3270 Information Display System

GA23-0059-07

Data Stream Programmer's Reference

3270 Information Display System

GA23-0059-07

Data Stream Programmer's Reference

3174 Licensed Internal Code

Licensed Internal Code remains the property of IBM and is to be used only in accordance with the Agreement for Purchase of IBM Machines or the Agreement for Lease or Rental of IBM Machines, whichever Agreement is applicable.

In this document, the term **3174 microcode** refers to Licensed Internal Code.

Eighth Edition (June 1992)

This major revision obsoletes and replaces GA23-0059-06. See "Summary of Changes" on page xvi for the changes made to this book. Changes or additions to the text and illustrations are indicated by a vertical line to the left of the changes.

Changes are made periodically to the information herein. Before using this publication in connection with the operation of IBM systems or equipment, consult your IBM sales representative or the latest *IBM System/370, 30xx, and 4300 Processors Bibliography*, GC20-0001, for editions that are applicable and current.

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 Reader's Comment Form is provided at the back of this publication. If the form has been removed, address 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.

Contents

Figures

Tables

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM licensed program in this publication is not intended to state or imply that only IBM's licensed program may be used. Any functionally equivalent program may be used instead.

Preface

This book describes 3270 data stream operations; however, your particular device may not implement the described function. Consult your product library for information about the 3270 data stream functions your product uses.

Who Should Use This Book

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

Programmers who write the access method macro instructions or other input/output (I/O) instructions should use this book 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 thirteen chapters and six appendixes:

Chapter 1: Introduces the 3270 data stream, gives an overview of it, and discusses 3270 data stream concepts.

Chapter 2: Describes partitions and functions related to partitions.

Chapter 3: Describes the commands used in the 3270 data stream and their operations. It also designates the specific commands that are necessary for SAA support.

Chapter 4: Describes the orders and attributes used in the 3270 data stream and how they function. It also explains character set properties and identifies specific orders that are necessary for SAA support.

Chapter 5: Describes the outbound and outbound/inbound structured fields in alphabetic order, gives their syntax, and describes how they function. The chapter also designates outbound structured fields that are necessary for SAA support.

Chapter 6: Describes the inbound structured fields, including Query Replies, in alphabetic order, gives their syntax, and describes how they function. This chapter also designates inbound structured fields that are necessary for SAA support.

Chapter 7: 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: Describes printer functions with the 3270 data stream and the Local Copy function for Systems Network Architecture (SNA) and binary synchronous communications (BSC).

Chapter 9: Discusses a BSC environment and describes the differences in operation between a BSC environment and an SNA environment for the 3270 data stream.

Chapter 10: Discusses a non-SNA environment of locally attached devices (3272 version) and describes the differences in operation in an SNA or BSC environment.

Chapter 11: Describes the enhancements of the 3270 data stream that allow support of 3270 data stream workstations.

Chapter 12: Describes the 2-byte coded character set defined for use in South East Asia Region (SEAR) countries, Japan, Korea, and Taiwan.

Chapter 13: Describes the Local Format Storage function.

Appendix A: Summarizes the sense codes returned for data stream command, order, and structured field errors.

Appendix B: Summarizes the reset actions for the 3270 data stream.

Appendix C: Explains the addressing used in the 3270 data stream.

Appendix D: Describes how applications that do not implement 3270 data stream operations report those deviations to the host.

Appendix E: Describes the functions that the 3270 data stream must support for SAA.

Appendix F: Indexes all commands, orders, format control orders, structured fields, and SCS control codes by hexadecimal value.

Summary of Changes

Eighth Edition (June 1992)

Information has been changed or added in this edition to reflect architectural changes in the 3270 data stream for Read operations, Zero Length Structured Fields, Set Reply Mode Structured Fields, Early Print Complete operations, and the Delete Key.

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

Where To Find More Information

3270 General Information

The following publications provide a general introduction to the 3270 Information Display System:

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

- *IBM 3270 Information Display System*
  3174 Establishment Controller Introduction, GA27-3850
- *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 5080 Graphics System Principles of Operation*, GA23-2012
- *IBM AIX Workstation Host Interface Program*
  User's Guide and Reference, Version 1.1, SC23-2060
- *IBM Systems Reference Library*
  General Information - Binary Synchronous Communications, GA27-3004
- *IBM 3179 Color Display Station Description*, GA18-2177
- *IBM 3192 Display Station Description*, GA18-2535
- *IBM 3193 Display Station Description*, GA18-2364
- *IBM 3194 Device Functional Interface Programming Guide*, SA23-0314
- *IBM 3278 Display Station Description*, GA18-2127
- *IBM 3290 Information Panel Description and Reference*, GA23-0021
- *IBM Office Information Architectures: Concepts*, GC23-0765.

Systems Application Architecture

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
- *Systems Application Architecture: Common Communications Support Summary*, GC31-6810
- *Font Object Content Architecture Reference*, S544-3285
- *Intelligent Printer Data Stream Reference*, S544-3417.

Coded Character Sets

The following publications contain information on valid Coded Character Set Identifier parameter values and other aspects of Character Data Representation Architecture:

- *Character Data Representation Architecture - Level 1, Reference*, SC90-1390
- *Character Data Representation Architecture - Level 1, Executive Overview*, SC90-1392
- *Character Data Representation Architecture - Level 1, Registry*, SC90-1391.

Document Interchange Architecture

The following publications provide a description of the Document Interchange Architecture functions:

- *Document Interchange Architecture: Technical Reference*, SC23-0781
- *Document Interchange Architecture: Interchange Document Profile Reference*, SC23-0764
- *Document Interchange Architecture: Transaction Programmer's Guide*, SC23-0763.

3174 Establishment Controller

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

To Find Translations of Safety Notices:

Safety Notices, GA27-3824

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
3174 Introduction, GA27-3850

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, 3R, 11L, 11R, 12L, 12R, and 13R User's Guide, GA23-0337
Models 21L, 21R, 22L, and 23R User's Guide, GA27-3874
Models 51R, 52R, 53R, 61R, 62R, and 63R User's Guide, GA23-0333
Models 81R, 82R, 90R, 91R, and 92R User's Guide, GA23-0313

To Plan for and Customize the 3174:

Configuration Support A and S

Planning Guide, GA27-3844
Utilities Guide, GA27-3853
Central Site Customizing User's Guide, GA23-0342
Asynchronous Emulation Adapter Description and Reference, GA27-3872

Configuration Support B

Planning Guide, GA27-3862
Model 90R Tokenway Planning, GD21-0036
Utilities Guide, GA27-3863
Central Site Customizing User's Guide, GA27-3868
Asynchronous Emulation Adapter Description and Reference, GA27-3872

Configuration Support C

Planning Guide, GA27-3918
Utilities Guide, GA27-3920
Central Site Customizing User's Guide, GA27-3919
Asynchronous Emulation Adapter Description and Reference, GA27-3872
Configuration Support C Release 2.0 User's Guide, GA27-3966

To Perform Problem Determination:

Customer Problem Determination, GA23-0217
Status Codes, GA27-3832
Configuration Support C Release 2.0 User's Guide, GA27-3966

To Install Features or Convert Models on the 3174:

Fixed Disk Installation and Removal Instructions, GA27-3864
Diskette Drive Installation and Removal Instructions, GA23-0263
Device Control Adapters Installation and Removal Instructions, GA23-0265
Model Conversion Instructions, GA23-0295
Token-Ring Network Feature Installation and Removal Instructions, GA23-0329
Storage Expansion Feature Installation and Removal Instructions, GA23-0330
Communication Adapter Installation and Removal Instructions, GA27-3830
Asynchronous Emulation Adapter Installation and Removal Instructions, GA23-0341
Concurrent Communication Adapter and Integrated Services Digital Network Adapter Installation and Removal Instructions, GA27-3851
Models 21L, 21R, 22L, and 23R Feature Installation and Removal Instructions, GA27-3875

To Use the Asynchronous Emulation Adapter Feature:

Asynchronous Emulation Adapter Description and Reference, GA27-3872
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

To Obtain Data Stream Programming and Reference Information:

Functional Description, GA23-0218
Data Stream Programmer's Reference, GA23-0059
Asynchronous Emulation Adapter Description and Reference, GA27-3872
3174 Reference Summary, GX27-3872
3174 Character Set Reference, GA27-3831
3270 X.25 Operation, GA23-0204
Configuration Support C Release 2.0 User's Guide, GA27-3966

To Perform Maintenance (Service Personnel):

Models 1L, 1R, 2R, 3R, 11L, 11R, 12L, 12R, 13R Maintenance Information, SY27-2572
Models 21L, 21R, 22L, and 23R Maintenance Information, SY27-0323
Models 51R, 52R, 53R, 61R, 62R, and 63R Maintenance Information, SY27-2573
Models 81R, 82R, 90R, 91R, and 92R Maintenance Information, SY27-2584
CE Reference Summary, SX27-3873
Status Codes, GA27-3832

Other Publications

The following publications are available for the 3174. They are developed by the International Technical Support Center. The intended audience for these books are IBM System Engineers and Customer Network Planners.

IBM 3174 Establishment Controller Installation Guide, GG24-3061.

NetView Distribution Manager Release 2 and 3174 Central Site Change Management Implementation Guide, GG24-3424.

IBM 3174 CECP Migration Issues, GG24-3380.

Chapter 1. Introduction to the 3270 Data Stream

Introduction

The 3270 data stream controls the processing and formatting of data with commands, orders, control characters, attributes, and structured fields. This chapter describes the format and content of the 3270 data stream. It also describes the importance of the device buffer in sending and receiving data.

The 3270 data stream operations are designed primarily for transmitting data between an application program and a display with a keyboard. This book describes the 3270 data stream operations as they apply to displays. With certain exceptions, the operations described in this book also apply to printers. These exceptions are described in Chapter 8, "Printer Operations."

This book describes how the 3270 data stream operates in a Systems Network Architecture (SNA) environment. With certain exceptions, the operations described in this book also apply to Binary Synchronous Communications (BSC) and other non-SNA environments. These exceptions are described in Chapter 9, "Binary Synchronous Communications (BSC) Environment," and Chapter 10, "Non-SNA Environment (Locally Attached Devices—3272 Version)."

Format of the 3270 Data Stream

The data stream transmitted between an application program and a terminal has the following format for SNA:

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. The data stream can be *outbound* or *inbound*. An *outbound* data stream is a data stream sent from the application program to the device and has the following format:

Command	Write Control Character	Data

OR

Write Structured Field	Structured Field	....	Structured Field

An *inbound* data stream is sent from the device to the application program and consists of an attention identifier (AID) followed by data or an AID (X'88') followed by structured fields. The AID describes the action that caused the transmission of the inbound data stream.

The inbound data stream has the following format:

AID	Cursor Address (2 bytes)	Data

OR

AID (X'88')	Structured Field	....	Structured Field

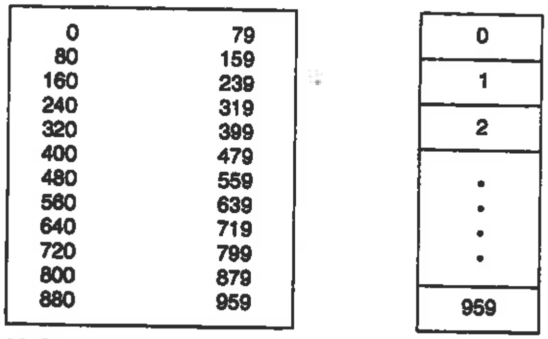
The *data* shown in this format is the information transferred between the application program and the display. It can be used or modified by either the application program, the operator, or both. In many cases it is optional.

The Device Buffer

The 3270 data stream depends on the presence of a mapped character buffer in the device. Data received from the application program and data to be transmitted to the application program are stored in a device buffer and displayed on the screen in the form of alphanumeric characters and symbols. The displayed data is updated when the operator modifies the buffer data and when new data is received from the application program.

Each character storage location in the buffer maps to a character position on the display. 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. Before a character can be entered, the cursor must be positioned in an unprotected field. If a character already exists at the current cursor (and device buffer) position, that character is overwritten by the new character (except in insert mode).

For example, if the display screen has 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. Figure 1-1 shows the character positions for a 12-row, 80-column screen on the display and in the buffer.



0	79
80	159
160	239
240	319
320	399
400	479
480	559
560	639
640	719
720	799
800	879
880	959

80-Character-per-Line Display

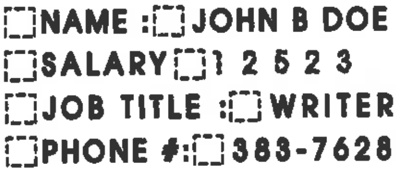
960-Byte Character Buffer

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

The character buffer can contain codes for graphic characters or field attributes. *Field attributes* define the start of a field and control the characteristics of the field. 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 and appears as a blank. The *field* is defined as the field attribute position plus the character positions up to, but not including, the next field attribute in the character buffer.

If the application program uses field attributes to define fields on the screen, then the screen is *formatted*. If there are no fields defined on the screen, then the screen is *unformatted*, and the operator uses it in free-form manner.

Figure 1-2 shows an example of the versatility of formatted fields. The solid characters represent the displayed form of characters stored in the character buffer. The squares represent buffer locations occupied by the field attribute characters (which are displayed as blanks). The dotted characters represent a field of data stored in the buffer but defined by the program as nondisplayable (not visible to the operator).



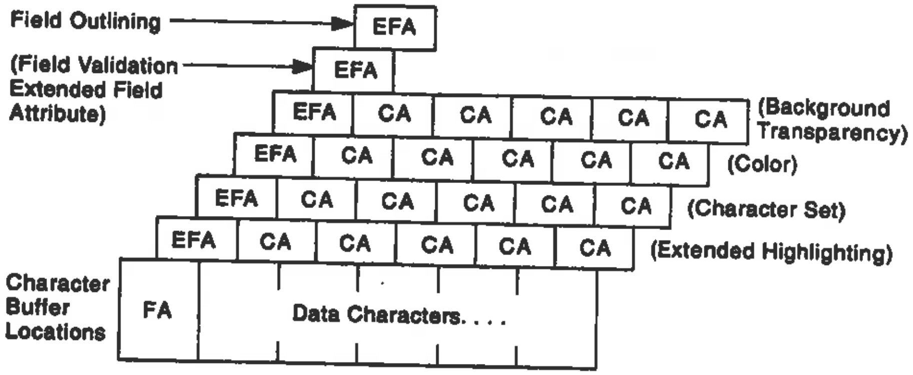
```
□NAME :□JOHN B DOE
□SALARY□1 2 5 2 3
□JOB TITLE :□WRITER
□PHONE #:□383-7628
```

Figure 1-2. Example of Formatted Fields

Attributes

The display uses three kinds of attributes: 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 displayed as a blank. The extended field attributes and character attributes do *not* occupy positions in the character buffer but control the characteristics of the field and characters, respectively. The extended field attributes are extensions of the field attribute, as shown in Figure 1-3.



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

Figure 1-3. Field, Extended Field, and Character Attributes

Field Attributes

The field attribute defines the start of a field and the characteristics of the field.

A field can wrap from the end of one row to the beginning of the next row on the screen. A field can also wrap from the last location on the screen 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.

Field attribute defines the following field characteristics:

- *Protected or unprotected.* A protected field cannot be modified by the operator. The operator can enter data or modify the contents of an unprotected field. Unprotected fields are classified as input fields.
- *Alphanumeric or numeric.* Unprotected alphanumeric fields are fields into which an operator enters data normally, using the shift keys (uppercase/lowercase or numeric/alphabetic) as required.

  Fields defined as numeric accept all uppercase symbols and numerics from a data entry keyboard. On a typewriter keyboard, numeric has no meaning and all entries are accepted.

- *Autoskip.* The cursor skips over fields that are defined as protected and numeric.
- *Nondisplay or display/intensified display.* The selected characteristics apply to the entire field. *Nondisplay* means that any characters entered from the keyboard are placed in 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 cannot intensify characters on the screen and highlight characters 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.

Extended Field Attributes

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.

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.

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 is associated with 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 set of 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 flashes on and off. Reverse Video—In each character cell affected, the on/off illumination state for every display point is reversed. The effect is analogous to white on black becoming black on white.

Table 1-1 (Page 2 of 2). Attribute Pairs

Attribute Type	Attribute Value
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 (PS) 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 Logical Channel Identifier (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.
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 are set as indicated. Underlying presentation space cannot be viewed.

Commands

Commands define the function to be performed by the display. They 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.

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

The commands and the data that can follow the command are:

Command	Data
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.

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.

Structured Fields

Structured fields are used to send and receive additional control functions and data to or from the terminal. There are three types of structured fields: outbound, outbound/inbound, and inbound. Outbound structured fields are introduced with the Write Structured Field (WSF) command. Inbound structured fields are preceded by an Attention Identifier (AID) X'88'.

The outbound and outbound/inbound structured fields are defined in Chapter 5, "Outbound/Inbound and Outbound Structured Fields." The inbound structured field functions that can be sent by the display are defined in Chapter 6, "Inbound Structured Fields."

Chapter 2. Partitions

Introduction

This chapter describes how the data stream allows the application program to divide the display surface into several areas, one of which is an active area. Each area is associated with a *partition*, and the partition that is active is the one that the operator is using to enter data or requests.

If the display supports partitioning, the application program can define a logical area called a *partition* that can differ in both size and shape from the physical display screen. The partition is defined with the Create Partition structured field. (Structured fields are discussed in Chapter 5, "Outbound/Inbound and 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.

Multiple partitions can be defined for the display that divide the display screen into several rectangular areas called *viewports*, where data from multiple partitions can be displayed on the same physical display screen. 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. (See Figure 2-1 on page 2-3.)

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.

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 an *implicit partition.*

Presentation Space, Partitions, Windows, and Viewports

The mapping of the presentation space to the physical screen is transparent to the application program.

The application program defines the amount of storage in the character buffer that is used by the partition. The application program and the operator can interact with the data in this buffer. The character buffer provides storage for characters to be shown 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.

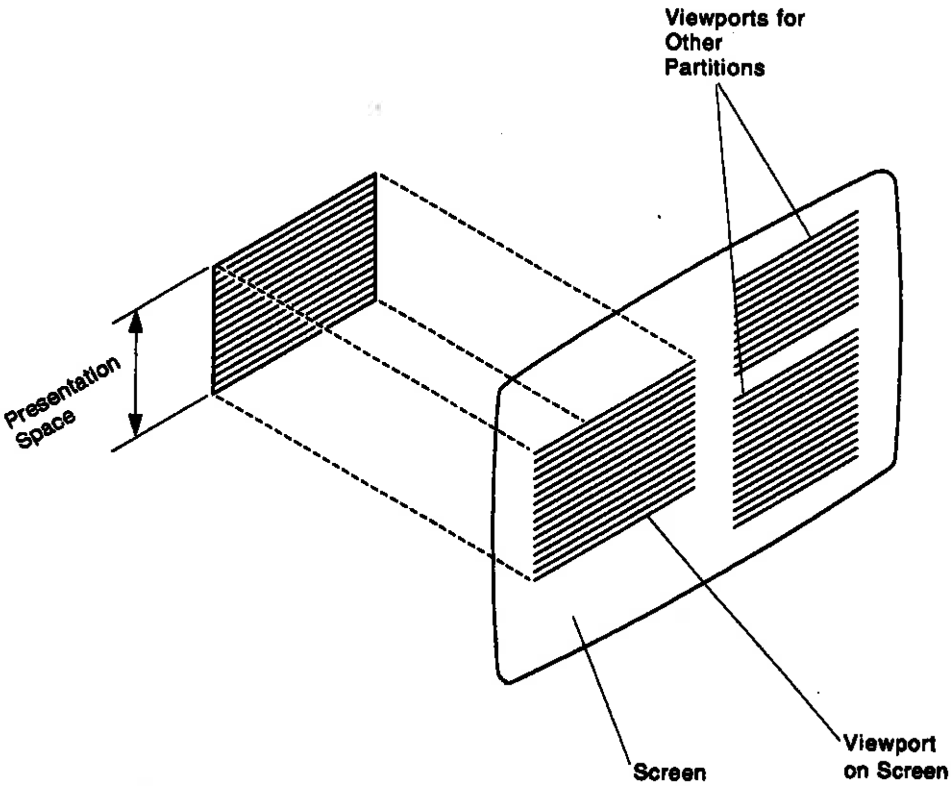


Figure 2-1. Presentation Space and Viewport (without Scrolling)

Conceptually, however, you could think of a partition as a two-dimensional presentation space, the size of which 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.

Each partition is associated with a conceptual two-dimensional surface, called the *presentation space*. You could think of data 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 2-4.) A *window* on the presentation space identifies that part of the presentation space available for viewing on the display surface.

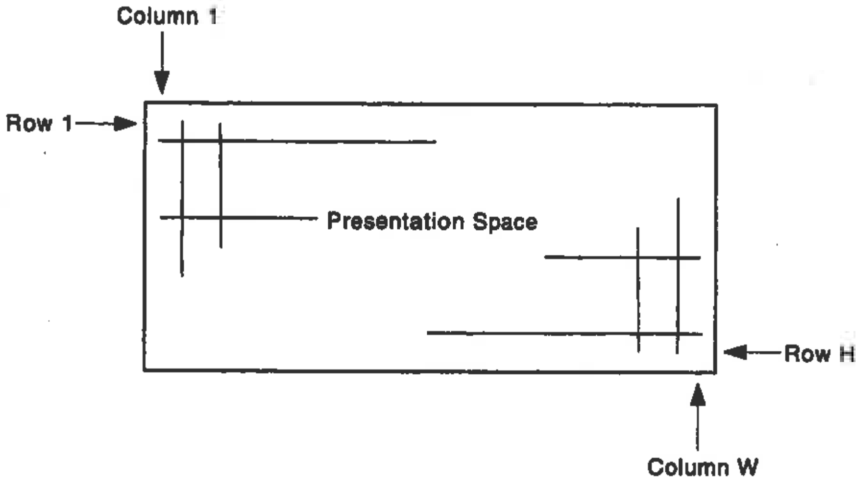


Figure 2-2. The Presentation Plane—A Conceptual View

The viewport is that area on the display surface where the 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 the 3270 data stream, 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) with the 3270 Set Buffer Address order. Orders are discussed in Chapter 4, "Character Sets, 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 on page 2-3.) Assuming it is not in a nondisplay field, each data character in the presentation space appears 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 2-5.) 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 appears 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 *scrolling*.

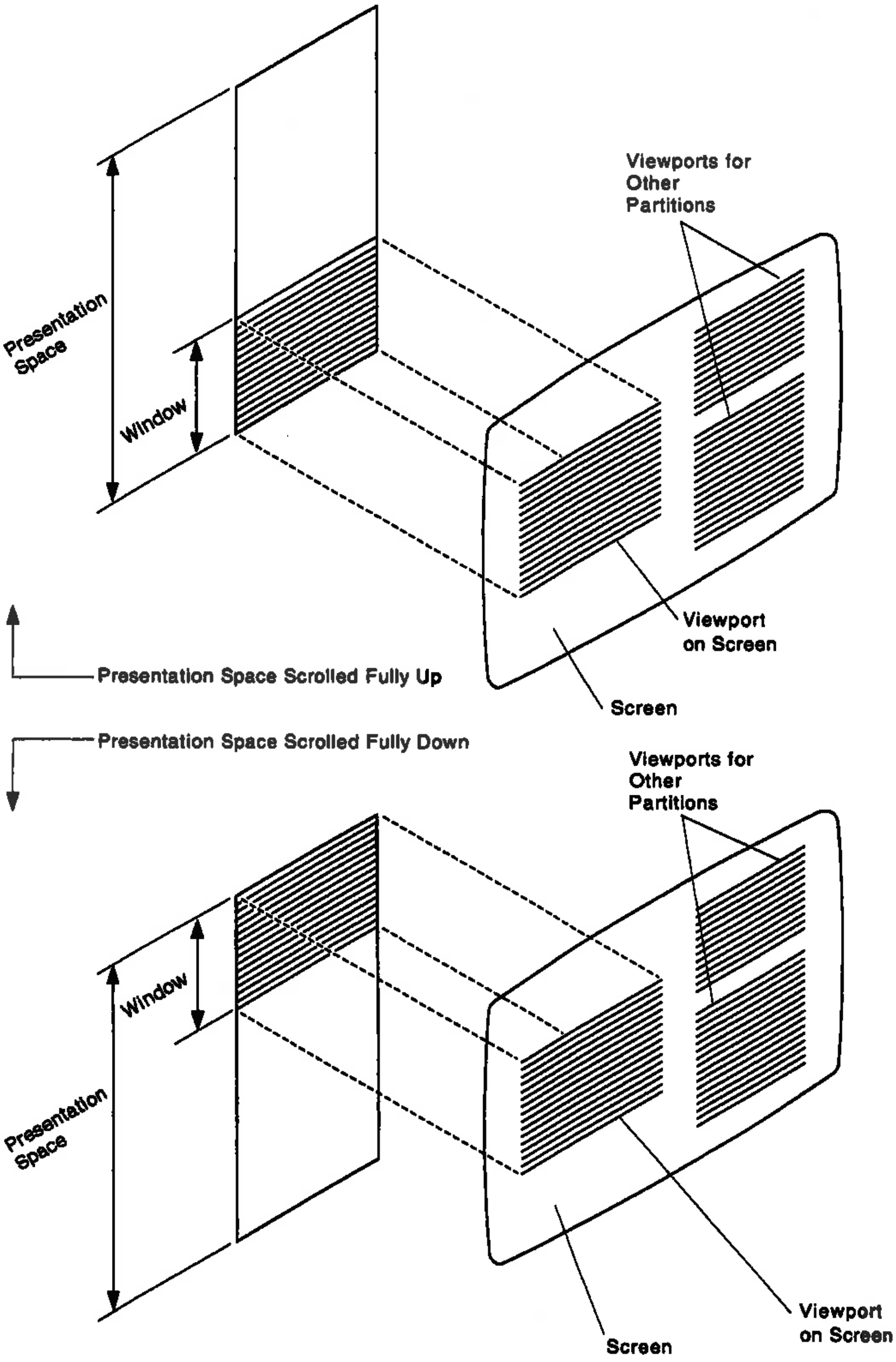

Presentation Space
Window
Viewports for Other Partitions
Viewport on Screen
Screen
Presentation Space Scrolled Fully Up
Presentation Space Scrolled Fully Down
Window
Presentation Space
Viewports for Other Partitions
Screen
Viewport on Screen

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

Scrolling

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

- Using the keys for scrolling
- Using the application program.

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 the range of cursor movement on the screen is always constrained by the bounds of a viewport. The cursor can be moved out of a given viewport only into another viewport; this can be done, for example, by using the Jump Partition key or by 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 the *active partition.* The operator can enter data only in the active partition.

Each partition has a current cursor position 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.

Implicit and Explicit Partitions

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 book 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/Inbound and Outbound Structured Fields."

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 can 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.

The transmission of data from a partition can be initiated either by the application program or by the display operator in the same way 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).

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. It is also controlled by the Erase/Reset structured field.

EW redefines implicit partition 0 with the default screen size. EWA redefines implicit partition 0 with the alternate screen size. The Erase/Reset structured field allows you to specify either the default screen size or the alternate 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 1920 characters (24 x 80) 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 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 in 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.

Using 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 can be active at a time. Each partition is identified by a PID.

The operator can move the screen cursor 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 can activate any partition by using the Activate Partition structured field and can deactivate any partition by activating another partition. The host application program can destroy any partition by using the Destroy Partition structured field. These structured fields are described in Chapter 5, "Outbound/Inbound and Outbound Structured Fields."

Managing the Presentation Space

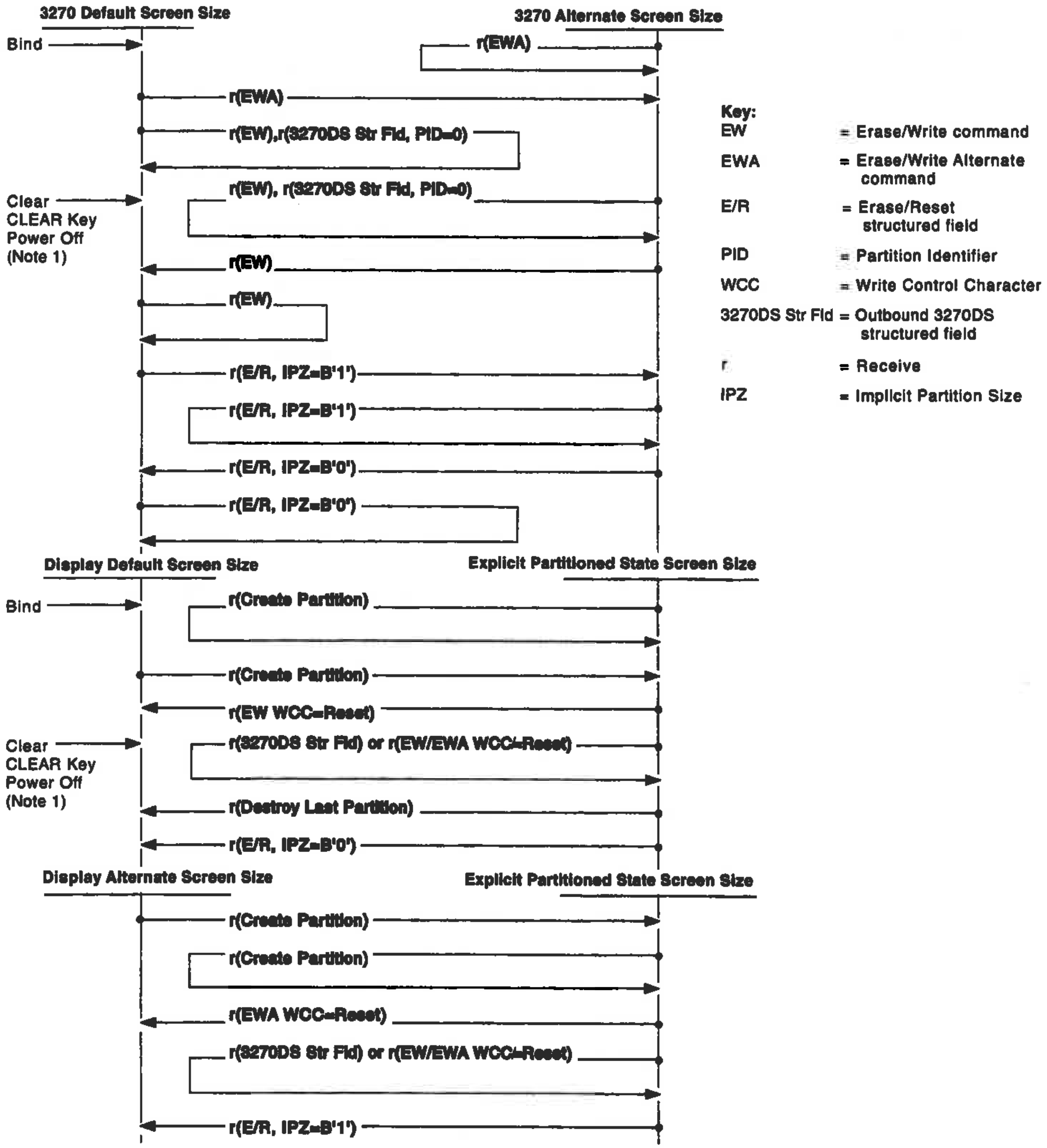
The presentation space controls the value at which buffer addresses wrap. For those devices that permit scrolling, the presentation space can be larger than the viewport. For devices that do not permit scrolling, the presentation space is the same size as the viewport.

Figure 2-4 on page 2-9 summarizes the management of presentation spaces. The commands, reset function, and structured fields shown in this figure are described in detail in this book.

When in implicit partition state, EW establishes a default screen size and EWA establishes an alternate screen size. The Erase/Reset structured field resets the device to implicit partition state, destroying all existing implicit or explicit partitions. It creates an implicit partition 0 with default partition characteristics and a default size if IPZ=B'0' or an alternate size if IPZ=B'1'.

When the Clear key is pressed, the device may either be set to the default size or remain the same, depending on the customization value selected or the option that has been designed into the product.

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.


3270 Default Screen Size
3270 Alternate Screen Size
Bind
r(EWA)
r(EWA)
r(EW),r(3270DS Str Fld, PID=0)
r(EW), r(3270DS Str Fld, PID=0)
Clear
CLEAR Key
Power Off
(Note 1)
r(EW)
r(EW)
r(E/R, IPZ=B'1')
r(E/R, IPZ=B'1')
r(E/R, IPZ=B'0')
r(E/R, IPZ=B'0')
Display Default Screen Size
Explicit Partitioned State Screen Size
Bind
r(Create Partition)
r(Create Partition)
r(EW WCC=Reset)
Clear
CLEAR Key
Power Off
(Note 1)
r(3270DS Str Fld) or r(EW/EWA WCC≠Reset)
r(Destroy Last Partition)
r(E/R, IPZ=B'0')
Display Alternate Screen Size
Explicit Partitioned State Screen Size
r(Create Partition)
r(Create Partition)
r(EWA WCC=Reset)
r(3270DS Str Fld) or r(EW/EWA WCC≠Reset)
r(E/R, IPZ=B'1')

Key:

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

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.

Figure 2-4. Management of Presentation Spaces

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:

- The host resets PWAIT 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 is removed from the screen, but the PWAIT condition is remembered, and when the original partition becomes active again the PWAIT indicator reappears on the screen.
- The PWAIT condition can only be applied against INPID.

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:

- The host or the operator can reset System Lock.
- 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 is removed from the screen, but the System Lock condition is remembered, and when the original partition becomes active again the System Lock indicator reappears on the screen.
- At any one time, several partitions can have a system lock condition.
- Any operator enter action, except a trigger action, activates System Lock with PWAIT. If the display is in Contention state, Bid or begin bracket (BB) activates 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.

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 BB 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

This chapter describes the commands and how they function in the 3270 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 write control character (WCC) followed by orders and data. If the write command is a Write Structured Field (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 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 unit (RU) chain (referred to throughout this book 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 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/Inbound and Outbound Structured Fields," and Chapter 6, "Inbound Structured Fields," for a description of the structured fields.)

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

Table 3-1 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 American National Standard Code for Information Interchange (ASCII) environment.

Write Control Character (WCC) Byte

The WCC byte 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 Write, 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 (NOP).

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 an NOP 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 3-4 for a description of each WCC bit and Table 3-3 on page 3-5 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 performed as follows:

Reset	Performed in each structured field as it is encountered.
Start print	Performed at the end of the RU chain, after the write operation has been completed. Only the last structured field can have a WCC that specified Start Print; otherwise, the chain is rejected (sense code X'1001' or sense code X'1005').
Sound alarm	Performed for each structured field, at the end of the operation specified for the structured field.
Keyboard restore	Performed for each structured field, at the end of the operation specified for the structured field.
Reset Modified Data Tag (MDT)	Performed for each structured field, prior to the writing of any data or the performing 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 C-1 on page C-2. 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, it resets partition characteristics to their system-defined defaults. When set to 0, the current characteristics remain unchanged (no reset operations are performed).
2 and 3	For printers. (See Chapter 8, "Printer Operations.")
4	Start-printer bit. When set to 1, it 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, "Printer Operations" for details of local copy operation.)
5	Sound-alarm bit. When set to 1, it 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, this bit unlocks the keyboard. It also resets the AID byte.
7	Bit 7 resets 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 performed.

Table 3-3. Write Control Character (WCC) Reset Actions for Displays

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

Write Operation

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

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

The Write, EW, and 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 EW 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 and EW are identical 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 performed by the display. The structured field identifies the specific partition by its partition identifier.

Write Command

The Write 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.

The Write command is required for Systems Application Architecture (SAA) support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "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 SBA 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

The EW command does the following:

- Sets the implicit partition size to the default size, if in implicit state.
- Resets a Program Check Indication, if one exists.
- Erases the character buffer by writing null characters into all buffer locations.
- Sets all the associated character attributes and extended field attributes to their default value (X'00').
- Erases all field validation attributes.
- 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).
- If bit 1 of the WCC is set to B'1', EW does the following:
  - Resets the inbound reply mode to Field.
  - 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.
- Provides an acknowledgment of any outstanding read or enter if the keyboard restore bit in the WCC is set to B'1'.
- Provides a negative trigger reply.
- Performs a write operation.

The EW 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 E, "Functions Required for Systems Application Architecture (SAA) Support."

Bit 1 in the WCC carries reset information used by EW. If no WCC is defined following an EW, the command is considered a NOP. 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 EW function in a named partition, other than partition 0, a WSF command must be used. An Outbound 3270DS structured field contains the PID and the partition command (EW).

Erase/Write Alternate Command

The EWA command performs the same operation as described for EW, but EWA uses the alternate implicit partition size. If there is no alternate screen size, the EWA is treated the same as EW.

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

Write Structured Field Command

WSF 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 WSF data stream is as follows:

WSF Command	Structured Field	Structured Field ...

The data stream can 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 they are 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 (W)
- Erase/Write (EW)
- Erase/Write Alternate (EWA)
- Erase All Unprotected (EAU).

W 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 Command

EAU 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.

EAU is required for SAA support. For more information about SAA, see the list of related publications at the beginning of this book and Appendix E, "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 can contain an AID byte.

When the 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 the following:

- 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 3-10. All AID bytes transmitted by the display are a result of operator actions except for the following:

- 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 can be sent as a result of some operator action, or it can 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. (Table 3-4 on page 3-10 lists the AID bytes sent from the display.)

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*, can be set to another value by operator action. (See Table 3-4 on page 3-10.) The application program can reset the AID value to *no AID generated* by sending a write command with the keyboard restore bit set on in the WCC.

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
- EAU.

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:			
Operator ID reader	E6	57	W
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 can 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 Operations

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

- 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 as follows:

AID	Cursor address (2 bytes)	Data

An inbound data stream can also consist of an AID (X'88') followed by structured fields as follows:

AID	Structured Field	......	Structured Field

Read Commands

Three read commands can 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 can initiate a read buffer, read modified, or read modified all operation, or it can 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 that 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 when the application program sends a Set Reply mode structured field. The inbound reply modes that can be set are Field, Extended Field, and Character; the default is Field mode. The three modes function as follows:

- Field mode supports inbound data that contains field attributes. SF and SBA orders, field attributes, characters, and Graphic Escape (GE) code X'08' can be included in the inbound transmission.
- Extended Field mode supports field attributes and extended field attributes. SFE and SBA orders, field attributes, extended field attribute, characters, and the GE code X'08' can be included in the inbound transmission.
- Character mode supports the field attributes, extended field attributes, characters, and character attributes. SFE, SBA, and SA orders, field attributes, extended field attributes, character attributes, and the GE code (X'08') can be included in the inbound transmission.

Read Buffer Command

Operation 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.

The Read Buffer 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 E, "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 significantly increases the 3270 subsystem and teleprocessing network response times. The Read Buffer command is intended primarily for diagnostic purposes.

Read Buffer Field Mode

In read buffer field mode, no extended field or character attribute information is returned. Only SF orders occur in the data stream. The format of the Read Buffer Field data stream is as follows:

AID	Cursor Address (2 bytes)	SF Order	Attribute	Data

Read Buffer Extended Field Mode

In read buffer extended field mode, SFE orders are generated in place of SF orders, and no character attributes are transmitted inbound. The form of the Read Buffer Extended Field data stream is as follows:

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 read buffer character mode, the inbound data stream is as defined for extended field mode above, except that SA orders can be inserted. 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 order generated is for the first attribute type not equal to its default value. However, SA orders are not generated for attribute types not listed in the Set Reply Mode structured field. The format of the Read Buffer Character data stream is as follows:

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 either a read modified or a short read operation, depending on the operator actions at the display station.

The Read Modified 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 E, "Functions Required for Systems Application Architecture (SAA) Support."

Read-Modified Operation

During a read modified operation, if an AID other than selector pen attention, cursor select key, 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 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.

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 reads 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.

For SNA, the buffer location at which the search for attribute locations begins is location 0. For non-SNA, the buffer location is determined as follows:

- If the Read Modified command is unchained or is chained from a Copy, Select, Sense, or No Operation command, the search begins at buffer address 0.
- If the Read Modified command is chained from a Write, Erase/Write, Read Modified All, or Read Buffer command, the search begins at the current address.

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 (Cancel), 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 and Extended Field Modes

In read modified field and extended field modes, no character attribute information is returned. Only SBA orders occur in the data stream. The format of the Read Modified Field and Extended Field data streams is as follows:

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

Read Modified Character Mode

This data stream for read modified character mode 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 is 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 form of the data stream is as follows:

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

The Read Modified All command operates exactly like the Read Modified command, except that it always causes 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').

The Read Modified All 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 E, "Functions Required for Systems Application Architecture (SAA) Support."

Read Modified All Modes

The Read Modified All Field, Extended Field, and Character Mode commands operate exactly like the Read Modified Field, Extended Field, and Character Mode commands.

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 can also be used in addition to the Read Buffer, Read Modified, and Read Modified All commands to initiate a read from implicit or explicit partition 0. The Read Partition structured field allows the application program to specify a Read Buffer, Read Modified, or Read Modified All operation. The inbound reply to a Read Partition structured field (directed to a nonzero partition) specifying a Read Buffer, Read Modified, or Read Modified All operation is 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" on page 3-19.)

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* (INOP), are defined as read operations caused by one of the following:

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

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

INOP defines the type of operations to be performed on a retry. See"Inbound Operation (INOP)" on page 3-20.

INPID is the inbound partition identifier and it defines the partition from which data is to be transmitted. See "Inbound Partition Identifier (INPID)" on page 3-21. 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 the host application acknowledges the INOP. 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 an RM command (**not** a partition Read Modified command).

The Read Modified command is the only mechanism for a complete retry of the INOP. When in a retry state, a Read Modified All or Read Buffer command results in performing the Read Modified All or Read Buffer on the INPID. 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 a Read Modified, Read Modified All or Read Buffer command when the display is in the normal read state. The Read Modified, Read Modified All or Read Buffer is performed 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 for SNA

This section describes read operations for SNA. For non-SNA read operations, 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, pressing 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 does the following:

- Sets INOP and INPID so that the operation and partition identifier are known if a retry is needed.
- Raises the enter-inhibit condition so that the operator cannot generate an AID from any partition. 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.

- Raises the PWAIT and system lock conditions if the enter action is not a trigger action (AID = X'7F'). See "Trigger" on page 4-25. To ensure data integrity for retry of operator enter actions, an input-inhibit condition, called PWAIT, prohibits the operator from working in the partition until the application program acknowledges that the transfer of data was successful.
- Places the device in retry enter state.

Application Initiated Reads

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

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

Read Partition Structured Field

A Read Partition structured field does the following:

- Sets INPID if the operation is Read Buffer, Read Modified, or Read Modified All.
- Sets INOP to Read Modified, Read Modified All, Read Buffer, Query, or Query List.
- Raises the enter-inhibit condition
- Sends the data to the application program and places the device in retry-read state.

Explicit partition 0 can 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 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 following form:

AID = X'88'	Structured Fields

where the inbound control information is encoded into structured fields as defined in Chapter 5, "Outbound/Inbound and 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 EW command, EWA 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 cannot 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 or Query List) 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 is retransmitted.

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 Read Buffer, Read Modified, or Read Modified All command in implicit partition state.

A Read Buffer, Read Modified, or Read Modified All command received for implicit or explicit partition 0 does the following:

- Transmits data from partition 0
- Leaves the device in its current read state.

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 Read Buffer or Read Modified All commands perform their respective read operations in the inbound partition. The Read Buffer and Read Modified All 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, in which the entered data has been transmitted to the application program
  - Retry read, in which the data read has been transmitted to the application program.

  While in retry state, the last INOP can be retried by using a Read Modified command. The application program must acknowledge receipt of the read data, which 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, an operator enter action or a host initiated read is accepted and processed. Table 3-5 shows the normal read state process.

Table 3-5. Normal Read State Process

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.

Notes:

[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. Table 3-6 shows the retry enter state process.

Table 3-6. Retry Enter State Process

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. Table 3-7 shows the retry read state process.

Table 3-7. Retry Read State Process

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

Table 3-8 summarizes the read state transitions.

Table 3-8. Read-State Transitions

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.

Inbound Operation (INOP)

The display records the 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, specifically, Read Buffer, Read Modified, Read Modified All, Query or Query List.
- Acknowledgment of an inbound transmission sets INOP to Read Modified.

Inbound Partition Identifier (INPID)

The display records the 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.

TWAIT/PWAIT Input-Inhibit

TWAIT is used during an 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 can 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, with the Reset Key).

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)
- Receipt of a transmission by the device (logical terminal) that leaves it in the send or contention state (for example, CD or EB).

System Lock Input Inhibit

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") is not 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 must 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, to read a host message.

Enter Inhibit Condition

Any enter action, *including a trigger action*, sets 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 because the input-inhibit conditions associated with INPID (for example, TWAIT, PWAIT or System Lock) override the enter-inhibit condition.

The operator can jump from INPID to one of the other partitions and enter data in 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 is inhibited and the appropriate indicator is 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 (Read Buffer, Read Modified, or Read Modified All as the first byte of the data stream) is processed as follows:

1. If any of the following conditions occur, the read command is rejected:

   - The display is not in RCV or contention (CONT) state.
   - The RU does not specify CD.
   - 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 the following:

   a. The command (Read Buffer, Read Modified, and Read Modified All)
   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 as defined by the following:

   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 Read Buffer, Read Modified, or Read Modified All, then data is transmitted from the inbound partition (INPID) as defined by:

      1) The INOP (Read Buffer, Read Modified, or Read Modified All)
      2) The AID (only when INOP=RM)
      3) The inbound reply mode of the inbound partition.

   c. If the command is Read Buffer or Read Modified All, and INOP specifies Query or Query List, then the Read Buffer or Read Modified All is performed on partition 0 if it exists. Otherwise, the command is rejected. Also, the enter-inhibit condition remains in effect, and the read state does not change.

   d. If the command is Read Buffer or Read Modified All, and INOP specifies Read Modified, Read Modified All, or Read Buffer, 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 Read Modified, Read Modified All, Read Buffer, Query or Query List, or by an operator enter action) is acknowledged by any transmission that, after completion, leaves the display in a send or contention state. Although the transmission can put the display in the send state, if the transmission also causes the display to send input, it is not an acknowledgment because after completing the transmission, the display is not left in a send or contention state.

In addition, the following inbound operations constitute a Read Acknowledgement or ACK:

- Query or Query List

  A Query or Query List operation is acknowledged by any outbound transmission except one with a read command. 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.

- An Operator Enter, or a Read Modified, Read Modified All, or Read Buffer Partition command

  A Read Modified, Read Modified All, or Read Buffer 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. Any of the following constitutes 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 or Reset (see note 1)
    - An EAU command.
  - If in the explicit partition state, an EW or EWA command with the WCC = reset (see note 1)
  - A WSF command followed by an Outbound 3270DS structured field to the inbound partition with either of the following (see note 2):
    - An EW, EWA or Write partition command with WCC = Keyboard Restore or Reset
    - An EAU partition command.
  - A WSF command followed by a Destroy Partition structured field to the inbound partition including explicit and implicit partition 0 (see note 2)
  - A WSF command followed by a Create Partition to the inbound partition (see note 2)
  - If in implicit partition state, a WSF command followed by a Create Partition structured field (see note 2)
  - A WSF command followed by an Erase/Reset structured field. This applies to both implicit and explicit partition state (see 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. Character Sets, Orders, and Attributes

Introduction

This chapter describes the character sets, orders, and attributes used by the 3270 data stream.

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.

Graphic code point integrity must be 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 are 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.

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 do the following:

1. Position, define, and format data being written into the character buffer
2. Erase selected unprotected data in the character buffer
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.

Table 4-1 lists the order codes.

Table 4-1. Order Codes

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	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)

The SF 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 E, "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 "Attributes" on page 4-12.) The display then stores the field attribute at the current buffer address and increments the buffer address by one.

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)

The SFE 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 following form:

Type	Value
X'C0'	Field Attribute

Other field properties, specifically those contained in the extended field attribute, can 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" on page 4-18.

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, the following SFE control sequence produces a field with red characters that have normal highlighting and the default character set:

SFE	2	3270	Value	Color	Red

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 are 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)

The SBA 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 E, "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, SBA 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 the following:

- The starting address for a PT, RA, or EUA order
- The address at which a field attribute is to be stored by an SFE or MF order
- 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 it finds these field attributes, 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 can 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 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.

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 always generates 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 as follows:

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 one of the following occurs:

- A new SA order changes it.
- Another write type command is sent.
- The 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 Table 4-6 on page 4-17). All type-value pairs are defined under "Attribute Values and Selection Rules" on page 4-18.

An EW or EWA 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 using 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. Table 4-2 provides an example of attribute type-value pairs.

Table 4-2. Example of Attribute Type-Value Pairs

	Current Character Attribute Characteristics		
Command or Order	**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 are generated with the assumption that the character attributes of the current set are all set to their default values. SA orders are generated as required to indicate changes. SA orders are 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 are 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 one.

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, suppose that 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 following 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.

If no attributes are specified (the number of type-value pairs field is 0) then the MF order determines if there is a field attribute at the current buffer address. If so, the current buffer address is incremented by one, 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)

The IC 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 E, "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)

The PT 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 E, "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)

The RA 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 E, "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 filled with nulls.

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 GE sequences or 2-byte coded character sets, the Character-to-Be-Repeated field can be two bytes long. The format of the RA order is as follows:

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)" on page 4-5 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 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 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)

The EUA 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 E, "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 two bytes immediately following the EUA order. The format of the EUA order is as follows:

X'12'	Stop Address (2 bytes)

The 12/14-bit and 16-bit addressing considerations discussed under "Set Buffer Address (SBA)" on page 4-5 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' is 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. The current buffer address after successful execution of EUA is the stop address.

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

Graphic Escape (GE)

The GE 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 as follows:

X'08'	Character Code

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

- 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.
- GE+code points of X'00' through X'3F' and X'FF' are undefined and are rejected (sense code X'1003').
- GE+code points of X'40' through X'FE' are valid graphic character codes.
- 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 in Table 4-3.

Table 4-3. Format Control Orders

Abbrev.	Order	EBCDIC	ASCII	Displayed as
NUL	Null	X'00'	X'00'	A blank, suppressed on Read Modified
SUB	Substitute	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 blank
CR	Carriage Return	X'0D'	X'0D'	A blank
NL	New Line	X'15'	X'0A'	A blank
EM	End of Medium	X'19'	X'19'	A blank
EO	Eight Ones	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.

FM and DUP can 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.

The SUB local function, Error Override, entered from the keyboard, is required only as a part of Field Validation.

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.

The following 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 the 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 the data stream. (The control sequences used to send this information are defined under "Orders" on page 4-3.) No matter which control sequence is used, the attribute's properties are sent as a type and value pair as follows:

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 Table 4-4 on page 4-14 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 4-4. 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 C-1 on page C-2.
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	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 EOF (EOF) key, the display sets the MDT (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 the MDT bit off in all unprotected fields.
- When the EOF key is pressed, it sets the MDT bit on 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 C-1 on page C-2, in attributes. This is for compatibility with previously used application programs.

Extended Field Attributes

Extended field attributes specify the following field properties:

- 3270 field attributes (see "Field Attributes" on page 4-13)
- 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 it is 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 it is normally highlighted in an alphanumeric partition. Refer to "Query Reply (Alphanumeric Partitions)" on page 6-23 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.

The attribute's color, highlighting, and character set do not apply to the field attribute position, whereas background transparency does apply.

Character Attributes

Character attributes specify the following character properties:

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

The highlighting, color, and character set functions are identical to the functions described under "Extended Field Attributes" on page 4-15, 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 under "Conflict Resolution Between Attributes"

Conflict Resolution Between Attributes

Extended field attributes can 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.

However, when nulls are converted to spaces, the associated character attribute for each resulting space should be set to:

1. The default character attribute if Entry Assist is in effect.
2. The character attributes of the nulls being converted if Null/Space Processing is in effect. This takes precedence over rule 1.

Table 4-5 on page 4-17 summarizes the rules for resolving attribute conflicts.

Table 4-5. 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).

Table 4-6 lists the attribute types defined in this chapter. Attribute types not shown are reserved and are rejected (sense code X'1003').

Table 4-6. Attribute Types

		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

Note: NA = Not Applicable

Attribute Values and Selection Rules

The value bytes for all defined attribute types are shown in the following pages. 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

All character attributes reset all character attribute types that are specifiable in the SA 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' can appear only in the SA order.

3270 Field Attribute

The 3270 field attribute 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 4-4 on page 4-14.

If the 3270 field attribute specifies non-display, it overrides 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 are transmitted inbound.

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

Extended Highlighting

Extended highlighting attributes identify the highlighting property of a field or character. The field can 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

The following table lists valid settings for the highlighting value byte. All other settings are reserved.

Setting	**Highlighting Property**
X'00'	Default
X'F0'	Normal (as determined by the 3270 field attribute)
X'F1'	Blink
X'F2'	Reverse video
X'F4'	Underscore.

Color

Color attributes identify 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

The 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.

Table 4-7 lists the colors and their architected color identifications.

Table 4-7. Color Identifications

Color	**I.D.**	**Defined**
Neutral	X'F0'	(black on displays, white on printers)
Blue	X'F1'	
Red	X'F2'	
Pink	X'F3'	
Green	X'F4'	
Turquoise	X'F5'	
Yellow	X'F6'	
Neutral	X'F7'	(white on displays, black on printers)
Black	X'F8'	
Deep Blue	X'F9'	
Orange	X'FA'	
Purple	X'FB'	
Pale Green	X'FC'	
Pale Turquoise	X'FD'	
Grey	X'FE'	
White	X'FF'	

The 'color' listed above as Neutral with a color identification of X'F7' is defined as White for a display and Black for a printer.

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)" on page 6-36.

Character Set

Character set attributes identify the character set to be used by the device to display the field or character. The field can have only one character set 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 2-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 (Load PS)" on page 5-25.

Field Outlining

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

Type	Value
X'C2'	Field outlining

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.

The following are 16 kinds of field outlining available by using combinations of the 4 lines. The different combinations are described below:

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

Transparency attributes can designate that the space surrounding a character be transparent to the screen operator, or it can designate that the space be opaque:

Type	Value
X'46'	Transparency

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

Field validation attributes define the validation properties of the field. The field can have more than one property, and all are optional:

Type	Value
X'C1'	Field Validation

Valid bit settings for the field validation value byte are as follows:

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 SFE and 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 can be specified are described under "Attribute Values and Selection Rules" on page 4-18.

If no type-value pairs are transmitted, no field validation occurs. A field can 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 doing the following:

- 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 in a primed mandatory-fill field, then fill validation occurs.

Before a field is primed or after it is unprimed, the operator can 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 following occurs:

- 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 can jump to another partition. The screen cursor moves, but the current cursor position of the partition does not change. Consequently, the Jump Partition key does not cause fill validation to occur.
2. Similarly, the Jump Partition key does not affect validation. 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.
5. If mandatory fill is specified for a protected field, it has no effect on the field.
6. 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.
7. 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

Mandatory entry attributes specify that the terminal operator cannot transmit character data without first making an entry in the field. The operator can 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

Trigger attributes specify 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 can 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 can 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 IC order contained in the reply.

While the display is awaiting a reply from the host, the operator can 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 CD or EB is not a reply or 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 if one of the following occurs:

- 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 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 3-15.

In summary, an affirmative reply is as follows:

- 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.
- A W command with a WCC that specifies keyboard restore, if INPID is 0.
- A null RU chain with CD or EB.

In summary, a negative reply is as follows:

- 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 EW or EWA 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.

- A W command with a WCC that does not specify keyboard restore, or an EAU command, if INPID is 0.
- A W, EW, EWA, or EAU command, if INPID is nonzero.
- 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:

- A 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

When programming, you should be aware of the following facts about trigger fields:

- To avoid operator confusion, the application program should not move the cursor out of the trigger field on a negative reply.
- 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. Any data entry keystrokes are placed at the host-specified position.
- 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.
- Trigger action occurs if scrolling action moves the cursor out of a primed trigger field.
- 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 moves 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 are generally 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 IC order, the mandatory-entry field cursor position overrides the host-specified cursor position. In other words, after the reply from the host, the cursor is positioned in the mandatory-entry field and the mandatory-entry input-inhibit indicator is displayed.

- A change of the active partition does not cause trigger action.
- If the keystroke queue becomes full before a write is received from the host, an input-inhibit condition is raised.
- 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.
- 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.
- 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 can jump to another partition. The screen cursor moves, but the current cursor position of the partition does not change. Consequently, the Jump Partition key does not cause a trigger action. Similarly, validation is not affected by the Jump Partition key.
- If a field has both the mandatory fill and the trigger attributes, fill validation occurs before the trigger action.
- 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 combined 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 combined 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 combined 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, if a dot is defined on more than one plane, then the color displayed for that dot is an additive color, as shown in the following table (except as defined in note 2):

Overlap in	Displays/Prints as
Red and Blue	Pink
Blue and Green	Turquoise
Green and Red	Yellow
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)" on page 6-36 for more information.)

Defaults for Attributes

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 4-8) the X'00' code specifies that the properties associated with the field are to be used.

Table 4-8 shows the attribute defaults.

Table 4-8. 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)