining parameter is always required when the device supports multiple, separate, and independent display surfaces.

Note: Printers cannot use the Multiple Usable Area self-defining parameter.

Format

Byte	Bit	Content	Content Description
0		X'13'	Length of self-defining parameter
1		X'03'	Multiple Usable Area
2		X'00'	Reserved - must be set to zero
3		Flags	
	0-1	B'00'	Reserved
	2	CELLUNITS B'0' B'1'	Cell units Value in bytes 4-5, 6-7 in cells Value in bytes 4-5, 6-7 in pels
	3	B'0'	Reserved
	4-7	DATATYPE B'0001' B'0010' All Others	Data types presented on Usable Area. Reserved. [1] Graphic data type Reserved
4-5		WG	Width of Multiple Usable Area
6-7		HG	Height of Multiple Usable Area
8		UNITS X'00' X'01'	Units of measure of pels/ in. or mm. Inches Millimeters
9-12		Xr	Distance between points in X direction as a fraction measured in UNITS, with 2-byte numerator and 2-byte denominator
13-16		Yr	Distance between points in Y direction as a fraction measured in UNITS, with 2-byte numerator and 2-byte denominator
17		AW	Number of X units in default cell
18		AH	Number of Y units in default cell

[1] Previously, this value was used to indicate alphanumeric data type.

Additional Content Description

- **CELLUNITS** -Indicates the unit of measure (character cells or pels) of the WG and HG parameters (refer to the Usable Area Query Reply Base CELLUNITS parameter).
- **DATATYPE** - Indicates which type of data is presented on this multiple display surface.
- **WG and HG** - Defines the size of the usable area for this multiple display surface (refer to the Usable Area Query Reply Base W and H parameters).
- **UNITS** - Indicates the unit of measure (inches or millimeters) of the Xr and Yr parameters.
- **Xr and Yr** - Gives the distance between the "pel centers" in fractions of the units specified by the UNITS parameter (inches or millimeters). The numerator

and denominator values for Xr and Yr are expressed in a 16-bit (2-byte), positive, hexadecimal number.

If Xr and Yr cannot be determined, X'FFFFFFFF' will be returned for both. Xr and Yr must not be X'FFFFFFFF' for a device that supports the Load Programmed Symbol Set structured field. (Refer to the Usable Area Query Reply Base Xr and Yr parameters.)

- **AW and AH** - Defines the default character cell size for this multiple display surface. (Refer to the Usable Area Query Reply Base AW and AH parameters.)

Alternate Usable Area Self-Defining Parameter

Function

The Alternate Usable Area self-defining parameter is present only for devices that support more than one usable area size on one display surface.

EXAMPLE - A physical device such as the 3180 has the capability to display up to 43 rows and up to 132 columns, but not concurrently. The device could report 43 rows and 80 columns (Model 4) in the Usable Area Query Reply Base and 27 rows and 132 columns (Model 5) in the Alternate Usable Area self-defining parameter.

The Alternate Usable Area self-defining parameter is limited to a single partition implementation. When the Alternate Usable Area self-defining parameter is used and the device receives a Create Partition structured field, the following apply:

- If the defined viewport fits within the usable area defined in the Usable Area Query Reply Base, then that usable area is used.
- If the defined viewport does not fit in the usable area defined in the Usable Area Query Reply Base, then the usable area defined in the Alternate Usable Area self-defining parameter is tried. If the viewport then fits, that usable area is used.
- If the viewport does not fit in either usable area, then the Create Partition structured field is rejected.

Note: The Alternate Usable Area self-defining parameter cannot be used for printers.

The Alternate Usable Area self-defining parameter is always required when the device supports more than one usable area size on one display surface.

For devices that support the Load Programmed Symbol Set structured field, bytes 8 through 18 must be supplied so that the host application can determine the correct character cell size.

Format

Byte	Content	Content Description
0	X'13'	Length of self-defining parameter
1	X'02'	Alternate usable areas
2	X'00'	Reserved - must be set to zero
3	AUAID X'01' X'FE'	Alternate Usable Area ID Identifier for the Alternate Usable Area Reserved values.
4-5	WAUAi	Width of AUAi
6-7	HAUAi	Height of AUAi
8	AUAUNITS X'00' X'01'	Units of measure Inches Millimeters
9-12	AUAXr	Distance between points in X direction as a fraction, measured in AUAUNITS, with 2-byte numerator and 2-byte denominator.
13-16	AUAYr	Distance between points in Y direction as a fraction, measured in AUAUNITS, with 2-byte numerator and 2-byte denominator
17	AWAUAi	Number of X units in default cell
18	AHAUAi	Number of Y units in default cell

Additional Content Description

- **WAUAi and HAUAi**- Define the size of the usable area for the alternate display surface. (Refer to the Usable Area Query Reply Base W and H parameters.) The units of WAUAi and HAUAi are the CELLUNITS defined in the Usable Area Query Reply Base.

- **AUAUNITS** - Indicates the unit of measure (inches or millimeters) of the AUAXr and AUAYr parameters.

- **AUAXr and AUAYr** - Give the distance between the 'pel centers' in fractions of the units specified by the AUAUNITS parameter (inches or millimeters). The numerator and denominator values for Xr and Yr are expressed in a 16 bit (2 byte), positive, hexadecimal number.

  If the device is unable to determine Xr and Yr, X'FFFFFFFF' will be returned for both. Xr and Yr must not be X'FFFFFFFF' for a device that supports the Load Programmed Symbol Set structured field. (Refer to the Usable Area Query Reply Base Xr and Yr parameters.)

- **AWAUAi and AHAUAi** - Defines the default character cell size for the alternate display surface. (Refer to the Usable Area Query Reply Base AW and AH parameters.)

Query Reply (3270IPDS)

Function

Indicates support of the Intelligent Printer Data Stream (IPDS) through the 3270 data stream (non-SNA).

When this function is supported, the Query Reply is transmitted inbound in reply to a Read Partition structured field specifying Query or Query List (QCODE = X'9A', Equivalent or All).

Format

Byte	Content	Content Description
0-1	X'0008'	Length of structure
2	X'81'	Query Reply
3	X'9A'	3270IPDS
4-5	Flags	Reserved
6-7	TRANLIM	Maximum transmission size allowed outbound. specifies the maximum number of bytes of IPDS data allowed in an outbound transmission. A value of X'0000' indicates no limit specified.

Chapter 7. Magnetic-Reader, Keyboard, and Selector Pen Operations

Magnetic-Reader Operations

This chapter describes the magnetic-reader operations for the magnetic slot reader (MSR). This description also applies to the magnetic hand scanner (MHS). Two modes of magnetic-reader operations are described: the 3275/3277-compatible mode and the numeric/alphanumeric mode.

3275/3277-Compatible Operation

Following is the numeric character set used for the 3275/3277-compatible mode operation.

Stripe Codes		**Display-Generated Codes**		
Character	*Hex*	*EBCDIC*	*ASCII*	
0	0	F0	30	
1	1	F1	31	
2	2	F2	32	
3	3	F3	33	
4	4	F4	34	
5	5	F5	35	
6	6	F6	36	
7	7	F7	37	
8	8	F8	38	
9	9	F9	39	
	A	7A	3A	Operator ID (OID)
	B	7B	23	Start of record (SOR)
	C	7C	40	unassigned
	D	7D	27	Field separator
	E	7E	3D	unassigned
	F	7F	22	End of record (EOR)

Magnetic stripes may contain up to 128 characters. This includes the start of record (SOR), which does not appear in the input data stream. Also included are the end of record (EOR) and the longitudinal redundancy check (LRC), which do appear in the input data stream. Note that the application program must do the LRC verification. (It is recommended that the application program verify that a X'F7' appear just prior to the LRC.)

The data portion of the stripe, the part between the initial start of record and the end of record (EOR), should contain only 0-9, *D* (field separator), or the OID (which is the *A* immediately following the start-of-record indicating secure data). The secure data is for the application program's use; the display treats all stripes as secure. Stripes containing *A* (other than special A), *B*, *C*, or *E* may be rejected. However, if accepted, the *A, B, C,* and *E* will be sent to the application program as 7A, 7B, 7C, and 7E, respectively. The graphics associated with *A*, *B*, *C*, and *E* are whatever the attaching device associates with the resulting EBCDIC or ASCII codepoint.

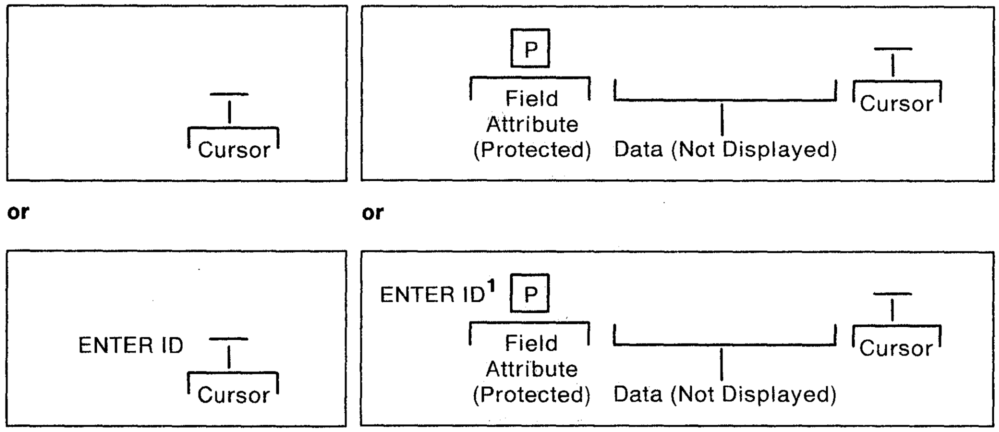
When the start-of-record character is read from the stripe, an attribute character is entered automatically into the cursor-identified location of the buffer, provided the cursor is at an unprotected character location. This attribute character defines the stripe data field following it as protected, alphanumeric, nondisplay/nonprint. As the rest of the stripe data is read into the buffer, it is stored, starting at the first character location after this attribute character. As each character is stored in the buffer, the cursor advances one buffer location. Note that the cursor may not be visible to the operator until the end of the operation. A magnetic-reader operation automatically causes the keyboard to lock and a magnetic-reader AID to become pending. At an appropriate time the AID character and the display buffer data are sent to the application program during a read-modified operation.

No additional keyboard data can follow a magnetic-reader operation. There are some differences in the data-stream content, depending on whether the device buffer is formatted or unformatted.

- Unformatted display. When a magnetic-reader input operation occurs, the device buffer becomes formatted because of the insertion of the protected, nondisplay attribute with the MDT bit set on. Any data that was previously displayed will no longer be displayed. (See Figure 7-1 on page 7-3.)

- Formatted display. A magnetic-reader operation may be initiated only when the cursor is located in an unprotected field. The input data stream resulting from a magnetic reader operation will contain at least two fields with the MDT bit set on (new data field and the previous data field) because all the information from the stripe is treated as data until the information has been entered. Also, the MDT bit is set on in the attribute character created by the start of record. Figure 7-2 and Figure 7-3 show how to prepare an unprotected field for magnetic-reader input. In Figure 7-2 the cursor follows the operator instructions. In Figure 7-3 the instructions are in a protected field and the cursor follows an unprotected field attribute character.

When the extended field and character attributes are supported, a magnetic-reader operation will cause an extended field attribute of X'00' to be generated in addition to the field attribute. Data characters will generate character attributes of X'00'.



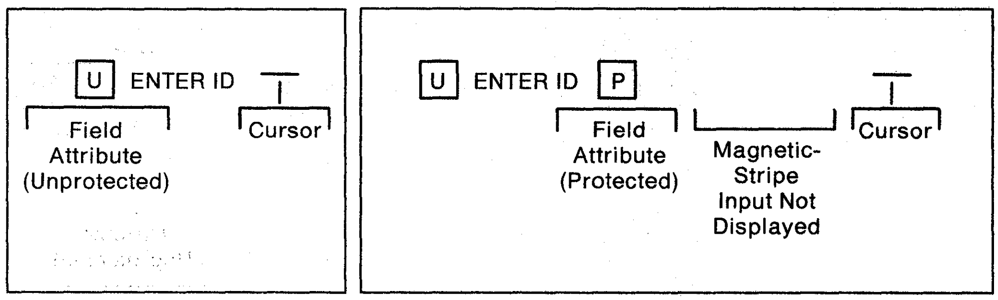
Display Screen before Magnetic-Stripe Input is Entered

Display Screen after Magnetic-Stripe Input is Entered

AID	Set to indicate magnetic-stripe input.
Cursor Address	Address of the cursor upon completion of the operation.
SBA	Address of the first data character following the field attribute.
Field Attribute Address + 1	The magnetic-stripe data followed by any information present in the buffer. The additional information could have been initiated by the application program as ENTER ID, as shown in the example, or entered by the operator before the magnetic-stripe operation was started.
Data	Note that, with an unformatted screen, the magnetic-stripe data is the first text in the data stream presented to the application program.

[1]The ID is not displayed, because it is within a nondisplay field, defined by the magnetic-stripe field attribute.

Figure 7-1. Display Screen Activity before and after Magnetic-Stripe Input (Unformatted Display)

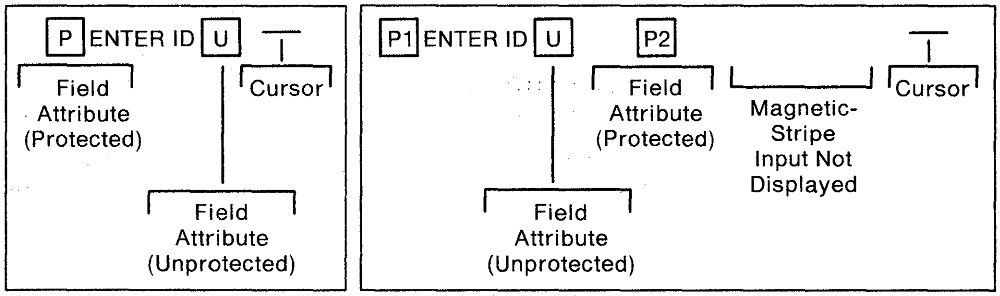


Display Screen before Magnetic-Stripe Input is Entered

Display Screen after Magnetic-Stripe Input is Entered

AID	Set to indicate magnetic-stripe input.
Cursor Address	Address of the cursor upon completion of the operation.
SBA	
Field Attribute Address + 1	Address of the unprotected (U) field attribute + 1.
Data	ENTER ID, in the example above.
SBA	
Field Attribute Address + 1	Address of the protected field attribute + 1. In this case, the address of the first data character from the magnetic stripe input following the field attribute.
Data	The magnetic-stripe input (and any data between the cursor and the next field attribute).

Figure 7-2. Display Screen Activity before and after Magnetic-Stripe Input (Formatted Display with Unprotected Field Attribute)



Display Screen before Magnetic-Stripe Input is Entered

Display Screen after Magnetic-Stripe Input is Entered

AID	Set to indicate magnetic-stripe input.
Cursor Address	Address of the cursor upon completion of the operation.
SBA	
Field Attribute Address + 1	Address of the unprotected (U) field attribute + 1. In the example above, it is the address of field attribute P2.
SBA	
Field Attribute Address + 1	Address of the P2 field attribute + 1. In this case, the address of the first data character from the magnetic stripe input following the field attribute.
Data	The magnetic-stripe input (and any data between the cursor and the next field attribute).

Figure 7-3. Display Screen Activity before and after Magnetic-Stripe Input (Formatted Display with Protected Field Attribute)

Numeric/Alphanumeric Operation

For magnetic stripes using the numeric/alphanumeric character sets, the two character positions following the start sentinel (SS) perform a header function that designates to the display the data field contents, as follows:

, = numeric-character set, nonsecure data
'A',* = numeric-character set, secure data
'C','A' = numeric/alphanumeric-character set, nonsecure data
'A','C' = numeric/alphanumeric-character set, secure data.

Note: * = 0 through 9 or D.

Any magnetic stripes not conforming to the above header information will be rejected; the information will not be sent to the application. Also, if a display does not support a particular character set, cards with magnetic stripes using that character set will be rejected.

For the alphanumeric character set magnetic stripes, the data field starts after the 2-character header. For the secure numeric character set magnetic stripes, the data field starts after the *A* character. For nonsecure numeric character set magnetic stripes, the data field starts after the SS *B*. For the numeric/alphanumeric character set magnetic stripes, the data field ends with (but does not include) the EOC (*F*). Only the data field is sent to the application program. The LRC check is done by the display. The LRC and the end sentinel are not sent to the application program.

The data field, up to 125 characters less the header, is sent to the application program in conformance with the magnetic stripe character set designated by the header (that is, the numeric or alphanumeric character set).

Stripe Codes and Application Program Codes

The following shows the relationship of the magnetic stripe code to the codepoint the display generates. This relationship is independent of the graphic character set supported by the attaching display, for example, the same codepoint is generated for the US EBCDIC character set displays as would be generated for a German character-set display. The graphic displayed when display is allowed depends on the graphic character sets supported by the attaching display.

Following is the magnetic-stripe numeric character set:

Stripe Codes		Display-Generated Codes	
Character	*Hex*	*EBCDIC*	*ASCII*
0	0	F0	30
1	1	F1	31
2	2	F2	32
3	3	F3	33
4	4	F4	34
5	5	F5	35
6	6	F6	36
7	7	F7	37
8	8	F8	38
9	9	F9	39

Control Data Characters	
Hexadecimal	
A (secure data)	not sent in the data stream
B (start sentinel)	not sent in the data stream
C (reserved)	not sent in the data stream
D (space)	40
E (reserved)	not sent in the data stream
F (end sentinel)	not sent in the data stream

The control characters *A* and *B* must appear in their respective positions on the magnetic stripe or they will be treated as errors. The control characters *C* And *E* will always be treated as an error. The *A* means secure data (protected, nondisplay, and nonprint) when located immediately after the SS. The *B* is SS. The *F* is the end sentinel (ES).

Another SS, called *reverse start sentinel (RSS)*, follows the LRC on the magnetic stripe. The RSS is for magnetic stripe card readers that can scan the card in both directions. The *B* is also used for the reverse start sentinel. This type of operation is primarily for magnetic-hand-scanner operations.

Following is the alphanumeric character set:

Stripe Codes	Display-Generated Codes	
Character	*EBCDIC*	*ASCII*
0A[1]	F0	30
1A[1]	F1	31
[1]Indicates that *A* is a filler character when numerics are not paired.		

Stripe Codes	Display-Generated Codes	
Character	*EBCDIC*	*ASCII*
2A[1]	F2	32
3A[1]	F3	33
4A[1]	F4	34
5A[1]	F5	35
6A[1]	F6	36
7A[1]	F7	37
8A[1]	F8	38
9A[1]	F9	39
C1	C1	41
C2	C2	42
C3	C3	43
C4	C4	44
C5	C5	45
C6	C6	46
C7	C7	47
C8	C8	48
C9	C9	49
D1	D1	4A
D2	D2	4B
D3	D3	4C
D4	D4	4D
D5	D5	4E
D6	D6	4F
D7	D7	50
D8	D8	51
D9	D9	52
E2	E2	53
E3	E3	54
E4	E4	55
E5	E5	56
E6	E6	57
E7	E7	58
E8	E8	59
E9	E9	5A
0C	4A	5B

[1]Indicates that *A* is a filler character when numerics are not paired.

Stripe Codes	Display-Generated Codes	
Character	*EBCDIC*	*ASCII*
1C	5A	5D
3C	7A	3A
4C	4C	3C
5C	5C	2A
6C	6C	25
7C	7C	40
0D	4B	2E
1D	5B	24
2D	6B	2C
3D	7B	23
4D	4D	2B
5D	5D	29
6D	6D	5F
7D	7D	27
0E	4F	21
1E	5F	5E
2E	6F	3F
3E	7F	22
4E	4E	2B
5E	5E	3B
6E	6E	3B
7E	7E	3D
E0	E0	5C
E1	61	2F
DA	50	26
EA	60	2D
CA	40	20
00	F0,F0	30,30
01	F0,F1	30,31
|	|	|
|	|	|
|	|	|
|	|	|
|	|	|
|	|	|

[1]Indicates that *A* is a filler character when numerics are not paired.

Stripe Codes	Display-Generated Codes	
Character	*EBCDIC*	*ASCII*
89	F8,F9	38,34
99	F9,F9	39,39
[1]Indicates that *A* is a filler character when numerics are not paired.		

When using the alphanumeric character set, the numerics are coded as two 4-bit numeric characters for each 8-bit byte. As a result, there must be either an even number of numerics in any continuous string of numerics, or an odd number of numerics with a filler character. The code sequence (12XYZ) is represented on a card as (12E7E8E9). The code sequence (123XYZ) is represented on a card as (123AB7E8E9). This limits the number of characters on a card to 62 alpha and special characters and to 124 numeric characters. The hexadecimal *A* (1010) is used as the filler character.

The initial *B* is start sentinel (SS). The *F* is the end sentinel (ES). The reverse start sentinel (RSS) follows the LRC on the magnetic stripe card. The RSS is used by devices using a magnetic stripe card reader that can scan the card in either direction.

Any stripe code not shown (except for CAEE—the test card) will cause a rejection of the card when that code appears between the header and the end sentinel. The first *F* encountered after the start sentinel will be treated as the end sentinel. The code following the ES is considered the LRC.

Secure/Nonsecure Magnetic-Stripe Cards

Display operations when using the numeric/alphanumeric character set magnetic readers closely parallels the operation of the 3275/3277-compatible operation as far as the application program is concerned.

There are differences in the handling of secure and nonsecure cards.

Secure Magnetic Stripe Cards

Handling of the numeric/alphanumeric magnetic-stripe cards designated as secure follows that of the 3275/3277-compatible operation. That is, the same rules for the cursor being in an unprotected field and generation of a protected, nonprint/nondisplay field attribute and automatic entry all apply. An AID code (E7 for EBCDIC, 58 for ASCII) is generated indicating magnetic reader extended. This informs the application program that the magnetic stripe data conforms to the numeric/alphanumeric character set operation. Note that no indication is given on whether the data field is in the numeric or alphanumeric character set. No end sentinel or LRC is sent with the data. When the application receives the E7,(58), AID, the data is valid; the LRC check has already been done.

Nonsecure Magnetic-Stripe Cards

Nonsecure numeric/alphanumeric character-set magnetic-stripe cards appear to the application program as keyboard entered data. An exception is that the magnetic-stripe data does not follow the keyboard rules relative to numeric field/numeric lock/keyboard type. Nonsecure magnetic-stripe data may be entered in any unprotected field. The cursor must still be in an unprotected field for a successful magnetic-reader operation. However, no protected, nonprint/nondisplay field attribute is generated. No AID is generated, and no automatic entry will take place. Sending of the data relies on normal entry operation—the ENTER key, selector pen attention, and so on. An implementation may optionally provide for a customer selected override that causes auto entry of nonsecure data. However, the AID generated will be coded as the ENTER key.

Test Card

For displays, any nonsecure card of known content may be used as a test card. The alphanumeric test card (CAEE—) will be treated as a normal nonsecure card, except that the EE will be deleted. There will be no automatic verification of the test data. Visual verification will be done by the operator. All magnetic-reader data will be inhibited from being sent to the application program when the display is in subsystem test mode.

Keyboard Functions

The keyboard enables the operator to change, edit, or create data except within fields defined as protected from keyboard operations. As data is being composed or modified by keyboard operations, the changes are inserted in the character buffer and displayed on the screen.

Keys that Affect the Data Stream

The data stream is oriented toward a keyboard display, but implementations have the option of providing or not providing a keyboard. If supported, the key functions and key positions depend on the type of keyboard and the keyboard definition provided by the user. The keyboard description that follows is limited to the functions that have an effect on the data stream.

Cursor

A special symbol, called a *cursor*, indicates where the next character entered from the keyboard will be displayed on the display surface (viewport). For example, assume the cursor character is an underscore. When the cursor is displayed under one character in a row of characters, that character can be changed or deleted by keyboard action. Also, if the cursor is displayed under a position without a character, a character can be placed in that position by keyboard action. All these operations, when performed on a field of a formatted display, cause the MDT bit (bit 7) of the field attribute for the field to be set to 1. However, when the cursor appears beneath a character in a protected field or beneath a field attribute, that position cannot be modified by keyboard action, and the MDT bit is not set.

When the display is turned on, the cursor is usually in the first location on the screen. It may be moved by an application. The cursor can be repositioned by the keyboard operator and also by the application program. The cursor is not affected by field attributes; it is displayed even when positioned in a nondisplay/nonprint field.

Alphanumeric Data

Alphabetic characters can be entered into the display buffer in the code for either uppercase or lowercase, depending on the position of the shift key. Only uppercase alphabetic codes can be entered from some keyboards.

Keyboard entry of an alphanumeric character into the buffer occurs at the cursor location, provided that the cursor is located in a character location within an unprotected field. (An attempt to enter an alphanumeric character into a protected field or into a field attribute location is blocked.) Successful keyboard entry of the alphanumeric character causes the cursor to advance to the next character location within the unprotected field.

Automatic Skip

Upon entry of a character into the last character location of an unprotected field, the cursor is repositioned according to the field attribute describing the next field.

If the field attribute defines the next field as (1) alphanumeric, or (2) numeric and unprotected, the field attribute is skipped and the cursor is positioned to the first character location in that field.

If the field attribute defines the field as numeric and protected, the cursor skips that field and is positioned to the first character location of the next unprotected field.

Erase to End of Field (ERASE EOF) Key

If the cursor is located in an alphanumeric character location in an unprotected field, this key clears the character location occupied by the cursor and all remaining character locations in that field to nulls. Each character attribute associated with the nulled characters is set to its default value.

If the display is unformatted (that is, there are no field attributes on the display), the character buffer from the cursor address to the end of the screen is set to nulls and the corresponding character attributes are set to their default values.

The operation can wrap from the end of the last row of the display to the beginning of the top row. The cursor does not move as a result of operating this key, and the MDT bit is set to 1.

When the cursor is in a field attribute location or is within a protected field, pressing this key locks the keyboard and notifies the terminal operator of an input-inhibit condition. No character locations are cleared, the cursor is not moved, and the MDT bit is not set.

ERASE INPUT Key

This key clears all unprotected character locations to nulls, resets the MDT bit to 0 in unprotected fields, sets all character attributes to their default value, and repositions the cursor to the first unprotected character location on the display.

In a buffer with only protected fields, no character locations are cleared and the cursor is repositioned to buffer address 0.

If the display contains no fields, the entire character buffer is cleared to nulls, the cursor is repositioned to character location 0, and all character attributes are set to their default value.

Insert Mode Key

This key turns on the insert-mode indicator on the display (if one exists) and places the display in an insert mode of operation, regardless of the cursor location.

If the cursor is located in an unprotected field that has a null character either in the character location identified by the cursor or in any character location in the field beyond the cursor, pressing an alphanumeric key causes that alphanumeric character to be entered at the cursor position and the MDT bit to be set to 1. The character formerly occupying the cursor location and all remaining characters within the field (except for null characters or characters to the right of null characters) are shifted one character location to the right. If more than one row of characters is contained within the field, a character occupying the last character location in the row is shifted into the first character location of the next row. If the location identified by the cursor at the time of the insert operation is a null, no character shifting occurs.

The character attributes of any shifted characters do not change. Each character attribute associated with the inserted character is set to its default value. In all cases, the programmed symbol (PS), color, and highlighting select keys (if available on the keyboard, and allowed by the Set Reply Mode structured field) set the attribute values associated with the inserted character as it is stored.

After all null characters at or beyond the cursor location in the field have been overwritten, or if there were no null characters, pressing an alphanumeric key locks the keyboard and notifies the terminal operator of an input-inhibit condition. The field attribute and the extended field attribute remain unchanged.

Pressing an alphanumeric key while in insert mode when the cursor is located in a field attribute location, or is within a protected data field, locks the keyboard and notifies the operator of an input-inhibit condition. No character locations are cleared, the cursor is not moved, and the MDT bit is not set.

Operation of the RESET key returns the keyboard from insert-mode to normal mode.

Delete Key

If the cursor is located in an alphanumeric character location in an unprotected field, pressing the delete key deletes the character from the character location occupied by the cursor and sets the MDT bit to 1. The cursor does not move. All remaining characters in the unprotected field (to the right of the cursor and on the same row) are shifted one character location to the left. The vacated character location at the end of the row is filled with a null and the default character attributes. If the unprotected field encompasses more than one row, characters in rows other than the row identified by the cursor are not affected.

The character attributes of any shifted characters do not change.

Operation of this key when the cursor is located in a field attribute location or is within a protected field disables the keyboard; no character locations are cleared, the cursor is not moved, and the MDT bit is not set.

Duplicate (DUP) Key

Operation of this key causes a X'1C' code to be entered into the presentation space, a Tab key operation to be performed, and the MDT bit to be set to 1. The operator uses the duplicate key to tell the application program that a duplicate operation is indicated for the rest of the field in which it is located. The character transferred to the application program is a X'1C' (EBCDIC) and is sent when the data is read. No duplicate operation is performed at the display. The duplicate character, when stored in the character buffer, is displayed as asterisk-overscore. Display devices operating in monocase mode display this character as an asterisk.

Operation of this key when the cursor is positioned at a field attribute location or is within a protected field disables the keyboard; no character locations are cleared, the cursor is not moved, and the MDT bit is not set.

FIELD MARK Key

Operation of this key causes a X'1E' code to be entered into the active partition buffer and the MDT bit to be set to 1. The field mark character informs the application program of the end of a field in an unformatted buffer or subfield in a formatted buffer. The field mark character transferred to the application program is X'1E' (EBCDIC) and is sent when the data is read. The field mark character, when stored in the character buffer, is displayed as semicolon-overscore. Display devices operating in monocase mode may also display the FIELD MARK character as a semicolon.

Operation of this key when the cursor is positioned at a field attribute location or is within a protected field disables the keyboard; no character locations are cleared, the cursor is not moved, and the MDT bit is not set.

Program Attention Keys

These keys solicit application program action by causing the sending of an attention identification (AID) character to identify the key. The MDT bit is not affected. The program attention keys are CLEAR, ENTER, CANCEL, all program function (PF) keys, and the program access (PA) keys.

Note: Not all program attention keys are available on all types of keyboards.

Clear Partition

The effect of invoking this function depends on whether the display is in implicit partition state or in explicit partition state.

The operation of the Clear Partition function is:

1. Sets all buffer locations for the active partition to nulls, and all character attributes to X'00'.
2. Sets the reply mode for the active partition to the default (field reply mode).
3. Causes an AID (of value X'6A') to be transmitted inbound in a Short Read operation.
4. Moves the current cursor position to the top left corner of the presentation space (buffer address 0). In a scrollable partition, this may cause automatic scrolling to occur.

Invoking this function while a previous inbound transmission is being processed will lock the input device, and signal an input-inhibited condition to the operator on the indicator row of the display.

Clear

If the display is in implicit partition state, invoking the Clear function will result in the following:

1. All buffer locations will be set to nulls.
2. All character attributes will be set to their default values.
3. The MDT bit will be set to 0.
4. The reply mode will be set to the default (field reply mode).

5. An AID of X'60' will be transmitted inbound in a Short Read operation.
6. The implicit partition will be set to its default size.
7. The cursor is moved to the current cursor position, buffer address 0.

If the device is in explicit partitioned state, invoking the Clear function will result in the following:

1. All existing partitions are destroyed.
2. A partition with a PID of 0 and default size is created.
3. Resets the device to Implicit Partition State.
4. The cursor is moved to the current position, buffer address 0.
5. All buffer locations will be set to nulls.
6. All character attributes will be set to their default values.
7. The MDT bit will be set to 0.
8. The reply mode will be set to the default (field reply mode).
9. An AID of X'6D' will be transmitted inbound in a Short Read operation.

Keyboard Actions with Attribute Selection Keys

For those keyboards that do not have attribute selection keys for character sets, color, and highlighting, pressing an alphanumeric key causes the MDT bit to be set to 1 and the alphanumeric character to be entered into the display's character buffer at the position indicated by the cursor. The keyed character is assigned default values for programmed symbols, color, and highlighting.

For keyboards with keys that allow the operator to select or change the character attribute without application program interaction, the default settings are obtained as stated in the previous paragraph. When enabled by the application program keying sequences are defined that allow character sets, color, and highlighting attribute values to be selected by the operator.

Once these attribute values have been selected, pressing an alphanumeric key causes the alphanumeric character to be entered into the character buffer at the cursor position and the MDT bit to be set to 1. In addition, the character attribute associated with the entered character is modified by the attribute values selected by the operator (if operator alteration of the character attribute have previously been allowed by the Set Reply Mode structured field). The operator's selections apply until changed, or until the keyboard is returned to the default setting.

Keyboard Actions in Partitions

As the operator enters data, the current cursor position is incremented by 1 for each character entered, and the cursor is displayed in the new character location. This cursor movement may cause automatic scrolling.

The operator may also move the screen cursor by using the Up, Down, Left, Right, Backspace, Tab, Backtab, Skip, or New Line keys. The Up, Down, Left, Right, and Backspace keys move the screen cursor within the viewport of the active partition and cause the current cursor position to be updated. Thus, the screen cursor wraps at the viewport boundaries.

The entering of data and the Tab, Backtab, Skip, and New Line keys move the current cursor position within the character buffer of the active partition. Where the buffer is larger than the viewport, automatic scrolling may occur.

If a character buffer contains at least one unprotected field, the first unprotected character location in the buffer is termed the *home position*. If the buffer contains no unprotected fields, the home position is defined to be character location zero.

The Home key resets the current cursor position in the active partition to the home position and causes the cursor to move to this position within the viewport. This may cause automatic scrolling to occur.

The effect of the Delete, ERASE EOF, and ERASE INPUT keys is constrained to the active partition; that is, no deletions or storing of nulls occurs outside the active partition. The Clear Partition key clears the active partition. The CLEAR key clears the entire screen by destroying all partitions, placing the device in implicit partition state, and re-creating an implicit partition 0 of default size.

Scrolling Partitions

Different areas of the presentation space can be seen in the viewport by a technique called *scrolling*. Scrolling within a partition is possible if the presentation space is larger than the associated viewport. Scrolling may be done by the operator using the keyboard keys or by the host using a structured field.

When a partition is created, it can be set up as a scrollable partition by making the Presentation Space larger than the viewport. The viewport seen by the operator has a one-to-one relationship with a window on the corresponding presentation space. The position of the window on the presentation space determines what data is seen in the viewport. The initial position of the window on the presentation space is specified by the Create Partition structured field. The position of the window on the presentation space is reset using the Set Window Origin structured field. The number of rows moved during scrolling is determined by the RS Value given in the Create Partition structured field.

Row-by-row vertical scrolling within the presentation space is provided in response to the operator's pressing the keys for the scroll up and scroll down. Vertical scrolling is achieved by moving the position of the window within the presentation space in response to the use of the keys for scrolling.

The operator interacts with the system through information displayed in the viewport. The cursor is always displayed within the viewport. Whenever an operator keystroke (of data or the field-oriented keys) causes the cursor to leave the viewport, there is an automatic scroll. Conversely, when a scrolling operation moves the window so that the cursor would no longer be within the viewport, the cursor is dragged along at the edge of the window and hence remains within the viewport.

The cursor movement keys (Up, Down, Left, Right, Backspace) wrap at the boundary of the viewport. The actions for data keystroking and the field-oriented keys are not affected by scrolling, that is, they wrap at the boundary of the character buffer. The wrap at the buffer boundary may cause an automatic scroll.

Vertical Scrolling

Assuming that the window is not already positioned at the bottom of the presentation space, the action in response to scroll up is to move the window *down* the presentation space. The effect, as seen by the operator, is to move the data *up* the viewport.

The previous top row is lost from the window; all other rows are moved up.

If the current cursor position is in the top row of the window, the cursor is moved down the presentation space so that it remains in the top row of the window. The effect seen by the operator is that the cursor remains on the edge of the viewport.

If the window is already positioned at the bottom of the presentation space, then scrolling up has no effect. Note that there is no input-inhibit condition and no indicator.

Assuming that the window is not already positioned at the top of the presentation space, the action in response to scroll down is to move the window *up* the presentation space. The effect, as seen by the operator, is to move the data *down* the viewport.

The previous bottom row is lost from the window; all other rows are moved down.

If the current cursor position is in the bottom row of the window, the cursor is moved up the presentation space so that it remains in the bottom row of the window. The effect seen by the operator is that the cursor remains on the edge of the viewport.

If the window is already positioned at the top of the presentation space, then scrolling down has no effect. Note that there is no input-inhibit condition and no indicator.

Keyboard Actions and Scrolling

The character-oriented keys (Up, Down, Left, Right, and Backspace) wrap the cursor at the viewport boundary. The field-oriented keys (Tab, Backtab, Skip, and New Line) and automatic skip operate on the entire presentation space and wrap at its boundary. If the resulting current cursor position is outside the window, there is an automatic scroll.

The Home key positions the cursor at the home position within the presentation space. This key may cause an automatic scroll.

Action for Data Entry Keystrokes

All data entry keystrokes cause normal incrementing of the cursor position, including a wrap at the boundary of the presentation space. If the resulting cursor movement places the cursor outside the window, there is an automatic scroll.

The ERASE EOF, ERASE INPUT, Clear Partition, CLEAR, and Delete keys, and the use of the Insert Mode key, can cause changes to the character positions anywhere in the presentation space; in particular, such changes can be outside the viewport.

Automatic Scroll

An automatic scroll occurs whenever a field-oriented key, a data entry keystroke, the Home key, or an IC order results in a current cursor position that is outside the window.

The window is moved by the minimum number of lines necessary to put the current cursor position in a peripheral row of the window. The effect is that the cursor always remains within the window, and hence within the viewport.

Selector Pen Operation

The selector pen is a light-sensitive pen-like device that can detect the light emitted from characters displayed on the display surface. With the selector pen, the operator can select from a list or table of displayed items and can then cause those selections to be identified to the application program.

Selector Pen Field Format

A field that is to be used for selector pen operations must be defined in the following format:

Field Attribute	The field attribute defines the field as displayed and selector pen detectable. (A field may be protected or unprotected, alphanumeric or numeric.)
Designator Character	The designator character defines the type of operation that will be performed by detection on this field.
Displayed Data	One or more displayed alphanumeric characters for sensing by the selector pen.

Designator Characters

Designator characters are used to define two types of selector pen fields: selection fields and attention fields. Each type of field performs a different selector-pen operation. If the field attribute defines a field as selector-pen-detectable but the character in the designator character position is not a valid designator, a detect cannot be made on the field.

Selection

The selection field is defined by a question mark (?) designator character (X'6F'). When the selector pen detects on a selection field, the MDT bit in the field attribute for the field is set to 1 and the X'6F' is changed to X'6E'. Also, the designator character is automatically changed in the character buffer and on the display surface to show to the operator that the detection was successful. In summary:

If Before Detect:		**Then After Detect:**	
Designator	*MDT*	*Designator*	*MDT*
?	0	>	1

If Before Detect:		**Then After Detect:**	
Designator	*MDT*	*Designator*	*MDT*
?	1	>	1

The field may be deselected by using the same technique as for selection.

If the designator is > (X'6E') and a detect is made on the field, the designator is changed to ? (X'6F') and the MDT bit is set to 0 and the X'6E' is changed back to X'6F'. In summary:

If Before Detect:		**Then After Detect:**	
Designator	*MDT*	*Designator*	*MDT*
>	0	?	0
>	1	?	0

Attention

The attention field is normally defined by a space or null designator character. In addition, an implementation may support the ampersand (&) designator character to provide an ENTER key simulation.

A detection on an attention field designated by a space or null causes an AID X'7E' to be sent to the application program that identifies the source of entry to be the selector pen. Also sent are the addresses of fields with the MDT bit set to 1 and the current cursor position for the partition that contains the selected field. The application program responds, usually by issuing a Read Modified All, to obtain the contents of modified fields and the addresses.

A detection on an attention field designated by an ampersand results in an AID X'7D' being sent that identifies the source of entry to be the ENTER key. Also sent are the addresses and contents of all fields with the MDT set to 1 and the current cursor position for the partition that contains the selected field. If the application defines an attention field with an ampersand designator and the implementation does not support the ampersand, the selector pen is inoperative on the field; the operator must then use the ENTER key.

Note: The application programmer should be aware that high-intensity/unprotected fields can be modified by the operator to become selector-pen-detectable fields.

Selecting Fields in Partitions

The operator may use the selector pen to select fields in any partition, whether it is active or inactive. Selection of a field in an inactive partition causes no movement of the cursor on the display surface and the partition containing the selected field does not become active, but the operator may notice a change in the indicators. The operation performed depends on the designator character selected.

A selection outside the viewports is ignored.

Selection of an immediate detect field (designator of blank, null, or ampersand) causes data to be transmitted. This data includes the addresses of all the modified fields and the current cursor position for the partition that contains the selected field. Note that this need not be the active partition.

Chapter 8. Printer Considerations

Printers

Although oriented toward a display with a keyboard, the 3270 data stream descriptions apply, with certain exceptions, to a printer without a keyboard. The exceptions are the following:

- Commands. The read commands are not valid in the 3270 data stream received by the printer in an SNA environment. If received, the request is rejected (sense code X'1003'). See also the *Structured Field* exception. However, the read commands are valid if received in the 3270 data stream in a non-SNA environment for locally attached 3274D models (or 3272 version) or BSC attached devices.
- Structured Fields. Structured fields that are not supported and structured fields with invalid values are rejected.
  - The Create Partition, Activate Partition, and Set Window Origin structured fields are not supported.
  - The Read Partition structured field is limited to query operations in an SNA environment. The Read Modified command is supported only for sending query replies to an application.
- Selector pen. Not applicable to printer.
- WCC byte. The WCC byte definition for printer use is shown in Figure 8-1.
- Format-control orders. Unlike displays, the NL, CR, EM, and FF orders provide a print format function when received by a printer, as follows:
  - NL (New Line). Moves the print position horizontally to the left margin and vertically down to the next line.
  - CR (Carriage Return). Moves the print position horizontally to the left margin.

Table 8-1 (Page 1 of 2). The WCC Byte (As Defined for Use with Printers)

Bit	Explanation
	In an *SNA/EBCDIC* environment, the bits are ignored by the printer. In an *SNA/ASCII* environment, the bits are set in accordance with Figure D-1 in Appendix D to make the WCC byte an EBCDIC/ASCII-translatable graphic.
2&3	Defines printout format as follows: 00 - The NL or CR orders in data stream determine print line length, and EM designates the end of the message. Provides a 132-character print line when orders are not present. 01 - Specifies 40-character print line 10 - Specifies 64-character print line 11 - Specifies 80-character print line

Table 8-1 (Page 2 of 2). The WCC Byte (As Defined for Use with Printers)	
Bit	**Explanation**
4	Start-printer bit. When set to 1, initiates a printout operation at the completion of the write operation.
5	Sound-alarm bit. When set to 1, sounds the audible alarm if an audible alarm is provided.
6	Keyboard-restore bit.
7	Reset MDT bit.

- EM (End of Message). Terminates the print operation.
- FF (Form Feed). Moves the print position to the top and left margin of the next page.

In normal operation, when bits 2 and 3 of the WCC are 0, a new-line function is performed each time a valid NL character is encountered. In addition, if no valid NL is encountered before the printer reaches the end of a line (as determined by the maximum physical carriage length), the printer automatically performs NL and continues printing.

During a print operation, if line-length format is specified in bits 2 and 3 of the WCC (bits 2 and 3 not equal to 0), data characters in the printer buffer are scanned one line at a time before they are printed. A line feed is executed after each line is printed. If a line contains only null characters and one or more space characters, a line feed is performed to cause a blank line in the printout. When null characters, field attributes, or alphanumeric characters in a nonprint field are encountered, they are treated as follows:

- If in a line that contains another print field, they are printed as spaces.
- If they constitute an entire line, they are ignored, and the line feed is not performed; as a result, a blank line does not appear in the printout, and the data is compressed vertically one line.

When line-length format is not specified, printout of the buffer begins at buffer address 0 and continues until the last position of the character buffer is printed or until a valid EM character is encountered. Each print line is left-justified. At the end of each printout, a final New Line is executed so that the printer is ready to start the next printout. When the print-terminating EM order appears in the first print position of the print line, a final New Line is not executed, because the printer is already positioned at the left margin for the next printout.

The validity requirements for NL, CR, EM, and FF are as follows:

- NL, CR, and EM are valid only when encountered in a print field during a printout that does not have a line-length format specified by the WCC.
- FF is valid in any buffer position.

When the printer supports vertical forms control (VFC), an FF causes the form to index to a predetermined line, and the first print position (the buffer location containing the FF character) is printed as a space character. If VFC is not supported, the FF is invalid.

The rules for NL apply to CR. However, the printer must support CR at least to the extent of accepting and printing it as a space. Invalid NL, CR, EM, and FF are not executed and print as spaces.

- Data Integrity. Since printers in an SNA environment do not support a read operation, the data integrity requirement on returned data is not applicable. Where the requirement says *must be returned as sent*, for printers it becomes *must be accepted.*

Because printers in a non-SNA environment can be read by the application program, these printers are subject to the same data integrity requirements as displays.

Local-Copy Function in an SNA Environment

A hard copy of the display screen or a portion of a display screen may be accomplished by "host" application use of the conventional display read function and the printer write function. However, to simplify the host application and reduce line traffic, a specific copy function is provided that accomplishes a transfer of data to a printer without routing through the host application.

Whether initiated by the host or by the operator, the resultant hard copy is a replica of the screen or portion of the screen, with the following exceptions:

- Display lines containing all nulls are suppressed.
- Mismatch of uppercase and lowercase may occur. When the configured printer(s) has the capability for selection of Mono-Dual case without requiring operator action, the printed data must match the case of the displayed data. It is not necessary for the printer to recognize any manual override by the display of uppercase and lowercase.
- Color may not match.

The local-copy operation is described on the basis of a nonpartitioned screen, that is, for a display that does not have partitioning capability or for a display with partitioning capability that is in implicit partition state (implicit partition 0). Differences due to partitioning are described under "partition Mode Considerations."

Copy Initiation

The host initiates a local copy by sending the display a write-type command with the start-print (SP) bit set to 1 in the WCC byte. If the write-type command includes data, the screen will be updated prior to print execution. Once a copy request is accepted, the display keyboard remains locked until the printer completes the print operation. The host is required to send the Copy request either as an RQD chain or as an RQE, CD, not-EB chain; this prevents the host from following a copy request immediately with another command.

Printer Availability

When more than one printer is authorized for a display's use, a Copy request could result in the printout on any of the authorized printers. From the host view, however, there is only one logical printer, regardless of how many physical printers are authorized for the display's use; the host application has no control over the printer selected. The user must predefine (for example, IML) the printer configuration. A printer must be capable of being configured for system use only (no copy use), for copy use only, or for shared use between the system and copy.

A logical printer is considered available if one or more of the physical printers, configured for the display's use, can immediately execute a display printout. When none of the configured printers is available, the logical printer assumes the status of the *most available* physical printer. The following are the not-available categories listed in order of best to worst:

1. Short-term busy. A display printout can be executed after n queued display printout requests are executed. The maximum delay is 30 minutes. Use by a session is not allowed.
2. Intervention-required condition.
3. Allocation for session use or local use (not copy).
4. Permanent error condition (or an intervention-required situation on an unattended printer).
5. No logical printer configured.

The display will reject a copy request whenever the logical printer is not available. The following are the sense codes used:

- 082E. The logical printer has an intervention-required condition, for example, out of paper, power off, and cover interlock open. This code is used if the printer is attended. In general, *attended* means the condition will likely receive quick attention; for example, the printer is located near a display. If the printer is unattended, the permanently unavailable code should be used.
- 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 display for a display printout to many hours if all printers are in session. The display should mask *short term busy* from the host by withholding the -rsp (0807).
- 082F. Effectively, the logical printer is permanently not available; for example, a hardware failure.
- 0801. No printer configured.

After sending an 082E or 0807, an LUSTAT must be 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 0001 (with source=printer) indicates to the host that a Copy request will find either an available or short-term-busy logical printer. Exceptions are as follows:

- The LUSTAT 0001 (source=printer) reporting the clear of a 082E or 0807 condition is not sent if either of the following is true:
  - After sending 082E/0807 -rsp, the Data Traffic Subtree is reset or the session is terminated.

- After sending 082E/0807 -rsp, the logical printer develops a permanent error or becomes not configured.

  An LUSTAT 081C/0801 (source = printer) is sent instead of LUSTAT 0001.

  The LUSTAT 0001 (source = printer) does not indicate the host will necessarily find an available or short-term-busy logical printer if one of the following is true:

- After sending 082E/0807 -rsp, the SLU receives any normal FM data request chain other than a Copy request chain.
- After sending LUSTAT 0001 (source = printer), the logical printer develops an intervention-required, permanent-error, or not-configured condition. The Copy request will be rejected with the appropriate -rsp, that is, 082E, 082F, or 0801.
- It is not valid for an implementation to allow session contention for a printer prior to honoring outstanding LUSTATs; that is, at least one physical printer must be held after having sent the LUSTAT(s) so that the logical printer is available to execute copy sessions. However, if as a result of an exception condition developing on the held printer the logical printer status of *in session* results, then 0807 -rsp may also be sent.
- After sending 082E/0807 -rsp, the display detects a nonprinter exception condition which will cause a -rsp to a received request.

The sense codes 082E, 0807, 082F, and 0801 indicate to the host that, if the copy request included screen update data, the screen update was accomplished. If a copy request (with update data) is re-sent, it cannot be guaranteed that the screen will be unchanged. An example where the screen would be changed is if the data used a positioning reference, for example, a Program Tab order that was revised later in the data stream.

Display/Printer Compatibility

In general, a print operation will not be inhibited because of a mismatch in display/printer capability. However, a copy operation may be inhibited if the printer is unable to accept all the display data, for example, print buffer too small, printer maximum line length less than display width. When the copy is inhibited because the printer cannot accept all the display data, the printer is considered to be *not authorized*.

APL Mismatch

Where the display has the APL feature, but the printer to which the copy is directed does not have APL, the copy operation will occur using the nonloadable character set. There will be no indication of this mismatch. Such mismatches can be avoided by the proper configuration of copy printers.

Character Attribute/Extended Field Attribute (CA/EFA) Mismatch

On a copy operation the CA/EFA must be sent to the printer if the printer supports CA/EFA *and* if the display data to be copied references nonzero values of CA/EFA. When not all the extended functions are referenced in the copy data, an implementation may send only the required information or may send all the CA/EFA information. For example, if only extended color were referenced in the display data, that is, no character set, or extended highlighting selected, an implementation may send only the color information from the CA/EFA or all the CA/EFA information.

Character Attribute/Extended Field Attribute (CA/EFA) Mismatch

On a copy operation the CA/EFA must be sent to the printer if the printer supports CA/EFA *and* if the display data to be copied references nonzero values of CA/EFA. When not all the extended functions are referenced in the copy data, an implementation may send only the required information or may send all the CA/EFA information. For example, if only extended color were referenced in the display data, that is, no character set, or extended highlighting selected, an implementation may send only the color information from the CA/EFA or all the CA/EFA information.

Programmed Symbols (PS) Considerations

A copy operation will not be inhibited because of a mismatch of display/printer character-set capability or character-set load. Whenever the printer cannot print the data against the same loadable character set referenced by the display data, the print is executed against the nonloadable character set of the printer.

The Load PS structured field contains a compare field of 1 bit. If set to compare (bit=0), the LCID may be used in establishing a PS match in a copy operation; for a valid match, the matching LCIDs must both be set to compare. If set to no-compare (bit=1), the LCID cannot be used for establishing a match.

The following summarizes the copy printout operation:

IF The referenced display PS is set to compare, and the printer has a matching LCID also set to compare, then execute printout against the matched LCIDs.

ELSE Execute the printout against the printer default character set.

Note that the printed data could have a mixed appearance, that is, some matching the displayed data and some not matching the displayed data.

Extended Color Mismatch

A copy operation will not be inhibited because of a mismatch in color capability between the display and the printer; that is, no check is made for display/printer compatibility. If the printer does not support color, display data in color is printed in monochrome. If the printer supports color but a different set of colors than the display, the colors not supported are printed in the default color defined by the printer Query Reply color.

Extended Highlighting Mismatch

Copy will not be inhibited because of a mismatch in extended highlighting capability between the display and the printer; that is, no check is made for display/printer extended highlighting capability. If the display specifies a particular form of highlighting not supported by the printer, the data is printed with defaults defined in the printer's Query Reply highlight.

Partition Mode Considerations

When the display is in partition mode with or without scrolling, the copy operation changes as follows:

> An Outbound 3270DS structured field with a WCC=SP initiates a printout of the window of the designated partition. If a Write Structured Field command contains multiple structured fields, the WCC=SP may be set only in the last

structured field. If set in any other structured field, the Write Structured Field is rejected with sense code X'1001'. The printout of the partition viewport has the same width as the display partition and is left-justified. If the display partition width exceeds the printer line length, the copy operation is rejected with sense code X'0801', no printer configured.

If a Write, EW (reset off), or an EWA (reset off) is sent with WCC=SP to a partitioned screen, a printout of partition 0 will occur. If partition 0 does not exist, the copy operation is rejected with sense code X'1005', parameter error. If an EW (reset on) or an EWA (reset on) is sent with WCC=SP to a partitioned screen, the screen is reset to implicit partition 0.

Local-Copy Command in the BSC Environment

In BSC, the local-copy function is accomplished by use of the Copy command. This Copy command is used to transfer buffer data from one terminal to another terminal attached to the same controlling device. The selected terminal is the *to* terminal, the one to which the buffer data will be transferred. The *from* terminal, the source of the buffer data to be copied, is identified in the second two bytes that follow the Copy command code; the first byte, called the *copy* control character (*CCC*), identifies the type of data to be copied. The CCC can also, at the *to* device, start print operations, specify the printout format for those operations, and, when the terminal is a display station, sound the audible alarm.

The copy data stream is shown in Figure 8-1 on page 8-8.

Table 8-2 on page 8-9 describes the function of each CCC bit. A CCC and an address byte must always follow the command code. If they do not, the controlling device aborts the command and generates error status.

The *from* terminal buffer can be locked (made incapable of being copied) by writing a protected/alphanumeric attribute byte (bit 2=1 and bit 3=0) in address 0.

The Copy command can specify as the *from* device the same device that is selected (the *to* device). This procedure provides a means of programming selective device buffer "erase" operations as specified by CCC bits 6 and 7.

Programming Notes:

1. Copy should not be chained from a Write, Erase/Write, Erase/Write Alternate, or Erase All Unprotected command, since it will copy the data as modified by the Write or Erase command.
2. If the CCC start-print bit is set and commands are being chained, Copy should be the last command of the chain. If not, the control unit aborts the subsequent command.
3. Copy can be executed from a smaller buffer to a larger buffer, but an attempt to copy from a larger to a smaller buffer will cause an operation check.

If the Copy command references a *from* terminal, in implicit partition state, that has not received an SFE, SA, MF, or GE, or on which the operator has not entered a character with an extended attribute, or a specific APL/TN character, since the last buffer clear (for example, EW, EWA commands), a copy action takes place. If the *from* terminal does not satisfy the above criteria, the Copy command will be rejected with an operation check (OC) and unit specify (US) status, unless all the following conditions are met:

1. The *from* terminal is a display.
2. The destination terminal is a printer.
3. The *from* terminal does not have a protected/alphanumeric field attribute in the first buffer position.
4. The CCC has bits 4 (start print), and 6 and 7 (copy entire buffer) set to 1.

If these conditions are met, an attempt will be made to produce a local copy. Following print completion, the print buffer will be cleared and the appropriate completion status made available at the printer.

Byte		
0	STX	
1	ESC	
2	Copy Command Code	
3	CCC	← See below
4	*From* Device Address	
5	ETX	

CCC-Byte Format

*	1	Printout Format	Start Print	Sound Alarm	Type of Data to Be Copied
0	1	2 3	4	5	6 7

*Determined by the configuration of bits 2 through 7.

Figure 8-1. The Copy Data Stream

Table 8-2. Copy Control Character (CCC)	
Bit	**Explanation**
0,1	Defined to make the ccc a translatable character. (See Figure D-1 on page D-2.)
2,3	Define the printout format as follows: 00 - The NL, EM, and CR orders in the data stream determine point-line length. Provides a 132-print position line when the orders are not present. 01 - Specifies a 40-character print line. 10 - Specifies a 64-character print line. 11 - Specifies an 80-character print line.
4	The start-print bit. When set to 1, initiates a printout operation at the *to* device after buffer transfers are completed.
5	The sound-alarm bit. When set to 1, sounds the audible alarm at the *to* device after buffer transfers are completed if that device has an audible alarm.
6,7	Define the type of data to be copied as follows: 00 - Only attribute characters are copied. 01 - Attribute characters and unprotected alphanumeric fields (including nulls) are copied. Nulls are transferred for the alphanumeric characters not copied from the protected fields. 10 - All attribute characters and protected alphanumeric fields (including nulls) are copied. Nulls are transferred for the alphanumeric characters not copied from the unprotected fields. 11 - The entire contents of the storage buffer (including nulls) are copied.

IPDS Data/Non-SNA

This section addresses the "carrying" of the Intelligent Printer Data Stream (IPDS) by the 3270 data stream in a non-SNA environment. For the SNA environment, IPDS data stream is available through LU1.

When the 3270 data stream/IPDS is supported, the printer will be in either a 3270 data stream mode or in IPDS mode. The 3270 type data and controls and IPDS type data and controls can not be "mixed." In other words, you cannot send 3270-type data and IPDS-type data and then print the composite.

The 3270IPDS Query Reply will indicate the support of IPDS and will also define the maximum transmission size allowed outbound in the IPDS mode. However, the mechanisms provided by the IPDS (for example, STM/ACK REPLY) will be used to provide IPDS-related printer characteristics. Also, when in the IPDS mode, the exception handling functions provided by the IPDS will be used.

A change of mode will clear the printer buffers of any data associated with the previous mode. Use of the printer for 3270 local copy is considered to be 3270 mode. For example, if the printer buffer was loaded in the 3270 mode, and then the printer changed to the IPDS mode, the 3270 data would be cleared. Therefore, a return to the 3270 mode would find a cleared buffer. However, the printer must maintain forms sync across modes as is done between 3270 and SCS. Refer to the 3270IPDS Query Reply for non-SNA systems and the Data Stream Query Reply for SNA systems for more information about IPDS selection.

IPDS Selection

The default (for example, power on) mode will be the 3270 mode. When in the 3270 mode, the structured fields defined in the IPDS will be rejected. When a device is in the IPDS mode, any non-IPDS structured field (except a valid Data Chaining or Select IPDS structured field) will be invalid (refer to the IPDS manual for action taken on invalid structured fields). Also when in the IPDS mode, 3270 orders and control sequences will be treated as IPDS data with unpredictable results.

Support of IPDS requires that data chaining (for example, the Data Chain structured field and the Data Chaining Query Reply) be supported. See "Structured Field Grouping" on page 5-3.

If the Data Chain structured field is not used in the selection sequence (that is, no data chaining), the IPDS mode will exist until the end of the transmission. If the Data Chain structured field is used (that is, data chaining), the IPDS mode will exist until the end of the data chain.

The IPDS mode is selected by the Select IPDS Mode structured field. The Select IPDS Mode structured field will also reset IPDS controls and conditions to the default values. On outbound (to the device) the Select IPDS Mode structured field must immediately follow a WSF, or immediately follow a WSF Data Chain structured field (GROUP = begin) sequence. On inbound (from the device) the Select IPDS Mode must immediately follow an AID X'88' or immediately follow an AID X'88', that is, a Data Chain structured field (GROUP + begin) sequence.

When using data chaining, the Select IPDS Mode structured field only appears immediately after the Data Chain structured field of the first transmission of the data chain. That is, the IPDS mode is continued by the subsequent Data Chain structured fields (GROUP = continue) and terminates at the completion of the last Data Chain structured field (GROUP = end). For example, IPDS structured fields and control sequences may span transmissions. If the printer is not in the IPDS mode, a Select IPDS Mode structured field appearing anywhere except after a WSF command, or after a WSF Data Chain structured field (GROUP = begin) sequence will be rejected. If the printer is in the IPDS mode, a Select IPDS Mode structured field received anywhere except after a Data Chain structured field (GROUP = begin) sequence will be treated as an invalid structured field and handled in accordance with the IPDS.

With the exception of the IPDS ACK structured field which reports exception and status conditions, IPDS is an output-only data stream. Sending of the ACK structured field by the device is limited to either of the following:

- In reply to an IPDS outbound structured field that requests a reply
- Reporting exception conditions that were detected while processing an outbound IPDS transmission.

Local Copy

When in the IPDS mode, the printer cannot be used for operator- initiated local copy. If it is in IPDS mode and the BSC Copy command is received, the IPDS mode will be terminated without error and the copy executed.

Pacing

The non-SNA protocols do not provide the chaining and pacing functions provided by SNA. Control of data to the printer in the IPDS mode is accomplished by a combination of data chaining, limiting the transmission size, and link and channel controls.

The 3270IPDS query reply indicates the maximum allowed transmission length when in the IPDS mode. The data chaining function allows a message to be divided into transmissions of suitable length without regard to structured field or control boundaries, for example.

The IPDS mode uses the same BSC and channel controls as does the 3270 mode to determine when more data may be sent to the printer. In BSC, a wait before transmit positive acknowledgment (WACK) is returned after acceptance of a transmission if the printer cannot accept more data at this time. When the printer is ready for another transmission, a device end (DE) is returned by the device. In a non-SNA local channel, if the printer cannot accept more data at this time, channel end (CE without DE) is returned after acceptance of a transmission. When the printer is ready for another transmission, a DE is returned to the device.

Input Transmissions

Although the IPDS data stream is essentially an output only data stream, the printer will generate some inbound data in the form of an acknowledgment reply. This reply can be generated asynchronously (exception reporting) or synchronously (in reply to a "host" request).

Chapter 9. Binary Synchronous Communications (BSC) Environment

The 3270 data stream operates the same in a binary synchronous communication (BSC) environment as it does in the SNA environment, except for the differences noted in this chapter.

Transparent Mode

The SNA environment provides inherent transparency on the data line. This mode of operation permits greater versatility in the range of coded data that can be transmitted. This is because all data, including the normally restricted data link line control characters, is treated only as bit patterns when transmitted in transparent mode-operation. This is required when transmitting binary counts and addresses that may appear in the data stream for the functions of extended highlighting, color, and programmed symbols.

BSC data links may be either nontransparent or transparent, however, when the SF, SFE, and MF orders and the WSF command are supported; the BSC link must be in the transparent mode. The format of the data stream for BSC is:

- For nontransparent mode:

STX	3270 Data Stream	ETX

- For transparent mode:

DLE	STX	3270 Data Stream	DLE	ETX

Write Commands

The Erase/Write and Erase/Write Alternate commands operate the same in a BSC environment as in the SNA environment. The Write command operates the same in a BSC environment as in the SNA environment except that the starting buffer location depends upon the following considerations:

- The starting location may be specified by a Set Buffer Address order that follows the WCC.
- The starting location will be the buffer address containing the cursor if the Write command is not chained from a Copy or Erase All Unprotected.
- The starting location will be the current buffer address if the Write command is chained from a Read or another Write command.

Read Commands

The read commands operate the same in a BSC environment as in the SNA environment except for the differences that follow.

Read Buffer Command

Execution of the Read Buffer command causes all the data in the addressed device's buffer, from the buffer location at which reading starts through the last buffer location, to be transferred to the application program's storage. The transfer of data begins:

1. From buffer address 0 if the Read Buffer command is unchained.
2. From the current buffer address if the Read Buffer command is chained from either a Write, Erase/Write, Erase/Write Alternate, Read Modified, or another Read Buffer command. Regardless of where the transfer of data begins, data transfer from the buffer will terminate when the last character location in the buffer has been transferred, or when the last character of a text block has been transferred.

Read Modified Command

Read Modified initiates one of three operations as determined by operator actions at the display: (1) read modified, (2) short read, or (3) test request read.

Read Modified functions the same as in an SNA environment except that the buffer location at which the search begins for field attributes that define modified fields is a function of command chaining. This location is:

1. Buffer address 0 if the Read Modified command is unchained or is chained from a Copy.
2. The current buffer address if the Read Modified command is chained from a Write, Erase/Write, Read Modified, Read Modified All, or a Read Buffer command.

   The search for modified field attributes ends when the last buffer location is checked.

The transfer of read data is terminated as follows:

1. If the last modified field is wrapped from the last buffer location (for example, 479 or 1919) to the first location, the operation is terminated after all the data in the field is transferred (nulls are suppressed). The buffer address at the end of the operation is the address of the next field attribute byte in the buffer. For example, if a modified field extends from address 1900 (the field attribute) to address 79 (wrapped field), the data from address 1901 through 79 is transferred (nulls suppressed); in this case, the read operation is terminated with the buffer address set to 80 (the field attribute byte of the next field).
2. If the buffer does not contain a wrapped modified field, the modified data stream is terminated when the last modified field is transferred. At the end of the operation, the buffer address is set to 0.

   If the buffer is formatted but none of the fields have been modified, the read data stream consists of the 3-byte read heading. (AID plus cursor address.)

   If the buffer is unformatted, the read data stream consists of the 3-byte read heading followed by all the alphanumeric data in the buffer (nulls suppressed), even when part or all of the data has not been modified. Since an unformatted buffer contains no field attribute bytes, no SBA codes with associated addresses or address characters are included in the data stream, and the modification of data cannot be determined. Data transfer starts at address 0, regardless of command chaining, and continues to the end of the buffer. At the end of the operation, the buffer address is set to 0. This read operation can also be terminated by the channel byte count's reaching 0 before all data is read. In this case, the buffer address after termination is undefined.

Test Request Read

The Read Modified command causes a test-request-read operation if the TEST REQ or SYS REQ key has been pressed at the selected device. The Test request Read data stream sent inbound to the application program is as follows:

SOH
%
/
STX
Input Data
ETX

The Test Request Read heading is generated by the display. The remainder of the data stream is the same as described previously for read-modified operations, excluding the 3-byte read heading (AID and cursor address). If the buffer is unformatted, all the alphanumeric data in the buffer is included in the data stream (nulls suppressed), starting at address 0. If the buffer is formatted, each field-attribute byte is examined for a set MDT bit. Each time a set MDT bit is found, the alphanumeric data in the field associated with that bit is sent to main storage (nulls suppressed). If no MDT bits are set, the read data stream consists of the Test Request Read heading only. The buffer location at which the search for the MDT bits begins and the transfer of data ends is the same as described for the read-modified operations.

Note: Usage of the test-request-read function is determined by the application program. Normally, the operator would (1) clear the display, (2) enter test-request data in a predefined format, and then (3) press the TEST REQ or SYS REQ key.

Inbound Transmissions

Inbound transmissions result from an operator *enter* action, an application-program-initiated (unsolicited) read request, or a host retry of an inbound transmission.

An operator enter action is one that causes an attention identifier to be transmitted inbound. The application program responds with a read request. The application program must acknowledge the inbound transmission before a new inbound operation can be performed.

An application-program-initiated read operation is an inbound transmission not caused by an operator enter action. No application program acknowledgment is required before a new inbound transmission can occur.

Host retry is a retransmission of the last unacknowledged inbound transmission from the device. The application program must acknowledge receipt of an inbound transmission before a new inbound transmission can take place. A host retry transmission does not cause read-state transitions and is not considered a new inbound transmission requiring acknowledgment by the application program. Host retry occurs until an application-program acknowledgment takes place.

The type of inbound transmission is either a Query Reply structured field (the response to the Read Partition Query structured field) or data from the device buffer (for example, modified fields of the display image). An inbound operation device characteristic (INOP), set by the control unit, defines the type.

Inbound Operation (INOP)

The INOP determines the operation to be performed when data is transmitted inbound on a retry transmission.

INOP is set by any of the following:

- An operator enter action sets INOP to Read Modified.
- Reception of a Read Partition Query structured field sets INOP to Query.
- Application-program acknowledgment of an inbound transmission sets INOP to Read Modified.

Read States

While powered on, a device is in one of seven states with respect to read operations. The three primary states are:

- Normal read
- Data pending read
- Read retry.

The data-pending and read-retry states have three substates:

- Enter
- Read
- Stacked enter.

The events that cause transitions between states are shown in Table 9-1.

Normal Read State

A device is in normal-read state when powered on, or prior to initiation of a new read operation, or after use of the RESET key in certain instances. (See "Read-State Transitions" on page 9-5.)

When in normal-read state an operator enter action or the reception of a Read Partition Query structured field causes the device to prepare to generate the inbound data stream and to go into a data-pending state.

When in normal-read state, an application-program-initiated read operation using a RM, RMA, or RB command causes the data to be transmitted with no state transitions occurring. The device remains in normal-read state.

Data Pending States

There are three data pending states:

- Data pending enter - the device state after an operator enter action.
- Data pending read - the device state after reception of a Read Partition Query structured field.
- Data pending stacked enter - the device state after a Read Partition Query structured field was received while the device was in data-pending-enter read state or retry-enter state (the enter is stacked).

A poll received while the device is in a data-pending state causes the data to be transmitted and the device to be placed in the corresponding retry state.

Read-State Transitions

The read-state transitions for BSC are summarized in Table 9-1.

Table 9-1. Read-State Transitions for BSC

Read States		**Data Pending**			**Retry**		
Events	**Normal**	**Enter**	**Read**	**Stacked Enter**	**Enter**	**Read**	**Stacked Enter**
Enter Action	2	R	R	R	R	R	R
Read Command	1	1	1	1	G	G	G
Read Partition Query	3	4	R	R	4	R	R
Poll	—	5	6	7	—	—	—
Host Acknowledge	—	1	1	2	1	1	2
RESET key	—	1	—	3	1	—	6

Key:

R - Reject, no state transition
G - Retry, no state change
— - No action or state change
1 - Normal-read state
2 - Data-pending-enter state
3 - Data-pending-read state
4 - Data-pending-stacked-enter state
5 - Retry-enter state
6 - Retry-read State
7 - Retry-stacked-enter state

Retry States

There are three retry states:

- Retry enter - the device state after enter data was transmitted to the application program.
- Retry read - the device state after Query Reply data was transmitted to the application program.
- Retry stacked enter - the device state after enter data was stacked and the Query Reply data transmitted to the application program.

While in retry state, the last inbound transmission can be retried by means of a Read Modified command.

A host acknowledgment causes the device to revert from a retry state to the normal-read state, or, in the case of retry stacked enter, to the data-pending-stacked-enter read state.

Indicators

With reference to Table 9-1, the indicators displayed in the operator information area of a display are as follows:

State	*Indicator*
Normal read	a. No indicator or b. System Lock
Data pending enter	TWAIT
Data pending read	TWAIT
Data pending stacked enter	TWAIT
Retry enter	System Lock
Retry read	TWAIT
Retry stacked enter	TWAIT

Host Acknowledgments

Read Acknowledgment

The following are the Read Acknowledgments:

1. When the Inbound Operation is a Query or Query List

   In general, a Query or Query List operation is acknowledged by any outbound transmission *except* one with a read command.

   Thus, the following will acknowledge a Query or Query List.

   - A WSF command - with or without following structured fields. The transmission is an acknowledgment regardless of an error being detected in the accompanying structured fields, as long as the WSF is accepted.
   - An EAU command.
   - An EW, EWA, or Write with or without a WCC or data. If data is present and an error is detected in the data, the transmission is *not* an acknowledgment.

2. When the Inbound Operation is Operator Enter or a RM, RMA, or RB Partition Command.

In general, an RM, RMA, or RB Partition Command or an enter operation is acknowledged by either writing to the inbound partition (the partition associated with the inbound operation) with a transmission which restores the keyboard or destroying the inbound partition.

Thus, any of the following (assuming the acknowledging function is supported) will constitute an acknowledgment when the display is in any of the Retry *or* Data Pending States.

- If the inbound partition is 0 (implicit or explicit), either of the following:
  - An EW, EWA, or Write command with WCC = Keyboard Restore (Note 1).
  - An EAU command.
- If in the explicit partition state, an EW or EWA command with the WCC = reset (Note 1).
- A WSF command followed by an Outbound 3270DS structured field to the inbound partition with either of the following (Note 2):
  - An EW, EWA, or Write partition command with WCC = Keyboard Restore.
  - EAU partition command.
- A WSF command followed by a Destroy Partition structured field to the inbound partition, including explicit and implicit partition 0 (Note 2).
- A WSF command followed by a Create Partition structured field to the inbound partition (Note 2).
- If in implicit partition state, a WSF command followed by a Create Partition structured field (Note 2).
- A WSF command followed by an Erase/Reset structured field. Applies to both implicit and explicit partition state (Note 2).
- A Copy command (BSC only). In addition, the following constitutes a Read Acknowledgment *only* when the display is in one of the retry states.
- A WSF command followed by an Outbound 3270DS structured field, with an EW, EWA, Write or EAU partition command, directed to *any* partition. Applies whether or not the EW, EWA, or Write partition command is followed by a WCC or data (Note 2).
- An EW, EWA, Write, or EAU command. Applies whether or not a WCC or data follows the EW, EWA or Write command (Note 1).

Notes:

1. If data follows the WCC and an error is detected in the data, the transmission is *not* a Read Acknowledgment.
2. If there is a detected error prior to, or within, the structured field providing the Read Acknowledgment, the transmission is *not* a Read Acknowledgment.

Processing of Read Commands

In a BSC environment, a read command (Read Buffer, Read Modified, or Read Modified All) as the first byte of the data stream is processed:

1. If the device is in normal-read state, then the command performs a read and the display data is transmitted inbound as defined by:

   a. The Read Modified, Read Modified All, or Read Buffer command

   b. The AID (Read Modified command only)

   c. The reply mode.

For items a, b, and c above, the device remains in normal-read state.

2. If the device is in a data-pending state, then the command performs a read and data is transmitted as defined by:

   a. The Read Modified, Read Modified All, or Read Buffer command

   b. The AID (Read Modified command only)

   c. The inbound reply mode.

The device is placed in normal-read state.

3. If the device is in a retry state, then the command performs a retry as follows:

   a. If the command is Read Modified, and INOP specifies Query, then appropriate query replies are transmitted inbound.

   b. If the command is Read Modified, and INOP specifies Read Modified, then data is transmitted as defined by:

      1) The Read Modified command
      2) The AID
      3) The inbound reply mode.

   c. If the command is Read Buffer, then data is transmitted inbound as defined by:

      1) The Read Buffer command
      2) The inbound reply.

   For items a, b, and c the device remains in the original retry state.

Processing of Read Partition Query Structured Fields

Read Partition Query and the Query Reply are processed as follows:

1. If the device is in normal-read state, then:
   a. The TWAIT indicator is displayed.
   b. INOP is set to Query.
   c. The device prepares to generate the required inbound data stream.
   d. The device is placed in data-pending-read state.
   e. A later poll causes the data to be transmitted and the device to be placed in retry-read state.
2. If the device is in data-pending-enter or retry-enter state, then:
   a. The outstanding data is stacked.
   b. The TWAIT condition remains in effect.
   c. INOP is set to Query.
   d. The device prepares to generate the required inbound data stream.
   e. The device is placed in data-pending-stacked-enter state.
   f. A later poll causes the Query Reply data to be transmitted inbound and the device to be placed in retry-stacked-enter state.

BSC Copy Command

BSC uses the Copy command to accomplish a local-copy function. See Chapter 8, "Printer Considerations" for a detailed description of the operation of the Copy command.

Chapter 10. Non-SNA Environment (Locally Attached Devices—3272 Version)

This chapter explains those data stream operations that differ from the SNA version, for locally attached devices in a non-SNA environment. Most of the explanations cover only the differences between SNA and non-SNA for locally attached devices.

Commands

Except for the Read Modified All command, the commands valid for an SNA environment are also valid for the non-SNA locally attached environment.

The additional commands of Select, No Operation (No-op), and Sense are used in the non-SNA locally attached environment to improve device utilization, to retrieve pending status, and to obtain unit check definition, respectively. These commands are not 3270 data stream commands and are not discussed in this manual. See the applicable product manuals for an explanation of these commands.

Write Commands

The Erase/Write and Erase/Write Alternate commands operate the same as in an SNA environment. The Write command operates the same as in the SNA environment except that the starting buffer location depends upon the following considerations:

- The starting location may be specified by a Set Buffer Address order that follows the WCC.
- The starting location will be the buffer address containing the cursor if the Write command is not chained or if it is chained from a Select, Erase All Unprotected, No Operation, or Sense command.
- The starting location will be the current buffer address if the Write command is chained from a Read or another Write command.

Read Commands

The read commands for devices in this non-SNA environment operate the same as for an SNA environment except for the differences noted in the following text.

Read Buffer Command

Execution of the Read Buffer command causes all data in the addressed device buffer location to be transferred to main storage. The transfer of data begins:

- From buffer address 0 if the Read Buffer command is unchained.
- From the current buffer address if the Read Buffer command is chained from either a Write, Erase/Write, Erase/Write Alternate, Read Modified, or another Read Buffer command. Regardless of where the transfer of data begins, data transfer from the buffer will terminate when the last character location in the buffer has been transferred or before the last character location has been transferred when the channel byte count reaches 0 (in this case, the buffer address after termination is undefined).

Read Modified Command

Read Modified initiates one of three operations as determined by operator actions at the display: (1) Read Modified, (2) Short Read, or (3) Test Request Read.

Read-Modified Operation: The Read Modified command functions the same as in an SNA environment except that the buffer location at which the search begins for the field attribute bytes that define modified fields is a function of command chaining. This location is:

- Buffer address 0 if the Read Modified command is unchained or is chained from a Select, Sense, or No Operation command.
- The current buffer address if the Read Modified command is chained from a Write, Erase/Write, Read Modified, or Read Buffer command.

The search for modified fields ends when the channel byte count reaches 0. The transfer of data is terminated as follows:

1. If the last modified field is wrapped from the last buffer location, the operation is terminated after all the data in the field has been transferred (nulls are suppressed). The buffer address at the end of the transfer is the address of the next field attribute byte in the buffer. For example, if a modified field extends from address 1900 (the field attribute byte) to address 79 (wrapped field), the data from addresses 1901 through 79 is transferred (nulls suppressed). In this case, the read operation is terminated with the buffer address set to 80 (the field attribute byte of the next field).
2. If the buffer does not contain a wrapped modified field and if the channel byte count has not reached 0, the modified data stream is terminated when the last modified field is transferred. At the end of the operation, the buffer address is set to 0.
3. If the channel byte count reaches 0 before all modified data is transferred, read operations are terminated and the remaining modified data is not transferred. The buffer address after termination is undefined.

If the buffer is formatted but none of the fields have been modified, the read data stream consists of the 3-byte read heading only.

If the buffer is unformatted, the read data stream consists of the 3-byte read heading followed by all alphanumeric data in the buffer (nulls suppressed), even when part or all of the data has not been modified. Since an unformatted buffer contains no field attributes, no SBA codes with associated addresses or address characters are included in the data stream, and the modification of the data cannot be determined. Data transfer starts at address 0, regardless of command chaining, and continues to the end of the buffer. At the end of the operation, the buffer address is set to 0. This read operation can also be terminated by the channel byte count's reaching 0 before all data is read. In this case, the buffer address after termination is undefined.

Test Request Read Operation: The Read Modified command causes a Test Request Read operation if the TEST REQ or SYS REQ key has been pressed at the selected device. The Test Request Read data stream sent inbound to the application program is the same as for the BSC environment, except there is no ETX.

Inbound Transmissions

Inbound transmissions result from an operator enter action, a host initiated (unsolicited) read request, or a host retry of an inbound transmission.

An operator enter action is one that causes the attention identifier to be transmitted inbound. The application program responds with a read request. The application program must acknowledge the inbound transmission before a new inbound operation can be performed. (See "Host Acknowledgments.")

A host-initiated read operation is an inbound transmission not caused by an operator enter action. No host acknowledgment is required before a new inbound transmission can occur.

Host retry is a retransmission of the last unacknowledged inbound transmission from the device. The application program must acknowledge reception of an inbound transmission before a new inbound transmission can take place. A host retry transmission does not cause read-state transitions and is not considered a new inbound transmission requiring host acknowledgment. Host retry occurs until a host acknowledgment takes place.

The type of inbound transmission is either a Query Reply structured field (the reply to the Read Partition Query structured field) or data from the device buffer (for example, modified fields of the display image). An inbound operation (INOP), set by the controller, defines the type.

Inbound Operation (INOP)

The INOP determines the operation to be performed when data is transmitted inbound, or when the device is in a data-pending state.

INOP is set by any of the following:

- An operator enter action sets INOP to Read Modified.
- Reception of a Read Partition Query structured field sets INOP to query.
- Host acknowledgment of an inbound transmission sets INOP to Read Modified.

Read States

While powered on, a device is in one of seven states with respect to read operations. The three primary states are:

- Normal read
- Data pending read
- Retry read.

The data-pending read and retry read have three substates: enter, read, and stacked enter.

The events that cause transitions between the states are shown in Table 9-1 on page 9-5.

Normal Read State

A device is in normal-read state when powered on, or prior to initiation of a read operation, or after use of the RESET key in certain instances. (See Table 10-1 on page 10-5.)

When in normal-read state, an operator enter action or the reception of a Read Partition Query structured field causes the device to prepare to generate the inbound data stream, and then to go into a data-pending state.

When in normal-read state an application-program-initiated read operation using a RM, RMA, or RB command causes the data to be transmitted with no state transitions occurring. The device remains in normal-read state.

Data Pending States

There are three forms of data-pending states:

- Data pending enter: enter data has been transmitted to the application program.
- Data pending read: the device state after reception of a read-partition-query structured field.
- Data pending stacked enter: when a read-partition structured field has been received while the device is in data-pending-enter state or retry-enter state (the enter data is stacked).

A read command received while the device is in data-pending state causes the data to be transmitted inbound and the device to be placed in the corresponding retry state.

An operator enter action will not be processed, it will be rejected. A host-initiated read partition will be rejected.

Retry State

There are three forms of retry states:

- Retry enter, when enter data has been transmitted to the host.
- Retry read, when the read data has been transmitted to the host.
- Retry stacked enter, when enter data has been stacked and the query-reply data has been transmitted to the host.

While in retry state, the last inbound transmission can be retried by means of a Read Modified command.

A host acknowledgment causes the device to revert from a retry state to the normal-read state or, in the case of retry stacked enter, to the data-pending-stacked-enter-read state.

The read-state transitions for non-SNA locally attached devices are summarized in Table 10-1.

Table 10-1. Read-State Transitions for Non-SNA Locally Attached Devices—3272 Version

Read States		**Data Pending**			**Retry**		
Events	**Normal**	**Enter**	**Read**	**Stacked Enter**	**Enter**	**Read**	**Stacked Enter**
Enter Action	2	R	R	R	R	R	R
Read Command	1	5	6	7	G	G	G
Read Partition Query	3	4	R	R	4	R	R
Host Acknowledge	—	1	1	2	1	1	2
RESET key	—	1	—	3	1	—	6

Key:

R - Reject, no state transition
G - Retry, no state transition
— - No action or state change
1 - Normal Read state
2 - Data Pending Enter state
3 - Data Pending Read state
4 - Data Pending Stacked Enter state
5 - Retry Enter state
6 - Retry Read State
7 - Retry Stacked Enter state

Indicators

With reference to Table 9-1 on page 9-5, the indicators displayed in the operator information area of a display are as follows:

State	*Indicator*
Normal read state	a. No indicator or b. System Lock
Data pending enter	System Lock
Data pending read	TWAIT
Data pending stacked enter	TWAIT
Retry enter	TWAIT
Retry read	TWAIT
Retry stacked enter	TWAIT

Host Acknowledgments

Read Acknowledgment

The following are the Read Acknowledgments:

1. When the Inbound Operation is a Query or Query List.

   In general, a Query or Query List operation is acknowledged by any outbound transmission *except* one with a read command.

   Thus, the following will acknowledge a Query or Query List.

- A WSF command - with or without following structured fields. The transmission is an acknowledgment regardless of an error being detected in the accompanying structured fields, as long as the WSF is accepted.
- An EAU command.
- An EW, EWA, or Write command with or without a WCC or data. If data is present and an error is detected in the data, the transmission is *not* an acknowledgment.

2. When the Inbound Operation is Operator Enter or a RM, RMA, or RB Partition Command.

   In general, a RM, RMA or RB Partition Command or an enter operation is acknowledged by either writing to the inbound partition (the partition associated with the inbound operation) with a transmission that restores the keyboard or destroying the inbound partition.

   Thus, any of the following (assuming the acknowledging function is supported) will constitute an acknowledgment when the display is in any of the Retry *or* Data Pending States.

   - If the inbound partition is implicit or explicit partition 0, either of the following:
     - An EW, EWA, or Write command with WCC = Keyboard Restore (Note 1).
     - An EAU command.
   - If in the explicit partition state, an EW or EWA command with the WCC = reset (Note 1).
   - A WSF command followed by an Outbound 3270DS structured field to the inbound partition with either of the following (Note 2):
     - An EW, EWA, or Write partition command with WCC = Keyboard Restore.
     - An EAU partition command.
   - A WSF command followed by a Destroy Partition structured field to the inbound partition, including explicit and implicit partition 0 (Note 2).
   - A WSF command followed by a Create Partition structured field to the inbound partition (Note 2).
   - If in implicit partition state, a WSF command followed by a Create Partition structured field (Note 2).
   - A WSF command followed by an Erase/Reset structured field. Applies to both implicit and explicit partition state (Note 2).

   In addition, the following constitutes a Read Acknowledgment *only* when the display is in one of the retry states.

   - A WSF command followed by an Outbound 3270DS structured field, with an EW, EWA, Write or EAU partition command, directed to *any* partition. Applies whether or not the EW, EWA, or Write partition command is followed by a WCC or data (Note 2).
   - An EW, EWA, Write, or EAU command. Applies whether or not a WCC or data follows the EW, EWA or Write command (Note 1).

Notes:

1. If data follows the WCC and an error is detected in the data, the transmission is *not* a Read Acknowledgment.
2. If there is a detected error prior to, or within, the structured field providing the Read Acknowledgment, the transmission is *not* a Read Acknowledgment.

Processing of Read Commands

In a non-SNA (locally attached devices—3272 Version) environment, a read command (Read Buffer or Read Modified) as the first byte of the data stream is processed as follows:

1. If the device is in normal-read state, then the command performs a read and the display data is transmitted inbound as defined by:
   a. The Read Modified, or Read Buffer command
   b. The AID (Read Modified command only)
   c. The reply mode.

The device remains in normal-read state.

1. If the device is in a data-pending state:
   a. If the command is Read Modified and INOP specifies Query, the appropriate query replies are transmitted.
   b. If the command is Read Modified and INOP specifies a Read Modified, then data is transmitted as defined by:
      (1) The Read Modified command
      (2) The AID
      (3) The inbound reply mode.
   c. If the command is Read Buffer, then data is transmitted as defined by:
      (1) The command
      (2) The reply mode.

Note: For items a, b, and c, the device is placed in the corresponding retry state (Enter, Read, or Stacked Enter).

1. If the device is in a retry state, the command performs a retry as follows:
   a. If the command is Read Modified, and INOP specifies Query, then the appropriate query replies are transmitted inbound.
   b. If the command is Read Modified, and INOP specifies Read Modified, then data is transmitted as defined by:
      (1) The Read Modified command
      (2) The AID
      (3) The inbound reply mode.
   c. If the command is Read Buffer, then data is transmitted inbound as defined by:
      (1) The Read Buffer command
      (2) The inbound reply.

   For items a, b, and c, the device remains in the original retry state.

Processing of Read Partition Query Structured Fields

Read Partition Query and the Query Reply are processed as follows:

1. If the device is in normal read state:
   a. The TWAIT indicator is displayed.
   b. INOP is set to Query.
   c. A channel attention occurs.
   d. The device is placed in data-pending-read state.
   e. A later read command causes the data to be transmitted and the device to be placed in retry-read state.
2. If the device is in data-pending-enter or retry-enter state:
   a. The outstanding data is stacked.
   b. The TWAIT condition remains in effect.
   c. INOP is set to Query.
   d. A channel attention occurs.
   e. The device is placed in data-pending-stacked-enter state.
   f. A later Read Modified command causes the data to be transmitted inbound and the device to be placed in retry-stacked-enter state.

Chapter 11. Auxiliary Devices and Workstations

Introduction

The 3270 data stream was defined for use between a "host" application program and a single display. This chapter addresses the enhancement of the 3270 data stream to allow support of a 3270 data stream workstation. A 3270 data stream workstation consists of a 3270 data stream display and one or more auxiliary devices.

An auxiliary device does not accept the usual 3270 data stream, for example, commands, orders, etc. However, the 3270 data stream is used to carry the data stream associated with the auxiliary devices. The data to and from the auxiliary devices must be in the form of structured fields.

The presentation space associated with an auxiliary device is independent of the display presentation space. Data directed to an auxiliary device must not alter the display presentation space and vice versa.

This enhancement of the 3270 data stream is based on a single session between the "host" application and the 3270 workstation. The workstation has only one network address. (The extensions for supporting a workstation also apply to the non-SNA environment).

The term "device" in this section is used in a general sense, that is, a device may be:

- An actual device (for example, a printer)
- A logical device or process (for example, a DDM or DIA file).

An auxiliary device may support either an IBM data stream (for example, DDM or DIA file) or a non-IBM data stream.

1. IBM Data Streams

   A different type auxiliary device query reply is defined for each different IBM data stream used by the auxiliary devices, for example, the DDM AUX Device and DIA AUX Device query replies.

   The Query Reply will provide a reference that identifies the IBM data stream supported. For example, the DIA Query Reply indicates the DIA function sets that are supported. The actual description of the data stream is provided by the IBM documentation associated with the data stream.

2. Non-IBM Data Streams

   The 3270 data stream workstation may support a non-IBM device such as an auxiliary device. A non-IBM device is defined here as a device that is manufactured outside IBM and does not use an IBM data stream. The device may carry either an outside manufacturer's logo or an IBM logo.

   Only one Query Reply, the OEM Aux Device Query Reply, is defined for all types of non-IBM auxiliary devices.

   An OEM Data structured field is provided for carrying the data to and from a non-IBM auxiliary device. A parameter (DSREF) in the OEM Aux Device Query

Reply identifies the content of the OEM Data structured field as one of the following:

- A non-IBM data stream that the non-IBM auxiliary device sends or receives. The host application must derive what this data stream is from the DTYPE (Device type) parameter in the OEM Aux Device query reply.
- "Value added" data stream, that is, the data stream contains controls in addition to the data recognized or sent by the device. The controls are used by the 3270 data stream workstation. These controls are removed by the workstation prior to sending the non-IBM data stream to the auxiliary device. The DSREF parameter of the OEM Aux Device Query Reply provides reference identification for the "value added" data stream.

Data Routing

In single display device implementations there is no ambiguity about the destination or source of the data. The using environment provides the routing, for example, via the network address in an SNA environment.

However, with the 3270 data stream workstation there is more than one destination or source for data. The destination or source must be explicitly identified. The using environment (for example, SNA) will provide the same routing function for a 3270 data stream workstation as is currently provided for the single display implementation. Additional routing control for data is provided within the 3270 Data stream.

The Destination/Origin structured field must be used to identify the destination or origin of all data to or from auxiliary devices. The same destination/origin (DOID) value is used in the Destination/Origin structured field for sending data to or from a particular auxiliary device. Outbound, from the host application, the DOID indicates the destination of the data. Inbound, to the host application, the DOID indicates the origin of the data.

It is a 3270 data stream workstation implementation responsibility to assign each auxiliary device a unique destination DOID for use in the Destination/Origin structured field. All values except X'0000' and X'FFFF' are available for use by the 3270 data stream workstation implementation.

The display is the default destination or origin if the data destination or origin is not explicitly identified by a Destination/Origin structured field.

Data of a type not supported that is directed to the display or an auxiliary device will be rejected. For example, directing DDM data to the display or display data to the DDM auxiliary device will cause the data to be rejected.

A Destination/Origin structured field may also be used in routing data to or from the primary display (DOID = X'0000'). It is invalid to use an inbound Destination/Origin structured field from the display with (DOID=X'0000') unless the transmission also contains input from one or more auxiliary devices.

At the start of each outbound transmission the destination is the display (and at the start of each inbound transmission the origin is the display). Once a Destination/Origin structured field has established the destination/origin of the data, that destination/origin applies for all structured fields that follow until the end of the transmission or unless changed by a subsequent Destination/Origin structured field.

Query Reply

The description for each Auxiliary Device Query Reply will specify whether it is sent in reply to either a Query or Query List or only in reply to a query list.

Return of the AUXDA Query Reply indicates a 3270 data stream workstation implementation. That is, the support of the Destination/Origin structured field and one or more auxiliary devices. The AUXDA query reply is returned in reply to either a Query List =(AUXDA Equivalent or All) or to a query.

The Query Reply for the individual auxiliary devices will provide the DOID value to be used in the Destination/Origin structured field and any other required information concerning the auxiliary device.

A Query or Query List directed to an auxiliary device instead of the display will be rejected.

A separate Query Reply must be returned for each auxiliary device supported. For example, if two identical auxiliary devices were supported, a Query Reply would be returned for each. The DOID reported would be different for each.

When a 3270 data stream workstation supports an auxiliary device, the Query Reply for that device is returned regardless if the auxiliary device is available or not (for example, necessary support code is not resident, powers off, etc.).

Input Control

Some 3270 applications, particularly those not "aware" of auxiliary devices, cannot cope with unsolicited input from auxiliary devices. Therefore, the host application is given control over when an auxiliary device is permitted to send in data. The control is achieved by the INCTRL (Input Control) flag in the Destination/Origin structured field. The INCTRL flag has meaning only outbound (to the auxiliary device) and is ignored on inbound. When the Destination/Origin structured field is directed to an auxiliary device, the INCTRL flag applies to that device. When the Destination/Origin structured field is directed to the display (DOID = X'0000'), the INCTRL flag applies to all (a global application) auxiliary devices supported. In other words, it provides global control.

The default (for example, POR) is input disabled. Once input is enabled for an auxiliary device, it remains enabled until disabled by any of the following:

- The auxiliary device receives a Destination/Origin structured field with INCTRL flag = B'10' (input disable).
- The display receives a Destination/Origin structured field with INCTRL = B'10' (global input disable).
- The workstation receives an EW or EWA with WCC = Reset.
- A Clear local function (for example, the CLEAR Key is pressed).
- A power-on-reset.
- The workstation receives a Bind (SNA only).

Receiving a Destination/Origin structured field from the host application with INCTRL = B'01' causes no change in the input enabled/disabled state at the auxiliary device. Also, if the INCTRL flag value is the same as the existing input enable/disable states, the state is unchanged. For example, if the auxiliary device input enable/disable state is input enabled, receiving a Destination/Origin structured field with INCTRL = B'00' (input enable) will be accepted and the input enable/disable state remains enabled.

Note: There is one exception where an auxiliary device may send input without being enabled. An Exception Condition structured field, reporting unavailability of the auxiliary device, may be sent in reply to a Destination/Origin structured field sequence attempting to use the auxiliary device.

Auxiliary Device and Display Interaction

The auxiliary devices will conform to the Read operations described in Chapter 3 except where otherwise noted in this chapter.

When data is read in from an auxiliary device, the rules or states for read retry and read acknowledgment apply. For example, once a transmission is sent from an auxiliary device, data from that device cannot be sent inbound until a read acknowledgment is received. If the data from an auxiliary device is transmitted in multiple transmissions, each transmission requires an acknowledgment. An inbound transmission may contain data from the display or data from one or more auxiliary devices. When display data is sent in the same transmission as auxiliary device data, the Outbound 3270DS structured field must be used for the display data. An inbound transmission containing data from auxiliary devices must start with an AID of X'88' which indicates structured fields follow. The same conditions that acknowledge a Query Reply will acknowledge inbound transmissions from an auxiliary device.

An outbound transmission to an auxiliary device constitutes a read acknowledgment per the description for outbound display transmissions. The fact that the transmission is to an auxiliary device adds no additional acknowledgment function. For example, a transmission to an auxiliary device would acknowledge an outstanding Query Reply transmission because the transmission contained a WSF. As another example, in an SNA environment a transmission to an auxiliary device would constitute an acknowledgment to an outstanding enter transmission only if the transmission put the workstation in a send or contention state.

Only one display type read may occur in an outbound transmission; when in structured field form, it must be the last structured field in the transmission. A display type read is defined as any of the following:

- A Query or Query List structured field
- A Read Partition structured field
- A Read Buffer, Read Modified, or Read Modified All command.

In an outbound transmission, data to an auxiliary device (for example, a DDM file) can initiate inbound data from the auxiliary device. Inbound data can be initiated from multiple auxiliary devices by a single outbound transmission containing multiple Destination/Origin structured fields. Inbound data from one or more auxiliary devices can be initiated in an outbound transmission which also contains a display type read. When this occurs, the display type read is executed first.

A display type read always takes priority over pending inbound data from an auxiliary device. A display operator enter action is considered a display type read.

If inbound data is pending from one or more auxiliary devices, an operator enter action will take priority and use the next available inbound transmission.

When the data from an auxiliary device must be sent in multiple transmissions (for example, a transmission size limit imposed for certain data), each inbound transmission is treated like an enter, to the extent that sending of the data is initiated by the device. A host read acknowledgment is required prior to sending the next part of the data. Therefore, data from an auxiliary device which is sent in multiple transmissions could have some interspersed display transmissions. Also, the display operator must not be "locked out" as a result of an auxiliary device condition, for example, power-off, diskette removed, etc.

Exception Handling

An exception condition on an auxiliary device must not cause termination of the host to 3270 data stream workstation session. That is, an auxiliary device exception condition must not cause a negative response. The exception conditions must be reported at the application level.

In general, the exception handling is defined by the data stream used by the auxiliary device. The DIA data stream documentation defines the exception heading for a DIA auxiliary device. The DDM data stream documentation defines that for a DDM auxiliary device, and so forth.

There are some instances where the exception condition is handled within the 3270 data stream. For example, if the auxiliary device is not available, for example, power off, processing code not resident, etc., the unavailability is reported by returning a Destination/Origin structured field followed by an Error Condition structured field with the code field = X'0801' (Resource Not Available). Another example is where the host exceeds the transmission size specified in an Auxiliary Device Query Reply. In this case the code is X'084C' (permanent insufficient resource).

Chapter 12. Double-Byte Coded Character Set (DBCS) Asia

Introduction

This chapter describes the 2-byte coded character set (DBCS) operation that is defined for use by the South Eastern Asian (SEAR) countries, Japan, Korea, Taiwan, and so on. The primary use is for Kanjii.

This Double-Byte Character Set (DBCS) operation provides the capability for 1- and 2-byte character sets.

Support of DBCS-Asia is indicated by the DBCS-Asia query reply and the Character Set Query Reply.

There are a number of differences in the operation of the DBCS-Asia character sets from the remainder of the 3270 data stream. However, when the DBCS-Asia implementation is operated exclusively in the 1-byte mode, the differences are limited to the:

1. Delete key operation
2. The insert mode operation on an unformatted screen.

For the DBCS-Asia mode the delete key operates as follows: If the cursor is located in a character location in an unprotected field, the Delete key deletes the character from the character location identified by the cursor and sets the MDT bit to 1. The cursor does not move. All remaining characters, to the right of the cursor in the *same field* shift *one column or two columns* to the left, depending on whether a 1-byte character or a 2-byte character is deleted. Vacated locations at the end of the field are filled with Nulls.

In an Insert operation on an unformatted screen, it will not wrap at the end of the screen. Normal operation allows wrap at the end of the screen.

Codepoints

The following defines the code points used by the DBCS-Asia operation.

Graphic Codes

When the current character set attribute specifies a DBCS character set, the data stream values are interpreted as follows:

- X'00' through X'3F' and X'FF' are control codes.
- X'4040' is a space (blank).
- Combinations of X'41' thru X'FE' are language-dependent graphics.

There is a maximum of 1 + 190x190 graphics (X'4040' and combinations of X'41' thru X'FE') in DBCS-Asia, but there are fewer than 1 + 190x190 unique graphics in each language.

The integrity of these codepoints is maintained; that is, if read back, the same codepoints are transmitted as were received.

Two-Byte Coded Field Selection Designators

The following five 2-byte codepoints are defined as Field Selection designators for the 2-byte character set. The function is the same as for that of 1-byte character sets.

Codepoint	**Meaning**
X'0000'	Two-byte coded NULL
X'4040'	Two-byte coded Blank
X'4250'	Two-byte coded &
X'426E'	Two-byte coded >
X'426F'	Two-byte coded ?

Two-Byte Coded Printer and Control Characters

When the current character set attribute specifies a DBCS character set, each character is represented by a 2-byte codepoint. The special characters for printer controls, namely NL, EM, FF, CR, DUP, and FM, are also represented by 2-byte codes as follows:

NL - X'0015'
EM - X'0019'
FF - X'000C'
CR - X'000D'
DUP - X'001C'
FM - X'001E'

The first byte in each of the above codes is not interpreted as a null character but is included in the 2-byte coded character representation definition.

When the current character set attribute specifies a DBCS character set, a null character is defined as X'0000'.

This 2-byte coded null has the same function as two 1-byte coded nulls in Insert Mode, and it is suppressed for the read modified operation.

DBCS Fields

This section describes the support of a 2-byte field that consists of only DBCS characters.

Character Set Attribute Type and Values

The character set attribute is used with the SFE and the MF orders in the 3270 data stream to specify a character set for the field.

Type	Value
X'43'	Character Set

The following are valid settings for the character set attribute value byte.

Codepoint	Meaning
X'00'	Default character set.
X'40' to X'EF'	Local ID for loadable character set.
X'F0' to X'F7'	Local ID for nonloadable 1-byte character set.
X'F8' to X'FE'	Local ID for nonloadable 2-byte character set.

Character set X'F8' is assigned to the DBCS character set. The sequence of *SFE-X'43'-X'F8'* defines a DBCS field.

Data Stream Processing

The graphic code points in the data stream reference the character set that is currently in effect.

The decision on whether a graphic code point is treated as a 1-byte character or as a 2-byte character is determined by the "look left" rule. This means that in the buffer the application looks back towards the beginning of the buffer for the extended field attribute (formatted buffer). If the buffer is unformatted, the graphic code points are treated as referencing the default character set, if when "looking left" in the buffer, the first buffer position is reached without encountering an SI or SO.

To achieve the desired results when using 2-byte fields DBCS-Asia requires that the appropriate Extended Field Attribute (character sets) be established before loading data into the field. If the application does not follow this procedure, there is a possibility of rejecting the transmissions.

Exception Conditions

If both bytes of a 2-byte graphic code are not in the range of X'40' through X'FE' the transmission is rejected with sense code 1003 or sense code 1001. Transmissions with invalid 2-byte representations of control codes will be rejected with sense code 1003 or sense code 1001.

Two-byte codes cannot be separated into two 1-byte codes with an order inserted between the two halves. If this is done, the order will be considered as the second half of a 2-byte code and the transmission will be rejected with sense code 1003 or sense code 1001.

In a DBCS field, sending the sequence of a Null-Start Field or Null-Start Field Extended does not cause an exception condition. This sequence is used for filling the "dead position." The dead position is a single-byte position that can occur at the end of a DBCS unprotected field and in which the operator is prevented from entering data.

If the end of processing of a Write, EW, or EWA command or of an Outbound 3270DS structured field (with Write, EW, or EWA) occurs after receiving the first half of a 2-byte code, the transmission is rejected with sense code 1005 or sense code 1001.

A transmission that attempts to start writing a 2-byte character into a buffer position that is the second half of a 2-byte character will be rejected with sense code 1005 or sense code 1001. One exception is that the second half of a 2-byte character may be replaced with a field attribute by an SF or SFE without error.

If a Character Attribute, type=character set, other than X'00' is received in the 2-byte field, the transmission is rejected.

If the 2-byte field contains a Character Attribute, type=character set, other than X'00', the data stream will be rejected during post processing. The Character Attribute, type=character set, will be changed to X'00'.

Display Buffer Manipulation

As stated previously, DBCS-Asia assumes that an application will properly establish a field as 1-byte or 2-byte before attempting to enter data into the field. DBCS-Asia also assumes, for example, that an application will not change a field from 1-byte to 2-byte (or vice versa) without also updating the contents of the field in the same transmission. This section defines what results when such buffer manipulations are performed by the application.

A transmission that only changes an extended field attribute from a 1-byte character set to a 2-byte character set will change how the code points are treated. That is, if the field character set is changed from 1-byte to 2-byte while leaving the field contents unchanged, the code points will be treated as forming 2-byte characters and the character attribute, type=character set, will be changed to default. Similarly, if the field character set is changed from 2-byte to 1-byte, the code points will be treated as 1-byte characters.

Also, the way the code points in a 2-byte field are treated is changed by moving the field attribute (or adding a new attribute = 2-byte character set) within the field such that the attribute is an odd number of buffer positions from the attribute location. The code points still are treated as forming 2-byte characters but the pairing is shifted from the original pairing. For example, the "new" 2-byte characters are formed by pairing the second half of an "old" 2-byte character with the first half of the following "old" 2-byte character.

When a field contains control codes (code points below X'40'), manipulation of the field attribute as described above can result in leaving the buffer with invalid DBCS characters. When this occurs, the invalid characters are handled as follows:

- No error response occurs.
- What is displayed for the invalid DBCS is device dependent.
- On a read operation, an X'4040' is sent inbound for the invalid character.

Also, the manipulation of a field attribute can result in a single, non-null code point being left over at the end of a DBCS field, for example, a non-null in the dead position. A non-null in the dead position is handled as follows:

- No error response occurs.
- What is displayed for the non-null is device dependent.
- On a read buffer, the non-null dead position is sent inbound as an X'00' (null). On a read modified operation, the non-null dead position is treated as if it were a null and suppressed.

Operator Interface

Spanning Rows

When a field spans rows, a DBCS character may span rows in the character buffer. That is, the first half of the character is in the rightmost column and the second half is in the leftmost column. One approach for handling this situation is that the left half of the DBCS character is displayed in the rightmost column on screen and the right half is displayed in the leftmost column of the next row. However, an implementation may provide a different approach.

For example, if the device has an additional column on the screen (for example, 81st column for 80 column displays), a DBCS character is displayed in the rightmost column and the additional (81st) column on the screen. The leftmost column in the next row on the screen is reserved.

Cursor

There are two types of cursors used in DBCS-Asia: a cursor that is one character cell long (referred to as a short cursor) and a cursor that is two character cells long (referred to as a long cursor). When an operator positions a cursor in a DBCS field, the cursor is automatically two character cells long.

Cursor Move Keys

An operator cannot position a long cursor starting on the second half of a DBCS character. Any attempt by the operator to position a long cursor starting on the second half of a DBCS character will cause the cursor to jump to the next position where the operator can enter data.

Delete Key

For DBCS-Asia, the delete key operates as follows:

If the cursor is located in a character location in an unprotected field, the Delete key deletes the character from the character location identified by the cursor and sets the MDT bit to 1. The cursor will not move. All remaining characters, to the right of the cursor in the *same field* shift *one column or two columns* to the left depending on whether a 1-byte character or a DBCS character is deleted. Vacated locations at the end of the field are filled with Nulls.

MSR

Data input into the DBCS field by the MSR is inhibited.

DBCS Character Attribute (SA)

This section describes the support of a DBCS character set character attribute. The interaction with the other methods of selecting a DBCS character set (DBCS field or Shift Out [SO]/Shift In [SI]) is described in the section of other methods.

The Set Attribute (SA) Order, with the character set attribute type X‘43’ and its attribute value X‘F8’, sets the current character set character attribute to a DBCS character set.

Shift Out (SO)/Shift In (SI)

This section defines the operation of the Shift Out (SO) and Shift In (SI) controls. SO = X'0E' and SI = X'0F'. The SO/SI controls do not have any effect in the data stream. However, the SO/SI controls are stored in the display buffer and *do* affect how the data in the buffer is interpreted, displayed, and read. The buffer locations from SO to SI are termed a DBCS subfield.

Support of SO/SI is reported in the DBCS-Asia query reply. The character set associated with the SO is the DBCS character set, which has a SET ID = X'80' and an LCID = X'F8'. The Set ID of this character set is reported in the DBCS-Asia query reply. The LCID and CGCSGID associated with the SET ID is reported in the Character Set Query Reply.

Referencing the Character Set

- Formatted Screen:
  - If there is no SO or SI in the buffer between the extended field attribute (EFA) and a subsequent buffer position N, the code point in buffer position N references the character set designated by the EFA.
  - In moving from a buffer position N left (back) to the EFA, if the first SO/SI encountered is an SI then the code point in buffer position N references the character set designated by the EFA.
  - In moving from a buffer position N left to the EFA, if the first SO/SI encountered is an SO then the code point in the buffer position N references the character set associated with SO, which is the DBCS character set.
- Unformatted Screen:
  - If there is no SO or SI between the first buffer position and a subsequent buffer position N, then the code point in buffer position N references the default character set.
  - In moving from buffer position N left to the first buffer position, if the first SO/SI encountered is an SI then the code point in buffer position N references the default character set.
  - In moving from buffer position N left to the first buffer position, if the first SO/SI encountered is an SO then the code point in buffer position N references the DBCS character set.

Exception Conditions For SO/SI

The following rules apply to the use of SO/SI:

- SO/SI must appear as SO-SI pairs in the buffer (unformatted screen) or in a field (formatted screen). To be paired, the SO must precede the SI, for example the buffer sequence

  ```
  FA - - - - SI - - - -SO - - - FA
  ```

  is invalid. Note that SO/SI do not have to be paired in the data stream. Unpaired SO/SI can occur in a buffer update.
- SO/SI must not be sent to a DBCS field.

- In a DBCS subfield, the buffer locations with a character attribute type of character set (= X'00') must be in contiguous pairs to accommodate the 2-byte characters.
- In a DBCS subfield, the buffer contents with a character attribute type of character set (= other than X'00') will be rejected.

There are some additional SO/SI exception considerations associated with the PT, RA, and EUA 3270 orders; see "Orders" on page 12-12.

The validity of SO/SI use is verified as the data stream is processed, and again at the end of processing the command or the Outbound 3270DS structured field. This second validity checked is termed "post processing."

Data Stream Processing

A transmission will be rejected and processing stopped immediately for any of the following SO/SI error conditions:

- An SO is received, and when looking left in the buffer an SO is encountered prior to encountering one of the following:
  - An SI
  - The first buffer position (unformatted buffer)
  - An FA or EFA (formatted buffer).
- An SI is received, and when looking left in the buffer an SO is not encountered before encountering one of the following:
  - An SI
  - The first buffer position (unformatted buffer)
  - An FA or EFA (formatted buffer).
- SO or SI is received and the EFA for the field designates DBCS character set.

Post Processing

The validity of SO/SI is checked again at the end of the processing of a command or of an Outbound 3270DS structured field.

A transmission will be rejected during post processing if any of the following conditions exist in the display buffer.

- Looking left from an SO, another SO is encountered before encountering either an SI, the first buffer position (unformatted buffer) or an FA or EFA (formatted buffer).
- Looking left from an SI, either an SI, the first buffer position (unformatted buffer) or an FA or EFA is encountered prior to encountering an SO.
- An SO or SI in a DBCS field.
- Within a DBCS subfield, the positions with the character attribute=field inherit are not contiguously paired.
- Within a DBCS subfield, the positions with the character attribute=other than X'00' are found.

Data integrity is not maintained for those errors detected during post processing since the buffer has already been altered.

Set Attribute (SA) Order and SO/SI Interaction

The SO/SI controls do not apply to buffer positions where an SA has set the character set character attribute of the code point to something other than default.

The character set character attribute set by an SA does not apply to SO and SI themselves, because they are control codes. However, the other character attributes (for example, color or highlighting) do apply because SO and SI take buffer positions (for example, NL or CR).

SO/SI Creation by Operator

The application program has control over whether or not SO/SI can be created as a result of operator entry of data. The control is provided via the Input Control attribute (type = X'FE'). This attribute may only be set on a field basis by either the SFE or MF orders. The Input Control attribute format is shown below:

Attribute Type	Attribute Value
X'FE'	Input Control

The Input Control attribute value has a 1-byte value:

Content	**Meaning**
X'00'	SO/SI creation disabled
X'01'	SO/SI creation enabled.

The default (for example, power on) is SO/SI creation disabled. Note that the Input Control attribute only controls the creation of SO/SI by the operator. This attribute has no effect on SO/SI use in the outbound or inbound data stream. For example, even if the SO/SI creation is disabled for a field, the application can send SO/SI to the field and the SO/SI will appear in a read of the field.

If the operator is entering 2-byte data, and the use of both SO/SI and a character attribute is "valid," the SO/SI will be used to identify the 2-byte data. For example, if the operator selects the 2-byte character set associated with SO and enters 2-byte data in an unprotected field (Extended Field Attribute = a character set other than that associated with SO), then if SO/SI creation is enabled, the SO/SI will be used to identify the 2-byte data, even if the reply mode = character. If the SO/SI creation is disabled and the reply mode = character, the character attribute will be used to identify the 2-byte data. If the SO/SI creation is disabled and the reply mode = field or extended field, input will be inhibited unless the operator is entering into a DBCS subfield that the application created.

Types of Field

The following table summarizes how operator entered DBCS characters are identified in the display buffer for different conditions.

It is assumed that an operator has selected the DBCS character set. The data is being entered into an unprotected field that contains no DBCS subfields.

Conditions			Two-Byte Data Identified Via			
Field Character Set	Reply Mode	SO/SI Creation	SO/SI	CA	EFA	Input Inhibited
Not the same as SO	Field or extended field	Disabled				X[1]
Not the same as SO	Character	Enabled	X			
Not the same as SO	Character	Disabled		X		
Not the same as SO	Character	Enabled	X			
Same as SO	Field or extended field	Enabled or disabled			X	
Same as SO	Character	Enabled or disabled			X[2]	

[1] If the application creates DBCS subfields within the field, a operator an then enter DBCS data into the subfields. The entered DBCS data would be identified in the display buffer by the SO/SI delimiters.

[2] If the operator selects the DBCS character set, then X = the character attribute. If the operator selects field inherit, then X = the extended field attribute.

Graphic Character Input

This section describes the operation in a field with a 1-byte character set and with the SO/SI Creation set on.

An operator cannot position a short cursor on the second half of a DBCS character nor a long cursor starting on the second half of a DBCS character. Any attempt by the operator to position a short cursor on the second half of a DBCS character or a long cursor starting on the second half of a DBCS character will cause the cursor to jump to the next position where the operator can enter data.

Outside DBCS Subfield: If the operator selects the DBCS character set and inputs a character into a position outside a DBCS subfield, the sequence created is:

SO, the character, SI (if required to maintain the SO/SI pairing).

The sequence starts with the position where the cursor is and after the operation the cursor is positioned following the entered character.

For example, the buffer contains:

FA	C1/C2	C3/C4	C5	C6	FA

Where:

FA = Field Attribute
Cx/Cy = DBCS characters
Cx = One-byte characters

The operator selects the DBCS character set, positions the cursor under C2 and C3, and enters C7/C8. The buffer becomes:

FA	C1	SO	C7/C8	SI	C6	FA

The cursor will be under SI and C6.

Inside DBCS Subfield: When the operator selects the DBCS character set, the operation is as described previously. If the SI is overwritten, an SI is added following the character.

When the operator selects a character set other than the DBCS character set, the operation can be summarized as follows:

- The DBCS subfield will be terminated with an SI.
- The character is entered after the SI. If it is a 1-byte character, a space will be automatically added in the next position and an SO is automatically added after the space, except when the character is entered into the last 2-byte character position of the DBCS subfield.
- If the character is entered into the last 2-byte position of the DBCS subfield, the SO would not be added.

For example, the buffer contains:

FA	SO	Ca/Cb	Cc/Cd	Ce/Cf	Cg/Ch	Ci/Cj	SI

If the operator positions the short cursor under Cc (the first half of the DBCS character Cc/Cd) and enters a 1-byte character, C1. The buffer becomes:

FA	SO	Ca/Cb	SI	C1	sp	SO	Cg/Ch	Ci/Cj	SI

A space will be automatically added and the cursor will be under the space.

Delete Key

The operation of Delete key on an SO or an SI itself is inhibited since the deletion of it changes the characteristics of the subsequent characters. The operation of Delete key on the sequence of SOSI, or SISO is allowed.

Erase EOF Key

When the cursor is at a DBCS character in a DBCS sub-field, the operation of Erase EOF key erases the character and all subsequent characters within the field, but re-creates SI in the cursor position.

Insert Mode

Nulls in the Sub-field: When there are no DBCS subfields in an unprotected field, a character may be inserted provided there are sufficient nulls in the field from (and including) the cursor position to the end of the field — one null for a 1-byte character and four nulls for a DBCS character. If there are sufficient nulls, and a character is inserted, the character formerly occupying the cursor location and all remaining characters within the field (except for nulls and characters to the right of nulls) will be shifted one or two locations to the right.

When there is a DBCS subfield in an unprotected field, the Insert Mode operation is as described above except *the nulls in the subfield cannot be used for the insert operation*. That is, if there are not sufficient nulls from (and including) cursor location to the end of the field, other than nulls in the subfield, an insert operation is inhibited. When there are sufficient nulls outside the DBCS subfield and an insert operation occurs, the nulls in the subfield are shifted right the same as the characters in the subfield.

Therefore, to avoid possible operator problems the host application program *should not create nulls in a DBCS subfield*.

Note that such nulls are never created by operator action.

Unformatted Screen: The Insert Mode operation in the unformatted buffer does not wrap at the right-lower corner of the screen. This definition is unique to DBCS-Asia and it applies to a device which supports DBCS-Asia even when the host is operating the device in a 1-byte mode.

Note that the Delete key and the Erase EOF key also work to the right lower corner of the screen.

Automatic Delete Operation

The following case shows the automatic delete operation.

The buffer contains:

FA	C1	C2	SO	C3/C4	C5/C6	SI	FA

Operator puts long cursor under C2 and SO, and enters Cx/Cy. The buffer becomes:

FA	C1	SO	Cx/Cy	C5/C6	SI	nu	FA

C4 (the second half of C3/C4) is deleted and the remainder of the DBCS subfield is shifted left one position. A null is added at the end of the field.

MSR

Data input into the DBCS subfield by the MSR is inhibited.

Orders

In this section the unique conditions for DBCS in processing 3270 orders is described.

Start Field (SF) Order

This order cannot be used for defining DBCS fields. It operates the same for DBCS-Asia.

Set Buffer Address (SBA) Order

This order sets the current buffer address. The buffer address can be set for the second half of a DBCS character. However, if an attempt is made to write graphic data to this address, the data stream is rejected with a sense code of 1005 or a sense code of 1001.

Insert Cursor (IC) Order

This order sets the cursor to the current buffer address. When the cursor address is set to the second half of a DBCS character, it is re-positioned to the first half at the end of processing of the command or Outbound 3270DS structured field, without informing the host.

Program Tab (PT) Order

This order operates the same for DBCS-Asia except that the padding function is done when the PT order follows SI. If the PT order follows SO the data stream is rejected with a sense code of 1003 or a sense code of 1001.

Repeat to Address (RA) Order

This order repeats a character in the buffer from the current buffer address to the specified stop address (not inclusive).

The character to be repeated may be either a 1-byte or a DBCS character. The RA order sequence does not explicitly indicate whether the repeated character is 1-byte or DBCS. Therefore, the decision is made implicitly on the basis of how the first byte following the stop address would be treated if received in place of RA.

- If the byte would have been treated as a 1-byte code, then the byte is repeated per the RA order.
- If the byte would have been treated as the first byte of a DBCS character, then the first 2 bytes following the stop address are repeated per the RA order.

If the current buffer address is set to the position for the second half of an existing DBCS character and the order is encountered in the data stream, the data stream is rejected with a sense code 1005.

When the order repeats a DBCS character, the number of positions to be repeated must be even. That is, the stop address minus current buffer address must be even. If not, the data stream is rejected with a sense code 1005 or a sense code of 1001 or 1005.

An RA order sequence with either SI or SO as the repeated character is invalid and will be rejected with a sense code of 1003 or a sense code of 1001 or 1005.

Erase Unprotected to Address (EUA) Order

An EUA order is not executed and will be rejected for any of the following conditions:

- Where the execution of the EUA would have resulted in an odd number of nulls in a DBCS field (except Dead Position) or DBCS subfield.
- Where the start address is the second half of a DBCS character.
- Where the start address is in a DBCS subfield and the stop address is in a following field.

A single (unpaired) Null must not be created by EUA in the DBCS field, except for the dead position.

Start Field Extended (SFE) Order

This order is used for defining DBCS fields. Also, it is used to specify a new attribute type, Input Control. There is nothing unique for the SFE operation in DBCS-Asia.

Modified Field (MF) Order

This order is used to specify a new attribute type, Input Control. There is nothing unique for the MF operation in DBCS-Asia.

Appendix A. SNA Sense Codes for 3270 Data Stream Commands and Orders

The SNA sense codes for 3270 data stream command and order errors are listed in Table A-1. For non-SNA environments, all 3270 data stream errors are rejected with an Op-Check (OC) except for invalid commands. Unsupported commands are rejected with Command Reject (CR). Supported commands that cannot be executed are rejected with an Op-Check.

Table A-1. SNA Sense Codes for 3270 Data Stream Command and Order Errors

	SNA Sense Codes (Hex)			
	1003	1005	0863	Other
Commands:				
Invalid command code	X			
Data following read command or EAU	X			
Orders (valid for all orders):				
Invalid data stream order	X			
Incomplete parameter list or parameter missing		X		
Invalid code point to be repeated by RA	X			
Invalid address		X		
RA with GE and no data following		X		
Start Field Extended:				
Invalid attribute type		X		
Invalid color or highlighting attribute value	X			
Unknown character set attribute value in range X'01' - X'FE'		X		
Character set attribute value = X'FF'	X			
Field Actions Attribute value is reserved	X			
Modify Field:				
Current buffer location does not contain field-attribute character		X		
Invalid attribute type	X			
Invalid color or highlighting attribute value	X			
Unknown character set attribute value in range X'01' - X'FE'		X		
Character set attribute value = X'FF'	X			
Field Actions Attribute value is reserved	X			
Set Attribute:				
Invalid attribute type	X			
Invalid color or highlighting attribute value	X			
Unknown character set attribute value in range X'01' - X'FE'		X		
Character set attribute value = X'FF'	X			
Set Buffer Address:				
Address not in form specified for partition (12-, 14-bit)		X		
Invalid address flag in 12-, 14-bit mode (high-order bits = B'10')		X		

Appendix B. SNA Sense Codes for 3270 Data Stream Structured-Field Errors

The SNA sense codes for structured-field errors are listed in Table B-1 on page B-2. An implementation may optionally return the X'1001' (RU data error) code for all X'1003' (function not supported) and X'1005' (parameter error) conditions.

Where an entry in the table is shown without any sense codes (for example, Erase/Reset), the only applicable sense code conditions are those shown in the Structured Fields (All) section.

The following are included for clarification:

- The validity of the field attribute is not checked.
- Write, Erase/Write, Erase/Write Alternate, and Write Structured Field commands without data are accepted without error, but are treated as a no-op.

Table B-1 (Page 1 of 4). SNA Sense Codes for Structured-field Errors	
Structured Field Error	**Sense Code (HEX)**
STRUCTURED FIELDS (All)	
Invalid structured field type	1003/1001
Missing structured field parameter	1005/1001
Reserved field is not zero	1003/1001
RCV state and all windows have zero extent	0883
PID is unknown partition	1005/1001
Incorrect Length	1005/1001
ACTIVATE PARTITION	1005/1001
Zero extent window	
CREATE PARTITION	
PID > maximum allowed	1005/1001
Invalid UOM value	1003/1001
Invalid A-MODE	1003/1001
Specification error in Viewport/Window	1005/1001
Base > Supported number	1005/1001
Insufficient resource	1005/1001
CS=X'0002'-X'FFFE' and horizontal windowing not supported	1001
RS=X'0002'-X'FFFE' and vertical windowing not supported	1001
DESTINATION/ORIGIN	
DESTROY PARTITION	1005/1001
PID > maximum allowed	
ERASE/RESET	
FMH-1	
Invalid field value	1008
Incomplete header	1008
RH indicates FMH-1 present but no FMH appears	1008
LOAD COLOR TABLE	
Table exhausted prior to LCTDEF.	1003/1001
Attempted to modify preserved entry.	1005/1001
Table modified but not LCTID.	1005/1001
LOAD LINE TYPE	
Unsupported UNITS value	1003/1001
Table modified but not LTTID	1005/1001
Invalid Parameter Length	1005/1001
Unsupported Parameters	1003/1001
Insufficient Storage	084C

Table B-1 (Page 2 of 4). SNA Sense Codes for Structured-field Errors	
Structured Field Error	**Sense Code (HEX)**
LOAD PROGRAMMED SYMBOL SET	
Invalid Data Type	1003/1001
Invalid LCID. (Values X'40' through X'EF' are not considered invalid)	1003/1001
CHAR not available	1005/1001
RWS not available	084C
RWS is valid but not installed	084C
EXTN not supported	1005/1001
P-length incorrect	1005/1001
Byte 8, bits 3-7 not zero	1003/1001
Invalid LH or LW	1005/1001
Invalid SUBSN	1003/1001
Byte 12	084C
Invalid ST.SUBSN	1005/1001
Excess bits in data	1005/1001
Compressed Data Terminator incorrectly specified	1005/1001
PS resource not available	084C
ECHAR not provided	1005/1001
ECHAR reached prior to exhaustion of data	1005/1001
Insufficient Storage	084C
SET not supported	1003/1001
NW or NH value too large	1005/1001
MODIFY PARTITION	
PID > maximum allowed	1005/1001
Invalid reserved bits	1003/1001
Specification error in Window/Viewport	1005/1001
Invalid windowing parameters	1005/1001
OBJECT DATA	
Spanning Error	1005/1001
Unsupported MODE	1003/1001
Invalid Command Code	1005/1001
Maximum Entity Length Exceeded	1003/1001
REP/APPEND = B'01'	1005/1001
Entity (REP) Length Exceeded	1005/1001
Specified Entity does not exist	1005/1001
Invalid Parameter Length	1005/1001
Unsupported Parameters	1003/1001
Insufficient Storage	084C
Insufficient Data	1005/1001
OBJECT CONTROL	
Spanning Error	1005/1001
Unsupported MODE	1003/1001
Invalid Command Code	1005/1001
Maximum Entity Length Exceeded	1003/1001
Invalid Parameter Length	1005/1001
Unsupported Parameters	1003/1001
Insufficient Storage	084C
Insufficient Data	1005/1001

Table B-1 (Page 3 of 4). SNA Sense Codes for Structured-field Errors

Structured Field Error	**Sense Code (HEX)**
OBJECT PICTURE	
Spanning Error	1005/1001
Unsupported MODE	1003/1001
Invalid Command Code	1005/1001
REP/APPEND = B'10'	1005/1001
Segment Storing not supported	1003/1001
Maximum Entity Length Exceeded	1003/1001
REP/APPEND = B'01'	
Entity (REP) Length Exceeded	1005/1001
Specified Entity does not exist	1005/1001
Invalid Parameter Length	1005/1001
Unsupported Parameters	1003/1001
Insufficient Storage	084C
Insufficient Data	1005/1001
OEM DATA [1]	
OUTBOUND TEXT HEADER	
PID > maximum allowed	1005/1001
Invalid command	1003/1001
Violation of PS integrity	082B
Invalid WCC bit	1003/1001
Unsupported control in header	1003/1001
Parameter error in header	1005/1001
Inconsistent controls in header	1005/1001
Graphic in header	1005/1001
Cursor column > partition width	1005/1001
Incorrect A-MODE	1005/1001
OUTBOUND TYPE1 TEXT	
PID > maximum allowed	1005/1001
Unsupported control	1003/1001
Parameter error in control	1005/1001
Incorrect A-MODE	1005/1001
Insufficient resource	084C
OUTBOUND 3270 DATA STREAM	
Incorrect A-MODE	1005/1001
Invalid CMD byte	1003/1001
Violation of presentation space integrity	082B
Data following EAU partition command	1003/1001
WCC Print Bit = 1, but this is not the last structured field	1001
PRESENT ABSOLUTE/RELATIVE FORMAT	
Invalid Format Presentation Command	1003/1001
Group not selected	1009
No formats loaded	0868
Format not found	0869
Format offset out of range	1005/1001

[1] When the OEM Data structured field is used in conjunction with the Destination/Origin structured field (i.e., for data to/from a direct accessed OEM auxiliary device), this table does *not* apply. In this situation, all exception conditions associated with the OEM Data structured field must be reported either through the OEM data stream or with the Exception/Status structured field. Use of the subsystem -rsp (or OP-Chk) is not allowed.

Table B-1 (Page 4 of 4). SNA Sense Codes for Structured-field Errors

Structured Field Error	Sense Code (HEX)
READ PARTITION	
Invalid Read Type code	1003/1001
Invalid Q-code	1003/1001
PID not equal to X'FF' for Query	1005/1001
Read Partition not last structured field in chain	1005/1001
Device in Re-try state	0871
RU Chain containing READ PARTITION does not specify CD	0829
RU Chain containing READ PARTITION does specify EB	0829
REQUEST RECOVERY DATA	
RESET PARTITION	
RESTART	
SAVE/RESTORE FORMATS	
Received chain containing Save/Restore Structured Field = primary/save which does not specify CD.	0829
Received chain containing Save Restore Format Structured Field = primary/save which specifies EB.	0829
Save/Restore Format Structured Field not last structured field in chain.	1005/1001
SCS DATA	
SELECT COLOR TABLE	
SELECT FORMAT GROUP	
SELECT IPDS MODE	
When in 3270 mode, receive a Select IPDS Mode structured field which is not immediately after a WSF or a WSF, Data Chain structured field (=begin).	1003/1001
SET CHECKPOINT INTERVAL	
SELECT LOAD COLOR TABLE	
Table exhausted prior to LCTDEF.	1003/1001
Attempted to modify preserved entry.	1005/1001
Table modified but not LCTID.	1005/1001
SET MSR CONTROL	
Invalid magnetic data or length specification	1001
Invalid encode character set or length specification	1005/1001
Encode aborted by operator	0824
Not last structured field in a chain or chain sent with neither RQD or CD	0843
SET REPLY MODE	
Invalid reply mode	1003/1001
Invalid attribute type in A-list	1003/1001
Incorrect Reply mode for partition type	1005/1001
SET WINDOW ORIGIN	
Window spec. outside presentation space	1005/1001
CW > 0, but horizontal windowing not supported	1005/1001

Appendix C. Reset Actions

Table C-1 summarizes the reset actions performed as a result of the actions listed.

Table C-1 (Page 1 of 2). Reset Actions

Action	Partitions	Inbound Reply Mode	Highlighting Color Character Set		INPID INOP	PS Content PS LCID
			Selection	Indicators		
Jump Key	6	1, 3	NC	7	NC	NC
CLEAR Key SSCP	9	9	9	9	NC	NC
CLEAR Key Unowned	R	ID	R	7	NC	NC
CLEAR Key LULU	R	R	ID	7	9	NC
System Request Key	SSCP (Unowned)	9	13	9	9	NR
NC						
System Request Key SSCP (LULU)	9	9	9	9	R	NC
System Request Key Unowned	R	R	R	7	NR	NC
Receipt of RU	R	R	R	7	NR	NC
System Request Key LULU	R	R	ID	7	NR	NC
Test Key ENTER	R	R	R	7	NR	NR
Test Key EXIT	9	9	R	7	NR	15
WCC in EW/EWA	R	R	ID	7	R	NC
WCC Reset in 3270DS EW/EWA Only	NC	12	NC	11	16	NC
Reset Partition	NC	12	NC	11	16	NC
Power On	R	13	NR	R	NR	R
Clear LULU	10	10		NA, 10	10	10
DACTLU LULU	ID	ID	ID	ID	NR	NC
DACTLU SSCP	ID	ID	ID	ID	NR	NC
ACTLU SSCP	9	9	9	9	NR	NC
DACTLU Unowned	ID	ID	ID	ID	NR	NC
ACTLU Unowned	ID	ID	ID	ID	NR	NC
Unbind LULU (8)	17	13	NC	7	NR	NC
Unbind SSCP	9	9	9	9	NR	NC
Bind SSCP	9	9	9	9	R	NC
Bind Unowned	R	R	R	R	R	NC
Set Inbound Reply Mode	NC	2	NC	7	NC	NC
Clear Partition	4	12	NC	7	16	NC
Destroy Active Partition	6	1	ID	7	16	NC

Table C-1 (Page 2 of 2). Reset Actions						
			Highlighting Color Character Set			
Action	Partitions	Inbound Reply Mode	Selection	Indicators	INPID INOP	PS Content PS LCID
Destroy Not-Active Partition	5	5	NC	NC	16	NC
Clear Partition Key LULU	NC	NC	NC	NC	9	NC
CD/EB Write Acknowledgment	NC	NC	NC	NC	R	NC

ID = Implementation defined (effect visible on external interface).
NA = Not Applicable.
NC = No change.
NR = Not relevant (will be changed by subsequent action).
R = Reset (effect visible on external interface).

Notes:

1. The inbound reply mode is set to the mode of the newly activated partition.
2. The inbound reply mode is changed to the mode described in the structured field.
3. The alternate character set is changed to the mode of the newly activated partition.
4. Add the partition name to the list along with its attributes.
5. Delete the partition name from the list along with its attributes. Reset if it is the last partition.
6. Change the active partition to the next in the list. Reset if it is the last partition.
7. Display exactly those attribute selection indicators that are honored as a result of the inbound reply mode in the current partition. If the inbound reply mode is reset, no operator selection is displayed.
8. In the unowned state, the last application write is still displayed (that is, the operator may be able to add data and perform a local copy).
9. The state will already be set.
10. No change. SNA does not permit Clear in the FM layer.
11. Reset if the active partition; otherwise, make no change.
12. Reset the referenced partition only.
13. Change the inbound reply mode to allow all selections.
14. Reset when the test mode alters the programmed symbols.
15. Reset if directed to INPID.
16. Screen remains—operator interaction restricted to the active partition.

Note that a response of 082B also causes resetting of all the functions shown in the figure, except for PS LCID and PS CONTENT.

Appendix D. 12-, 14-, and 16-Bit Addressing

The SNA 3270 data stream allows 12-, 14-, and 16-bit addressing. With 12-bit addressing, an address is created from 2 bytes of binary information. The 6 low-order bits of each byte are joined to provide a 12-bit address. The address specifies the buffer position, not the line and column position on the display surface. For example, on a 480-character display, the buffer addresses are 0 to 479. To specify a 12-bit buffer address of 160 (binary 000010100000), bits 2-7 of the first byte are set to 000010; bits 2-7 of the second byte are set to 100000:

xx000010	xx100000

The 12-bit binary value is a combination of the two 6-bit values of the first and second bytes.

With 14-bit addressing, an address is created from 2 bytes of binary information, but the 14 bits of the address are contiguous. For example, an address of 800 decimal (X'320') would be represented as follows:

xx000011	00100000

With 16-bit addressing, all bits in both bytes are used. For example, an address of 3100 decimal (X'C1C') would be represented as follows:

00001100	00011100

SNA products (displays) that support only 12-bit addressing ignore bits 0 and 1 of each address byte in outbound data streams. For inbound data streams, they should set these bits in accordance with Figure D-1 on page D-2., although the bits provide no function other than retaining compatibility with previously written applications.

When a product supports both 12- and 14-bit addressing, bits 0 and 1 of the first address byte are flag bits and have the following significance:

xx Setting	Meaning
B'00'	14-bit binary address follows
B'01'	12-bit coded address follows
B'10'	Reserved
B'11'	12-bit coded address follows

For inbound data streams that contain 12-bit addresses, the display generates the B'01' or B'11' setting, using Figure D-1 on page D-2.

A partition may be defined (using the Create Partition structured field) to operate with either 16-bit addressing or 12- and 14-bit addressing. When 16-bit address mode is specified in Create Partition, bits 0 and 1 are part of the address, outbound buffer addresses are interpreted as 16-bit binary, and inbound addresses are generated as 16-bit binary. If no partitions are defined, 12- and 14-bit addressing is assumed.

The SNA 3270 data stream can be EBCDIC or ASCII. With ASCII, each of the allowable characters is a graphic symbol; that is, there are no unprintable characters in the data stream. In addition, only 12-bit addressing is used with ASCII. Figure D-1 shows the 64 binary values permitted, using bits 2 through 7, and defines how they are transformed into ASCII values. Using the example of 160 decimal as each byte are set in accordance with Figure D-1.

Field attributes and write control characters also require special processing for conversion from binary values to graphic symbols when ASCII or 12-bit addressing is used.

Bits 2-7	EBCDIC	ASCII	Bits 2-7	EBCDIC	ASCII
00 0000	40	20	10 0000	60	2D
00 0001	C1	41	10 0001	61	2F
00 0010	C2	42	10 0010	E2	53
00 0011	C3	43	10 0011	E3	54
00 0100	C4	44	10 0100	E4	55
00 0101	C5	45	10 0101	E5	56
00 0110	C6	46	10 0110	E6	57
00 0111	C7	47	10 0111	E7	58
00 1000	C8	48	10 1000	E8	59
00 1001	C9	49	10 1001	E9	5A
00 1010	4A	5B	10 1010	6A	7C
00 1011	4B	2E	10 1011	6B	2C
00 1100	4C	3C	10 1100	6C	25
00 1101	4D	28	10 1101	6D	5F
00 1110	4E	2B	10 1110	6E	3E
00 1111	4F	21	10 1111	6F	3F
01 0000	50	26	11 0000	F0	30
01 0001	D1	4A	11 0001	F1	31
01 0010	D2	4B	11 0010	F2	32
01 0011	D3	4C	11 0011	F3	33
01 0100	D4	4D	11 0100	F4	34
01 0101	D5	4E	11 0101	F5	35
01 0110	D6	4F	11 0110	F6	36
01 0111	D7	50	11 0111	F7	37
01 1000	D8	51	11 1000	F8	38
01 1001	D9	52	11 1001	F9	39
01 1010	5A	5D	11 1010	7A	3A
01 1011	5B	24	11 1011	7B	23
01 1100	5C	2A	11 1100	7C	40
01 1101	5D	29	11 1101	7D	27
01 1110	5E	3B	11 1110	7E	3D
01 1111	5F	5E	11 1111	7F	22

Figure D-1. Conversion of Binary Values to Hexadecimal Values That Obtain Graphic Symbols

Appendix E. Special Applications

This appendix contains information on how some implementations that do not conform to the 3270 Data Stream report those deviations to the host.

Query Reply (Anomaly Implementation)

Function

Indicates that an implementation of a function does not conform to the 3270 data stream operation for that function.

When the Anomaly Implementation query reply is supported, it is transmitted inbound in reply to a Read Partition structured field specifying Query List (QCODE List [=X'9D'], or All).

Although a host application must accept the Anomaly Implementation query reply, the associated anomaly implementation may not be supported.

Format

Byte	Content	Content Description
0-1	L	Length of structure
2	X'81'	Query reply
3	X'9D'	Anomaly Implementation
4	RES	Reserved - must be zeros
5	ANREF	Anomaly reference number
6-7	LIMIN	Maximum bytes/inbound transmission
8-9	LIMOUT	Maximum bytes/outbound transmission
10	LPARM	Length in bytes of device dependent parameters (includes LPARM byte)
11-n	DPARMS	Device dependent parameters

Additional Content Description

- **ANREF** - Provides the reference number assigned to the anomaly implementation. Refer to the associated product documentation for the description of the anomaly implementation.
- **LIMIN** - Indicates the maximum number of bytes of Anomaly Auxiliary Device information the host application can expect in an inbound transmission.

  The sum of bytes contained in all the structured fields associated with the Anomaly Auxiliary Device following an AID X'88' will be equal to or less than the value specified (in hexadecimal) in LIMIN.
- **LIMOUT** - Indicates the maximum number of bytes of Anomaly Auxiliary Device information the host application is allowed to send in an outbound transmission.

  The sum of bytes contained in all the structured fields associated with an Anomaly Auxiliary Device, following a WSF command, must be equal to or less than the LIMOUT value (in hexadecimal). If this limit is exceeded, the transmission will be rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound data to the Anomaly Auxiliary Device.
- **LPARM** - Gives the number of bytes of device-dependent parameters + the LPARM byte.
- **DPARMS** - Provides the device-dependent information associated with the anomaly implementation.

3270 PC Application-to-Application

Function

This version of the Anomaly Implementation Query Reply is approved for use by release 2.0 of the 3270 PC.

Format

Byte	Content	Content Description
0-1	X'0019'	Length of structure
2	X'81'	Query reply
3	X'9D'	Anomaly Implementation
4	RES	Reserved - must be zeros
5	ANREF X'01'	Anomaly reference number for a 3270 PC Application-to-Application
6-7	LIMIN	Maximum bytes of 3270 PC application data that will be sent in an inbound transmission
8-9	LIMOUT	Maximum bytes of 3270 PC application data allowed in an outbound transmission
10	X'0F'	Length in bytes of device-dependent parameters (includes LPARM byte)
11-12	DOID	Destination/Origin ID
13-24	APLNME	Application name

Additional Content Description

- **ANREF**

  - X'01' - 3270 PC Application/Application.

  Refer to the *IBM 3270 PC Control Program Programming Guide*, SA23-0221, for a description of the anomaly implementation.

- **LIMIN** - The sum of bytes contained in all the structured fields associated with the Anomaly Auxiliary Device, following an AID X'88', will be equal to or less than the value specified (in hexadecimal) in LIMIN.

- **LIMOUT** - The sum of bytes contained in all the structured fields associated with an Anomaly Auxiliary Device following a WSF command must be equal to or less than the LIMOUT value (in hexadecimal). If this limit is exceeded, the transmission will be rejected. Note that the data received prior to reaching the limit may have been processed. A LIMOUT value of X'0000' indicates no implementation limit on outbound data to the Anomaly Auxiliary Device.

- **DOID** - 2-byte ID provided by the 3270 PC for use in the Destination/Origin structured field used in routing data to or from the 3270 PC application.

- **APLNME** - 12-byte EBCDIC name assigned to the application by the 3270 PC.

Appendix F. Functions Required for Systems Application Architecture (SAA) Support

This appendix contains a list of the functions that the 3270 data stream must support for Systems Application Architecture (SAA). These functions are those required for Extended Function Base Support (EBASE) in the 3270 data stream. Each function is listed in this appendix with a pointer to the specific section(s) of this manual that explain that function.

To read about SAA, refer to *Systems Application Architecture: An Overview*, GC46-4341. Other SAA publication titles are provided in the list of related publications in the front of this book.

Query Replies

The query replies listed below are necessary for SAA support. See Chapter 6, "Inbound Structured Fields," for information about these query replies.

Query Reply (Character Sets)
Query Reply (Implicit Partition)
Query Reply (Null)
Query Reply (Summary)
Query Reply (Usable Area)

Structured Fields

The structured fields listed below are necessary for SAA support. See Chapter 5, "Outbound Structured Fields," for information about these structured fields.

Read Partition
Erase/Reset
Outbound 3270DS

Basic 3270 Commands

The list below is of 3270 commands that are necessary for SAA support. Chapter 3, "3270 Data Stream Commands," contains information about these commands.

Erase All Unprotected (EAU)
Erase/Write
Erase/Write Alternate
Read Buffer
Read Modified
Read Modified All
Write
Write Structured Field

Basic 3270 Orders

The orders shown below are those required for SAA support. They are explained in Chapter 4, "3270 Data Stream Orders and Attributes."

Start Field
Set Buffer Address
Program Tab
Insert Cursor
Repeat to Address
Erase Unprotected to Address

3270 Controls/Special Characters

The controls and special characters shown below are those that must be present for SAA support. They are shown and explained below.

Order	Meaning	EBCDIC	ASCII	Displayed as
NUL	NULL	X'00'		A space, suppressed on Read Modified
SUB	Substitute Character	X'3F'	-----	A solid circle
DUP	Duplicate	X'1C'	X'1C'	An overscore asterisk
FM	Field Mark	X'1E'	X'1E'	An overscore semicolon
FF	Form Feed	X'0C'	X'0C'	A space
CR	Carriage Return	X'0D'	X'0D'	A space
NL	New Line	X'15'	X'0A'	A space
EM	End Media	X'19'	X'19'	A space
EO	End Op	X'FF'	-----	A space

NUL is read back as a null (X'00') on a Read Buffer operation but is not read back on Read Modified operations.

NL, EM, FF, and CR are printer control codes with no display function. However, the code must be supported to the extent of being accepted and, on reading back, must appear as NL, EM, FF, and CR, respectively. All are displayed as a space.

FM and DUP are displayed as above. When read back, they appear as the FM and DUP codes.

SUB, FM, and DUP may be entered from the keyboard. They are stored in the display buffer as controls; the current character set selection has no effect on them. They are transmitted to the application program as control codes.

Some of these control codes provide a print format function when they are received by a printer. The control codes and their functions are:

NL (New Line). Moves the print position horizontally to the left margin and vertically down to the next line

CR (Carriage Return). Moves the print position horizontally to the left margin

EM (End of Message). Terminates the print operation

FF (Form Feed). Moves the print position to the top and left margin of the next page.

List of Abbreviations

A

A. (1) Ampere. (2) Attention.

AID. Attention identifier.

APA. All points addressable.

B

B. Busy.

BSC. Binary synchronous communication.

C

C. Celsius.

C&D. Cause and diagnostic (codes).

CECP. Country extended code page.

CECP. Country extended code page.

CGCSGID. Coded Graphic Character Set Global Identifier

D

D. Display.

DBCS. Double-byte character set.

E

EAU. Erase all unprotected.

EUA. Erase Unprotected to Address.

EWA. Erase/write alternate.

F

F. Fahrenheit.

I

I. Information (format).

IC. Insert Cursor.

ID. Identification, identifier.

in. inch (or inches).

I/O. Input/output.

L

l. Coaxial cable type for indoor or outdoor installations.

L. Left.

LCID. Logical channel identifier.

LT. Logical terminal.

LU. Logical unit.

M

MF. Modify field.

N

No. Number.

NUL. The null character.

NUM. Numeric.

P

PT. Program Tab.

PA. Program access key.

PLU. Primary logical unit.

PS. Programmed symbol.

PWAIT. Partition wait.

Q

QCODE. Query code.

R

RA. Repeat to address.

RB. Read buffer.

RH. Request/response header.

RM. Read modified.

RMA. Read modified all.

RPQ. Request for price quotation.

RU. Request/response unit.

S

S. Sequenced (format), side.

SA. Selection addressing.

SAA. Systems Application Architecture

SBA. Set Buffer Address.

SCS. SNA character string.

SF. (1) Special feature. (2) Specify feature. (3) Start field.

SFE. Start Field Extended.

SNA. Systems Network Architecture.

T

TH. Transmission header.

TWAIT. Terminal wait.

U

U. Unprotected.

U.S. United States.

W

WCC. Write control character.

WSF. Write Structured Field.

Glossary

This glossary includes terms and definitions from the *IBM Dictionary of Computing: Information Processing, Personal Computing, Telecommunications, Office Systems, IBM-specific Terms*, SC20-1699.

The terms in this glossary are defined here as they apply to the 3270 Information Display System.

A

access method. A technique for moving data between main storage and input/output devices.

activate partition.

active. Able to communicate on the network. An adapter is active if it is able to pass tokens on the network.

active logical terminal (LT). In MLT, the currently displayed logical terminal. Synonymous with *foreground logical terminal.* Contrast with *background logical terminal.*

address. (1) A value that identifies a register, a particular part of storage, a data source, or a data sink. The value is represented by one or more characters. (2) To refer to a device or an item of data by its address. (3) In word processing, the location, identified by an address code, of a specific section of the recording medium or storage. (4) The location in the storage of a computer where data is stored. (5) In data communication, the unique code assigned to each device or work station connected to a network.

alphanumeric field. A field that may contain any alphabetic, numeric, or special characters.

application. The use to which an information processing system is put, for example, a payroll application, an airline reservation application, or a network application.

application program. (1) A program written for or by a user that applies to the user's work, such as a program that does inventory control or payroll. (2) A program used to connect and communicate with stations in a network, enabling users to perform application-oriented activities.

attention (ATTN). An occurrence external to an operation that could cause an interruption of the operation.

attention identifier (AID). (1) A code in the inbound 3270 data stream that identifies the source or type of data that follows. (2) A character in a data stream indicating that the user has pressed a key, such as Enter, that requests an action by the system.

attribute. (1) A characteristic. (2) A terminal display language or transformation definition language (TDL) keyword that specifies a particular quality for the TDL object with which it is associated.

attribute type. In the 3270 data stream, a code that identifies the properties from which an associated set of attribute values can be selected. See also *extended color.*

attribute value. In the 3270 data stream, a code immediately following the attribute type that specifies a particular property from the set defined by the attribute type.

audible alarm. (1) An alarm that is sounded when designated events occur that require operator attention or intervention before system operation can continue. (2) A special feature that sounds a short, audible tone automatically when a character is entered from the keyboard into the next-to-last character position on the screen. The tone can also be sounded under program control.

B

background logical terminal (LT). In MLT, any logical terminal that is not currently displayed. Contrast with *active logical terminal (LT).*

base color. The capability of displaying or printing all characters in a field, in one of four colors, on a color terminal by use of combinations of the field protection and the field intensify bits of the field attribute.

binary synchronous communications (BSC). Data transmission in which character synchronism is controlled by timing signals generated at the sending and receiving stations.

blink. An extended highlighting attribute value (for emphasis) of a field or character.

buffer. (1) A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another. (2) An isolating circuit used to prevent a driven circuit from influencing the driving circuit. (3) To allocate and schedule the use of buffers. (4) A portion of storage used to hold input or output temporarily.

buffer address. The address of a location in the buffer.

C

channel-attached. Pertaining to attachment of devices directly by data channels (I/O channels) to a computer. Synonym for *local*. Contrast with *telecommunication-attached.*

character attribute. The properties of a character with respect to its color, highlighting, and character set. See also *extended field attribute.*

character buffer. The read/write storage used by a partition for storing character or graphic data for display or printing on a terminal.

character position. A location on the screen at which 1 character can be displayed; also, an addressed location in the buffer at which 1 character can be stored.

character set. (1) A defined collection of characters. (2) A group of characters used for a specific reason, for example, the set of characters a printer can print. (3) The collection of graphic characters required to support a specific language.

code page. An assignment of graphic characters and control function meanings to all code points.

command. An instruction that directs a control unit or device to perform an operation or a set of operations.

control character. (1) A character whose occurrence in a particular context specifies a control function. (2) A character used to specify that a control unit is to perform a particular operation.

Control (CTL) diskette. A customized diskette containing the microcode that describes a particular control unit's attached terminals, and its method of attachment to the host.

control unit. A general term for any device that provides common functions for other devices or mechanisms. The 3174 is an example of a control unit.

country extended code page (CECP). A function of the 3174 microcode that provides for a code page containing additional code points beyond those available with Table 5A code pages. CECP is supported by a universal character set, Character Set 697, which contains 190 characters.

create. In 3174 central site customizing, to create a library member for a network control unit, and store the customizing data for that library member on a Library diskette.

cursor. (1) A movable, visible mark used to indicate the position at which the next operation will occur on a display surface. (2) A unique symbol that identifies a character position in a screen display, usually the character position at which the next character to be entered from the keyboard will be displayed.

D

data stream. (1) All data transmitted through a data channel in a single read or write operation. (2) A continuous stream of data elements being transmitted, or intended for transmission, in character or binary-digit form, using a defined format. See also *data stream format*.

data stream format. In SNA, the format of the data elements (end-user data) in the request unit (RU). See also *3270 data stream* and *SNA character string (SCS)*.

designator character. A character or space that immediately follows the field attribute character in a detectable field to denote either a selection field or an attention field. The designator character controls whether a detect on the field will or will not cause an attention. For a nonattention-producing field, the designator character also determines whether the modified data tag for the field is to be set or reset as a result of a selector-pen detect.

detectable. An attribute of a display field.

device. A mechanical, electrical, or electronic contrivance with a specific purpose.

diskette. A flexible magnetic disk enclosed in a protective container.

display field. (1) An area in the display buffer that contains a set of characters that can be manipulated or operated upon as a unit. (2) A group of consecutive characters (in the buffer) that starts with an attribute character (defining the characteristics of the field) and contains one or more alphanumeric characters. The field continues to, but does not include, the next attribute character.

display frame. (1) In computer graphics, an area in storage in which a display image can be recorded. (2) In computer micrographics, an area on a microform in which a display image can be recorded.

display station. An input/output device containing a display screen and an attached keyboard that allows a user to send information to or receive information from the system.

E

Erase All Unprotected (EAU) command. A 3270 data stream command that erases all unprotected fields and inserts nulls.

Erase Unprotected to Address (EUA) order. A data stream order that erases all unprotected character positions (inserts nulls) from the current buffer address up to, but not including, the specified stop address.

extended color. (1) A capability that allows color terminals to display or print fields or characters in colors using extended field and character attributes. (2) An attribute type in the extended field attribute and character attribute.

extended field attribute. Additional field definition to the field attribute that controls defining additional properties; for example, color, highlighting, character set, and field validation. The extended field attribute is altered by information passed in the Start Field Extended and Modify Field orders.

extended highlighting. (1) A function that provides blink, reverse video, and underscore for emphasizing fields or characters on devices supporting extended field attributes and character attributes. (2) An attribute type in the extended field attribute and character attribute. (3) An attribute passed between session partners in the Start Field Extended, Modify Field, and Set Attribute orders.

F

field. See *display field.*

field attribute. A control character stored in the character buffer in the first character position of a field. For those devices supporting the 3270 data stream, a field attribute defines protected/unprotected, alphanumeric/numeric, detectable/nondetectable, display/nondisplay, intensity, and modified data tag (MDT).

file. A named set of records stored or processed as a unit.

foreground logical terminal (LT). Synonym for *active logical terminal (LT).*

frame. (1) The portion of a tape, on a line perpendicular to the reference edge, on which binary characters can be written or read simultaneously. Synonymous with *tape row*. (2) A housing for machine elements. (3) The hardware support structure, covers, and all electrical parts mounted therein that are packaged as one entity for shipping. (4) A formatted display. See *display frame.*

I

input/output (I/O). (1) Pertaining to a device whose parts can perform an input process and an output process at the same time. (2) Pertaining to a functional unit or channel involved in an input process, output process, or both, concurrently or not, and to the data involved in such a process. (3) Pertaining to input, output, or both.

Insert Cursor (IC) order. An order that displays the cursor at the current buffer address.

intensified display. An attribute of a display field; causes data in that field to be displayed at a brighter level than other data displayed on the screen.

interface. (1) A shared boundary between two functional units, defined by functional characteristics, common physical interconnection characteristics, signal characteristics, and other characteristics as appropriate. (2) A shared boundary. An interface may be a hardware component to link two devices or a portion of storage or registers accessed by two or more computer programs. (3) Hardware, software, or both, that links systems, programs, or devices.

L

light pen. A light-sensitive pick device that is used by pointing it at the display surface.

link. The logical connection between nodes including the end-to-end link control procedures.

local. Pertaining to a device accessed directly without use of a telecommunication line. Synonym for *channel-attached*. Contrast with *remote*.

location. With reference to a 3174, a place within the 3174 chassis where a particular card or adapter is inserted.

logical terminal (LT). In MLT, one of five sessions available to share one display station.

M

main storage. Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent processing.

mark. A symbol or symbols that indicate the beginning or the end of a field, a word, an item of data or a set of data such as a file, record, or block.

memory. Program-addressable storage from which instructions and other data can be loaded directly into registers for subsequent execution or processing. Synonymous with *main storage*.

N

network. (1) An arrangement of nodes and connecting branches. Connections are made between data stations. (2) A configuration of data processing devices and software connected for information interchange.

P

programmed symbol set (PSS). A set of fonts that can be system-defined or defined by the user and to which a code can be assigned.

programmed symbols (PS). In the 3270 Information Display System, an optional feature that stores up to six user-definable, program-loadable character sets of 190 characters each in terminal read/write storage for display or printing by the terminal.

Program Tab (PT) order. An order that advances the current buffer address to the address of the first character location of the next unprotected field and resets the character attributes of all characters in the field that are replaced by nulls.

protected field. (1) In word processing, preset data or an area that cannot be changed or overridden by an operator without altering the program. (2) On a display device, a display field in which a user cannot enter, modify, or erase data. Contrast with *unprotected field*.

R

remote. Pertaining to a system, program, or device that is accessed through a telecommunication line.

Repeat to Address (RA) order. An order that stores a specified alphanumeric or null character in up to 480 buffer locations, starting at the current buffer address and ending at, but not including, the specified stop address.

request for price quotation (RPQ). An alteration or addition to the functional capabilities that the control unit provides.

S

selector pen. A pen-like instrument that can be attached to a display station. When a program using full-screen processing is assigned to the display station, the pen can be used to select items on the screen or to generate an attention. Synonym for *light pen*.

Set Attribute (SA) order. (1) An order that specifies an attribute-type-value pair defining the property to be applied to subsequent characters in the data stream. An SA order is required for each property assigned. (2) An order that associates attributes in the EAB with individual characters.

Set Buffer Address (SBA) order. An order that sets the buffer address to a specified location.

SNA character string (SCS). A character string composed of EBCDIC controls, optionally intermixed with end-user data, that is carried within a request/response unit.

Start Field Extended (SFE) order. (1) A data stream order that defines the start of a field that includes extended field attribute type-value pairs. (2) An order that generates an extended field attribute in the EAB and at the current buffer location.

Start Field (SF) order. (1) A data stream order that establishes the start of a data field for displaying or printing. (2) An order that indicates a specified location that contains an attribute byte and not a text character.

station. (1) An input or output point of a system that uses telecommunication facilities; for example, one or more systems, computers, terminals, devices, and associated programs at a particular location that can send or receive data over a telecommunication line. (2) A location in a device at which an operation is performed, for example, a read station. (3) In SNA, a link station.

storage. A unit into which recorded text can be entered, in which it can be retained and processed, and from which it can be retrieved. See also *memory*.

structured field. A data stream format that permits variable-length data and controls to be parsed into its components without having to scan every byte.

subsystem. A secondary or subordinate system, or programming support, usually capable of operating independently of or asynchronously with a controlling system. The 3174 and its attached terminals are an example of a subsystem.

synchronous. (1) Pertaining to two or more processes that depend on the occurrences of a specific event, such as common timing signal. (2) Occurring with a regular or predictable time relationship.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

T

telecommunication-attached. Pertaining to the attachment of devices by teleprocessing lines to a host processor. Synonym for *remote*. Contrast with *channel-attached*.

terminal. In data communication, a display station or printer capable of sending or receiving information.

Type. In the 3174 Subsystem Control Unit, the identifying number of a card. For example, 9150 is the type number of the terminal adapter in the 3174.

U

unprotected field. A displayed field in which a user can enter, modify, or delete data. Contrast with *protected field*.

W

write. To make a permanent or transient recording of data in a storage device or on a data medium.

write control character (WCC). A character used in conjunction with a Write command to specify that a particular operation, or combination of operations, is to be performed at a display station or printer.

Write Structured Field (WSF) command. A command used to transmit data in structured field format.

3

3270 data stream. (1) The commands, control codes, orders, attributes, and data or structured fields for 3270 devices, that are transmitted inbound to an application program or outbound to a terminal. (2) Data being transferred from or to an allocated primary or tertiary device, or to the host system, as a continuous stream of data and 3270 Information Display System control elements in character form.

Index

C

D

E

K

L

M

T

U

V

IBM 3270 Information Display System
Data Stream Programmer's Reference

Publication No. GA23-0059-4

READER'S COMMENT FORM

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

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

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, date, and location of your local IBM branch office:

____

____

____

____

____

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.)

GA23-0059-4

Reader's Comment Form

Fold and tape **Please Do Not Staple** **Fold and tape**

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
Dept. E02
P.O. Box 12195
Research Triangle Park, N.C. 27709-9990

Fold and tape **Please Do Not Staple** **Fold and tape**

IBM®

GA23-0059-4

File Number
S360/S370/53/4300/8100-30

GA23-0059-04

Printed in U.S.A.