**Systems**

**IBM 3270
Information Display System
Data Stream
Programmer's Reference**

IBM

Systems

**IBM 3270
Information Display System
Data Stream
Programmer's Reference**

IBM

## PREFACE

### Who This Book Is For

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

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

### How This Book Is Organized

This book has nine chapters and five
appendixes:

- Chapter 1, "The 3270 Data Stream:
  Overview and Concepts," introduces
  the 3270 data stream, gives an
  overview of it, and discusses 3270
  data stream concepts.

- Chapter 2, "Partitions," covers
  partitions and functions related
  to partitions, such as INPID,
  INOP, and PWAIT.

- Chapter 3, "3270 Data Stream
  Commands," describes the commands
  used in the 3270 data stream and
  their operations.

- Chapter 4, "3270 Data Stream
  Orders and Attributes," describes
  the orders and attributes used in
  the 3270 data stream and how they
  function. It also explains
  character-set properties.

- Chapter 5, "Magnetic-Reader,
  Keyboard, and Selector-Pen
  Operations," describes how
  magnetic readers work with the
  3270 data stream, the keyboard
  functions that affect data stream
  operation, and the use of the
  selector pen.

- Chapter 6, "Printer
  Considerations," covers printer
  functions with the 3270 data
  stream and the local-copy function
  for systems network architecture
  (SNA) and binary synchronous
  communications (BSC).

- Chapter 7, "Structured Fields,"
  lists the structured fields in
  alphabetic order, gives their
  syntax, and describes how they
  function.

- Chapter 8, "Binary Synchronous
  Communications (BSC) Environment,"
  discusses a BSC environment and
  describes the differences in
  operation from an SNA environment
  for the 3270 data stream.

- Chapter 9, "Non-SNA Environment
  (Locally Attached Devices -- 3272
  Version)," discusses a non-SNA
  environment of locally attached
  devices (3272 version) and
  describes the differences in
  operation from an SNA or BSC
  environment.

- Appendix A, "SNA Sense Codes,"
  summarizes the sense codes
  returned for data-stream errors.

- Appendix B, "SNA Sense Codes for Structured Fields," summarizes the sense codes returned for structured-field errors.

- Appendix C, "Reset Actions," summarizes the reset actions for the 3270 data stream.

- Appendix D, "12-, 14-, and 16-bit Addressing," explains the addressing used in the 3270 data stream.

- Appendix E, "Glossary of Terms and Abbreviations," defines terms and abbreviations used in this publication.

Other Books You May Need

The following publications provide a general introduction to the 3270:

- An Introduction to the IBM 3270 Information Display System, GA27-2739

- IBM 3270 Information Display System Library User's Guide, GA23-0058

- 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 3270 Information Display System
3278 Display Station Description, GA23-0063

- IBM 3270 Information Display System
3279 Color Display Station Description, GA33-3089

- IBM Systems Reference Library General Information - Binary Synchronous Communications, GA27-3004

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

iv

CONTENTS

## FIGURES

# CHAPTER 1. THE 3270 DATA STREAM: OVERVIEW AND CONCEPTS

## INTRODUCTION

This chapter introduces the 3270 data stream, gives an overview of what it is and what it does, and discusses data stream concepts.

Not all 3270 data stream functions are implemented by all products using the 3270 data stream. You should therefore consult your product library for the 3270 data stream functions used by your products.

Chapters 1 through 6 describe the 3270 data stream in a systems network architecture (SNA) environment: Chapters 1 through 5 discuss the 3270 data stream for displays in the SNA environment; Chapter 6 covers printer applications. Chapters 8 and 9 describe the differences for binary synchronous communications (BSC) and non-SNA locally-attached-device environments, respectively.

Note: Some environments may operate in a mixed mode, for example, BSC and synchronous data link control (SDLC); refer to the appropriate chapters to determine probable restrictions on using the 3270 data stream in this type of environment.

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

```
+----+----+----+-----------------+----+
|LH  | TH | RH |  3270 Data Stream | LT |
+----+----+----+-----------------+----+
```

The LH, TH, RH, and LT are the link header, transmission header, request/response header, and link trailer, respectively.

## OVERVIEW

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

| 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

| 0 |
|---|
| 1 |
| 2 |
| . |
| . |
| . |
| . |
| 959 |

960-Byte
Character Buffer

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

The data stream allows the application program to divide the display
surface into one active area and, optionally, one or more reference
areas.  Each area is called a partition, and the partition that is
"active" is the one the operator is using to enter data or requests.

The application program uses partitions by first creating one or more
partitions and then writing data to a specified partition.  If the
application programmer chooses not to create partitions, then the entire
display surface is the active area (and it is considered one partition).

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

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


DATA STREAM FORMAT


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


Outbound Data Stream

| Command | WCC | Data |
|---------|-----|------|

1-2

The write control character (WCC) is used with write commands only.

An inbound data stream is sent from the device to the application program and consists of an attention identifier (AID) followed by orders and data.

Inbound Data Stream

| AID | Cursor Address (2 bytes) | Data |
|-----|--------------------------|------|

In many cases the data is optional.

ATTENTION IDENTIFIERS (AIDS)

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

3270 DATA STREAM COMMANDS

Commands control such things as whether the application program writes to or reads from a display, and whether the screen is erased before new data is written.  An application program sends data (and control characters) to the display by means of commands.  The commands and data used by the application program and display are as follows:

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

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

## 3270 DATA STREAM ORDERS

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

- Set Buffer Address (SBA)

- Start Field (SF)

- Start Field Extended (SFE)

- Modify Field (MF)

- Set Attribute (SA)

- Insert Cursor (IC)

- Program Tab (PT)

- Repeat to Address (RA)

- Erase Unprotected to Address (EUA)

- Graphic Escape (GE)

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

- Set Buffer Address

- Start Field

- Start Field Extended

- Set Attribute

- Graphic Escape

## 3270 DATA STREAM ATTRIBUTES

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

## DATA

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

STRUCTURED FIELDS IN THE 3270 DATA STREAM

Structured fields are used to convey additional control functions and
data to or from the display.  The Write Structured Field (WSF) command
is used to transmit structured fields outbound.

The structured-field functions that can be accepted by a display are:

* Create Partition

* Activate Partition

* Destroy Partition

* Set Window Origin

* Load Programmed Symbols

* Set Reply Mode

* Reset Partition

* Outbound 3270DS

    - Write

    - Erase/Write

    - Erase/Write Alternate

    - Erase All Unprotected

* Read Partition

    - Read Modified

    - Read Modified All

    - Read Buffer

    - Query

The display uses an AID (X'88') to indicate inbound structured fields.
The structured field functions that can be sent by the display are:

* Query Reply

* Inbound 3270DS

CONCEPTS

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

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

UNFORMATTED AND FORMATTED SCREENS

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

•   A formatted screen (fields defined by field attributes) is organized into fields by the application program.

•   An unformatted screen (no fields defined) is regarded as one with a storage/display area that the operator uses in free-form manner.

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

[]NAME :[] JOHN B DOE
[]SALARY []1 2 8 2 3
[]JOB TITLE :[]WRITER
[]PHONE #:[]383-7628

Figure 1-2. Example of Formatted Fields

KINDS OF ATTRIBUTES


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

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

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


Field Attributes


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

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



CA = Character attribute
EFA = Extended field attribute
FA = Field attribute
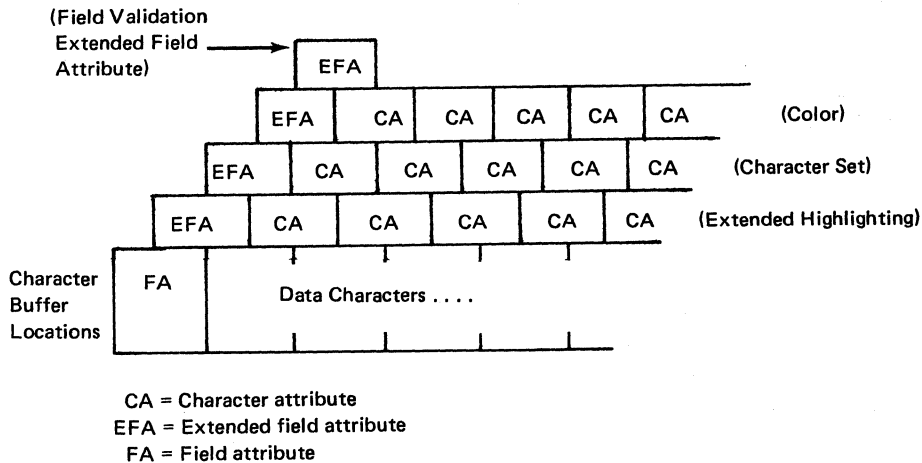
Figure 1-3. Character and Extended Field Attributes – A Conceptual
View

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

To define the following characteristics for the entire field:

<u>Protected</u> <u>or</u> <u>Unprotected</u>. A protected field is protected from modification by the operator. An unprotected field is available for the operator to enter or modify data. The unprotected definition classifies a field as an input field.

<u>Alphanumeric</u> <u>or</u> <u>Numeric</u>. Subject to its being unprotected, an alphanumeric field is one into which an operator enters data normally, using the shift keys (uppercase/lowercase or numeric/alphabetic) as required.

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

<u>Autoskip</u>. A field defined as protected <u>and</u> numeric causes the cursor to skip a subsequent field.

<u>Nondisplay</u> <u>or</u> <u>Display/Intensified</u> <u>Display</u>. The selected characteristics apply to the entire field. <u>Nondisplay</u> means that any characters entered from the keyboard are entered into the buffer for possible subsequent transmission to the application program but they are not displayed. <u>Intensified</u> <u>display</u> means the intensified characters appear on the screen brighter than the nonintensified characters.

<u>Detectable</u> <u>or</u> <u>Nondetectable</u>. A field defined as detectable can be detected by the selector pen or the cursor select key, subject to the use of a designator character

Base color (4 colors) can be produced on color displays and color printers from current 3270 application programs by use of combinations of the field intensify and field protection attribute bits. For more information on color, refer to the <u>IBM</u> <u>3270</u> <u>Information</u> <u>System:</u> <u>Color</u> <u>and</u> <u>Programmed</u> <u>Symbols</u> manual, GA33-3056.


## Extended Field Attributes


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

The attribute structure used for extended field attributes defines all characteristics with attribute type-value pairs, as shown in Figure 1-4. Each attribute type has associated with it a set of attribute values.

```
Attribute Pair --  | Attribute Type | Attribute Value |
```

Attribute Type                Attribute Value

Extended Highlighting         Default.  If used as a character attribute, the
                              default is the characteristics defined by the
                              extended field attribute.  If used as an
                              extended field attribute, the default becomes
                              those characteristics indicated by the Query
                              Reply (Highlight) structured field.

                              Underscore.  Each character in the field is
                              underlined.

                              Blink.  Each character of field affected is
                              caused to flash on and off.

                              Reverse video.  In each character cell
                              affected, the on/off illumination state of
                              every display point is reversed.  The effect is
                              analogous to white on black becoming black on
                              white.

Color                         Default.  If used as a character attribute it
                              assumes the characteristics of the extended
                              field attribute.  If used as an extended field
                              attribute, it is as indicated by the Query
                              Reply (Color) structured field.

                              Multicolor.  Indicates that the color is
                              defined by a triple-plane Programmed Symbol
                              set.

                              All others.  Assigned to the color
                              identifications as indicated by the Query Reply
                              (Color) structured field.

Character set                 Default.  If used as a character attribute, it
                              assumes the characteristics of the extended
                              field attribute.  If used as an extended field
                              attribute, it is the nonloadable character set
                              that has the LCID of X'00' in the Query Reply
                              (character sets) structured field.

                              Local character set ID.  For the loadable
                              character set.

**Figure 1-4 (Part 1 of 2).  Attribute Pairs**

| Attribute Type | Attribute Value |
|---|---|
| | Local character set ID. For the nonloadable character set. |
| Field validation | Mandatory field. A field that must be modified by the operator before the operator can transmit any data from the display. |
| | Mandatory-fill field. 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 field. 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. |

**Figure 1-4 (Part 2 of 2). Attribute Pairs**

## 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; characters in nondisplay fields, however, are never displayed. The attribute structure used for character attributes is the same as for extended field attributes. (See Figure 1-4.)

## THE CURSOR

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

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

Normally, a character entered at the keyboard is stored in the display buffer (and is displayed) at the current cursor position -- after which the cursor advances one position and is ready for the next character to be entered. For this to occur, the cursor must be positioned in an unprotected field. If a character already exists at the current cursor position, then (except in insert mode) that character is overwritten by the entered character.

PARTITIONING


Partitioning allows the application program to define a presentation
space that may be different in both size and shape from the physical
display surface.  Multiple partitions can be defined for the display
which will allow the display surface to be divided into several
rectangular areas called viewports, where data from multiple partitions
may be displayed on the same physical display surface.  The viewport is
the area used to display and enter data for that partition.(See Figure
1-5.)

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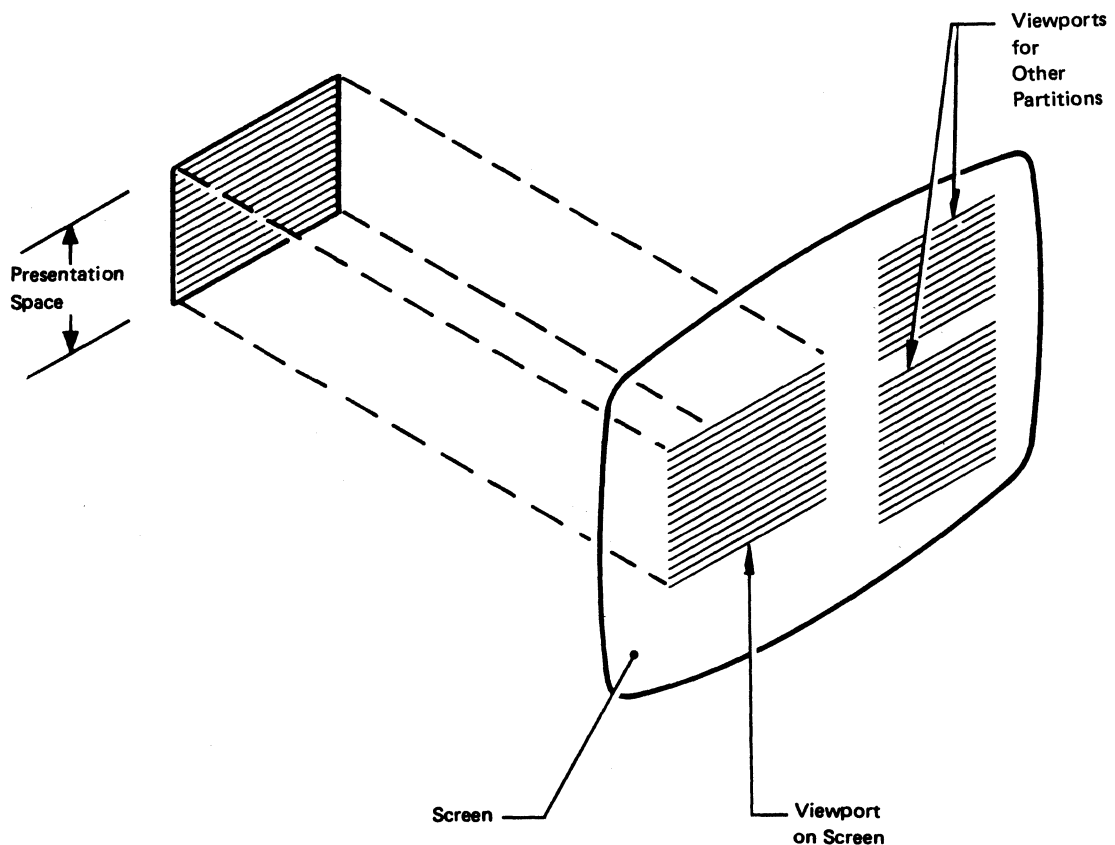


Figure 1-5. Presentation Space and Viewport (without Scrolling)

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

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

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


READ FUNCTIONS FOR A PARTITION


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

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


EXPLICIT PARTITIONED AND IMPLICIT PARTITIONED STATES


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

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

The display may be returned to implicit partitioned 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.

# CHAPTER 2. PARTITIONS


## CREATING A PARTITION


The application program can define a "logical" screen, called a
partition, which may differ in both size and shape from the physical
display screen.  The partition is defined by use of the Create Partition
structured field.  (Structured fields are discussed in Chapter 7.)  Once
a partition has been created, data is transmitted to and from the
partition as if it were a physical screen with the geometrical
characteristics specified in the Create Partition structured field.  The
mapping from the application program's view of the device to the
physical screen is transparent to the application program, once the
partition has been successfully created.


## CHARACTER BUFFER AND CONCEPT OF PRESENTATION SPACE


The amount of storage, in the character buffer, that is used by the
partition is defined by the application program.  Data in this buffer
may be interacted with by the application program and by the operator.
The character buffer provides storage for characters that may be
displayed on the display surface.  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.

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


## PRESENTATION SPACES, WINDOWS, AND VIEWPORTS


A partition has associated with it a conceptual two-dimensional surface,
called the presentation space.  Data may be thought of as being
presented on this two-dimensional surface although the surface does not
exist physically as such on the device.  (See Figure 2-1.)  A window on
the presentation space identifies that part of the presentation space
available for viewing on the display surface.
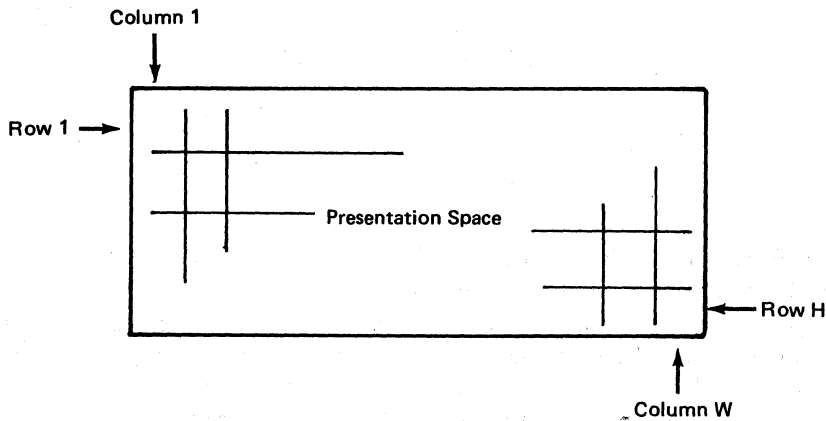
**Figure 2-1. The Presentation Space–A Conceptual View**

On the display surface, there is a rectangular area called a <u>viewport</u>. The viewport is that area on the display surface where the terminal operator sees the partition data displayed. Each viewport is related to a window so that the data in the presentation space within the window appears on the screen within the viewport.

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

RELATIONSHIP BETWEEN PRESENTATION SPACE AND VIEWPORT

When the scrolling display function is not in use, a partition's presentation space and viewport have the same dimensions. (See Figure 1-5.) Assuming it is not in a "nondisplay" field, each data character in the presentation space is displayed in the corresponding row and column of the viewport.

When the scrolling display function is in use, the viewport displays the data from a subarea of the presentation space called the <u>window</u>. (See Figure 2-2.) The window and the viewport have the same dimensions (rows and columns). Assuming it is not in a nondisplay field, each data character in the window is displayed in the corresponding row and column of the viewport. The position of the presentation space data in relation to the window and viewport can be altered by an action known as <u>scrolling</u>.
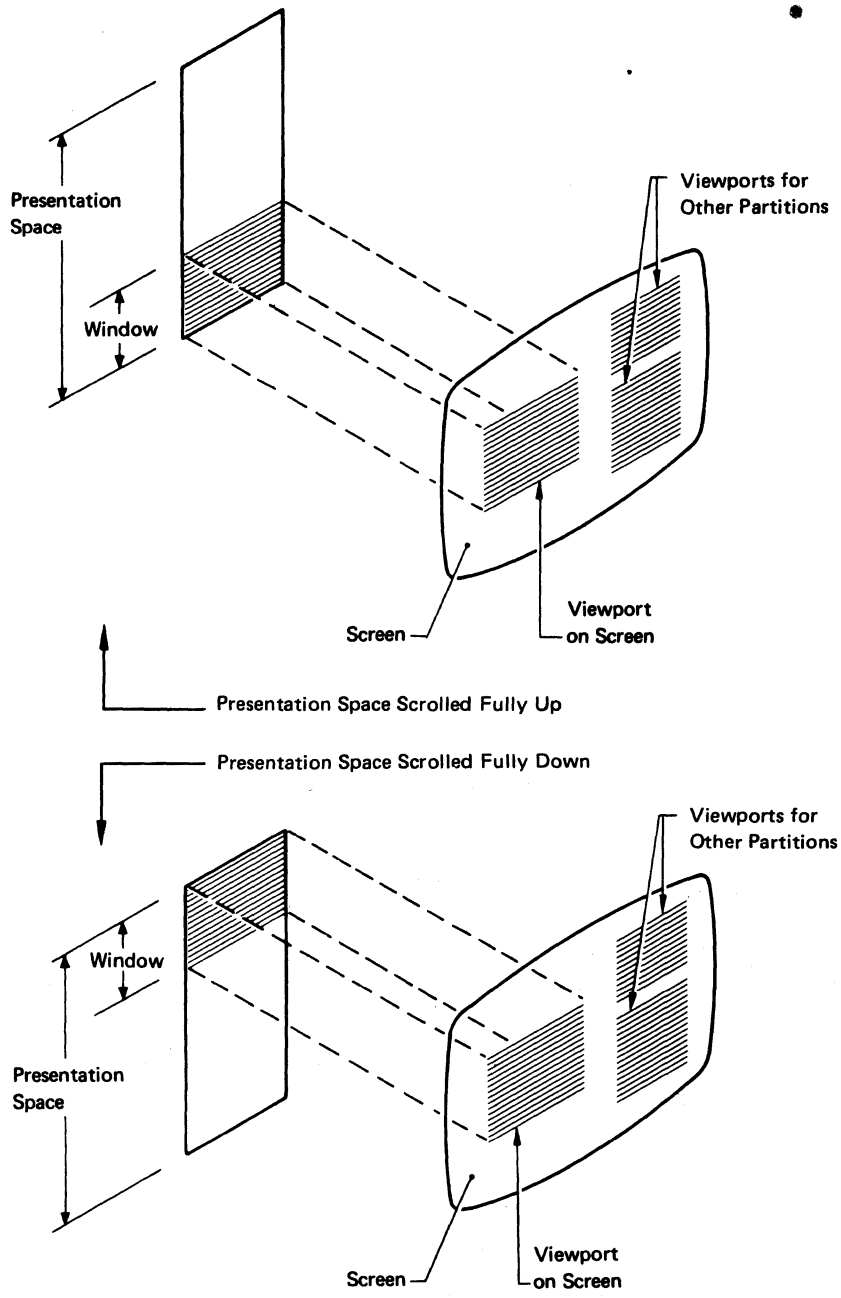
Figure 2-2. Presentation Space, Window, and Viewport (with Scrolling)

SCROLLING

In a scrollable partition, the presentation space data can be moved up or down relative to the window by:

* The operator using the SCROLL keys

* The application program

With this function, the application program positions the window by specifying the number of rows (see "Set Window Origin" in Chapter 7) by which the top of the window is to be offset from the top of the presentation space.


THE CURSOR IN PARTITIONS


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

Associated with a partition is a current cursor position, which determines where alphanumeric data is placed in the presentation space during operator keystroking. The cursor is displayed at the current cursor position of the partition. Data entry or cursor movement causes the current cursor position of the partition to be changed.


MULTIPLE PARTITIONS


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

The operator may cause the screen cursor to jump from one partition to the next by pressing the jump key. Pressing the jump 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.)

The application program may activate any partition by using the Activate Partition structured field, and may deactivate any partition by activating another partition.  (These structured fields are described in Chapter 7.)


## THE IMPLICIT PARTITION


When the display is powered on, it is placed in implicit partitioned state.  A single implicitly defined partition is created automatically and assigned a PID of 0 with the default screen size.  In implicit partitioned state, the size of the implicit partition is controlled by the Erase/Write (EW) and Erase/Write Alternate (EWA) commands in the data stream.

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

The characteristics of the implicit partition are as follows:

* Partition parameters expressed in row/column coordinate system

* Partition size = Screen size

* Window size = Partition size

* Viewport size = Window size

* Viewport origin = Screen origin

* No scrolling permitted

* Unprotected (operator interaction allowed)


## EXPLICIT PARTITIONS


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

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

partition 0 operates in implicit partitioned state, while explicit partition 0 operates in explicit partitioned state.

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

## STATES AND STATE TRANSITIONS

Displays using the 3270 data stream can operate in one of two states. When operating without partitions (operating with implicit partition 0), the device is said to be in implicit partitioned state. When partitions have been created explicitly, with the Create Partition structured field, the device is said to be in the explicit partitioned state. In each state, all the orders and commands described in this manual are valid. The distinction between the two states relates to the way in which the usable area is managed. In implicit partitioned 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 7.

The presentation space (character buffer) controls the value at which buffer addresses wrap. In general, the presentation space may be equal to, or greater than, the viewport on the screen, with the window and the viewport always equal in horizontal and vertical dimensions. In practice, for those devices that permit scrolling, the presentation space may be larger than the viewport. For devices that do not permit scrolling, the presentation space is of the same size as the viewport.

The management of presentation spaces is summarized in Figure 2-3. The commands, reset function, and structured fields shown in this figure are described in detail in the manual.

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

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

2-6
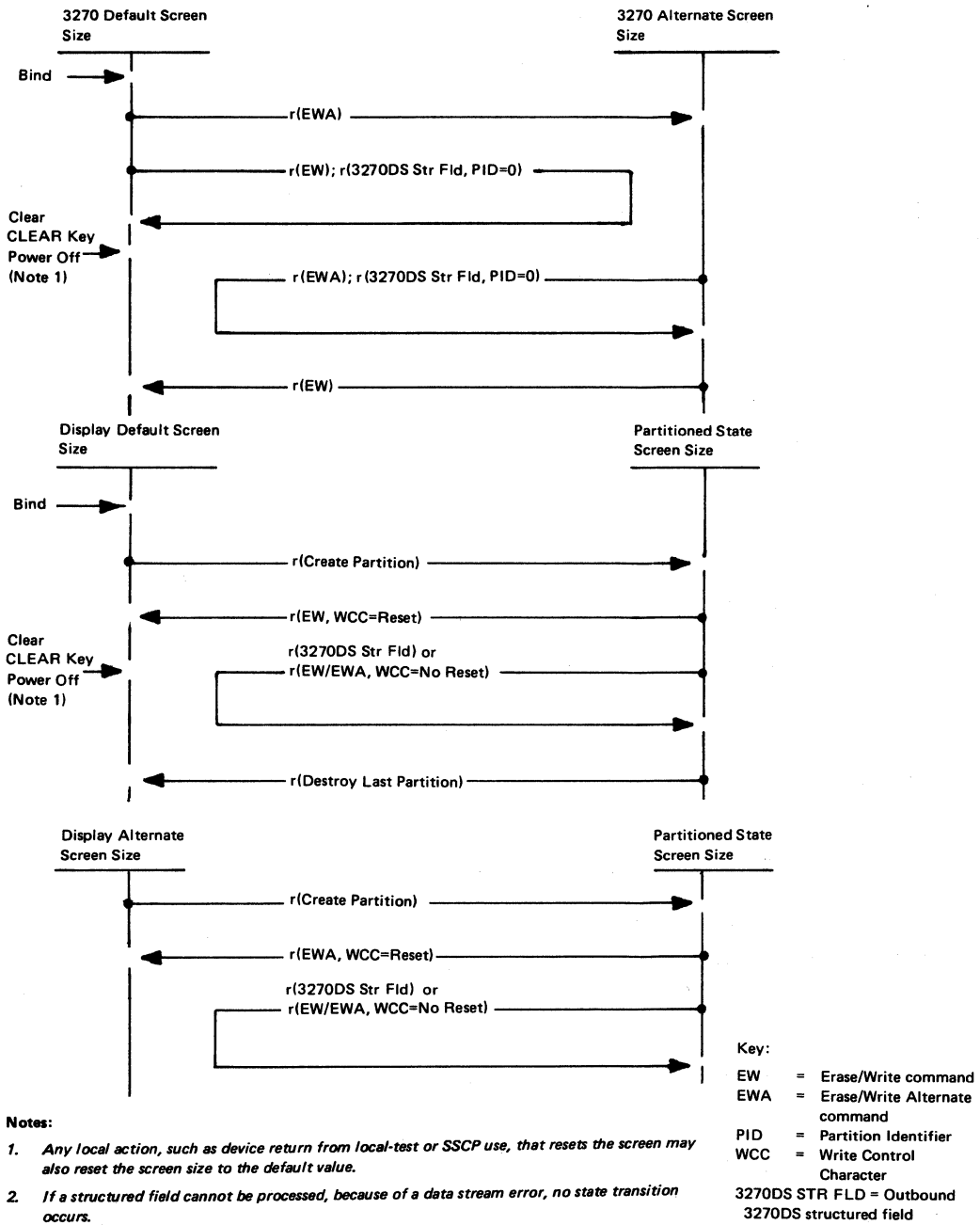
Figure 2-3. Management of Presentation Spaces

**Notes:**

1. *Any local action, such as device return from local-test or SSCP use, that resets the screen 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.*

Key:

| | | |
|---|---|---|
| EW | = | Erase/Write command |
| EWA | = | Erase/Write Alternate command |
| PID | = | Partition Identifier |
| WCC | = | Write Control Character |

3270DS STR FLD = Outbound 3270DS structured field

INBOUND OPERATION (INOP)


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

INOP is set by any of the following:

* With the exception of a Read Partition structured field directed to
  partition 0, whenever INPID is set to 0, INOP is set to Read
  Modified.

* An operator enter action, including a trigger action, sets INOP to
  Read Modified.

* A Read Partition structured field sets INOP to the specified
  operation, namely, RB, RM, RMA, Query.

* Acknowledgment of an inbound transmission sets INOP to Read
  Modified.


INBOUND PARTITION IDENTIFIER (INPID)


The display records the inbound partition identifier (INPID) so that it
knows the 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 by any of the following:

* 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 sets INPID to the specified PID
  unless the PID is X'FF' (query operation), in which case INPID is
  unchanged.

* Destruction of the inbound partition sets INPID to the PID of the
  next higher-numbered partition.  If the destroyed partition was the
  highest-numbered partition, INPID is set to the lowest-numbered
  partition.  If the destroyed partition was the only remaining
  partition, INPID is set to 0.

- Acknowledgment of an inbound transmission sets INPID to 0.

## PARTITION WAIT CONDITION (PWAIT)

The partition-wait condition (PWAIT) is a partition-related input-inhibit condition that, when raised, prevents operator keystroking into that partition. The following rules apply:

- PWAIT is reset by the host; it cannot be reset by the operator.

- While the PWAIT condition is raised for the active partition, an appropriate indicator is displayed.

- At any one time, only one partition can have a PWAIT condition (namely, the inbound partition, INPID).

- PWAIT is used to prevent the operator from changing data after performing an enter action, but before the host acknowledges receipt (for example, by keyboard restore).

- Thus, PWAIT is caused by any operator enter action except a trigger action and is reset by the host's read acknowledgment.

## THE SYSTEM LOCK CONDITION

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

- System lock can be reset by the host or by the operator.

- While system lock is raised for the active partition, an appropriate indicator is displayed (provided a higher-priority condition, such as PWAIT, does not exist).

- At any one time, several partitions can have a system-lock condition.

- System lock is raised, together with PWAIT, by any operator enter action except a trigger action.

- System lock is removed by any of the following:

  - A write with keyboard restore removes system lock from the partition addressed by the write.

- A RESET key pressed by the operator removes system lock from the active partition.

- An end bracket (EB) indicator in the RU chain removes system lock from the inbound partition whose PID is defined by INPID.

## THE TERMINAL WAIT (TWAIT) CONDITION

TWAIT is a terminal-related input-inhibit condition that prevents the operator from keystroking. TWAIT is raised when there is only one partition and the operator performs an enter action. TWAIT is a special case of the PWAIT condition, and the rules defined for PWAIT apply to TWAIT.

# CHAPTER 3. 3270 DATA STREAM COMMANDS

## INTRODUCTION

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

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

```
┌───────────────┬─────┬─────────────────┐
│Write Command  │ WCC │ Orders and Data │
└───────────────┴─────┴─────────────────┘
```

                            or

```
┌─────┬─────────────────────┐
│WSF  │ Structured Field(s) │
└─────┴─────────────────────┘
```

## COMMANDS

Commands are sent to a display to initiate the total or partial writing, reading, or erasing of data in a selected character buffer. Commands are sent as a command code in the first byte of a 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 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 chain with other types of structured fields. (See Chapter 7 for a description of the structured fields.)

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

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


COMMAND CODES


The command codes are not unique code points and, therefore, 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.

Following are the command codes and command abbreviations:

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


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


THE WCC BYTE


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

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

All WCC functions except for Reset MDT are deferred until data is written and orders are performed. See Figure 3-1 for a description of each WCC bit and Figure 3-2 for a summary of the reset actions.

When a data stream contains multiple WCCs (because of their appearance in structured fields), the WCC functions are executed as follows:

Reset----------- Executed in each structured field as it is encountered.

Start print ----- Executed at the end of the RU chain, after the write operation has been completed. Only the last structured field may have a WCC that specified Start Print; otherwise, the chain is rejected (sense code X'1001').

Sound alarm ----- Executed for each structured field, at the end of the operation specified for the structured field.

Keyboard reset -- Executed for each structured field, at the end of the operation specified for the structured field.

Reset MDT ------- Executed for each structured field, prior to the writing of any data or the executing of any orders in the data stream.

| Bit | Explanation |
|---|---|
| 0 | See Note. |
| 1 | WCC reset bit. When set to 1, resets partition characteristics to their system-defined defaults. When set to 0, the current characteristics remain unchanged (no reset operations are performed). See Note. |
| 2 and 3 | Reserved. |
| 4 | Start-printer bit. When set to 1, initiates a local-copy operation of the display surface at the completion of the write operation. When no printer is available, a negative response (0801) is returned. (See Chapter 6 for details of local-copy operation.) |
| 5 | Sound-alarm bit. When set to 1, sounds the audible alarm at the end of the operation if that device has an audible alarm. |
| 6 | Keyboard-restore bit. When set to 1, restores operation of (unlocks) the keyboard. It also resets the AID byte. |
| 7 | Reset modified data tag (MDT) bits in field attributes. When set to 1, all MDT bits in the device's existing character buffer are reset before any data is written or orders are executed. |

Note: *If the reset function is not supported, the only function of bits 0 and 1 is to make the WCC byte an EBCDIC/ASCII-translatable character. Bits 0 and 1 are set in accordance with Figure D-1.*

*If the reset function is supported, bit 1 controls reset/no reset and bit 0 has no function. When bit 1 is used for the reset function, the WCC byte is no longer always EBCDIC/ASCII-translatable; therefore, the reset function cannot be supported in an ASCII environment.*

Figure 3-1. Write Control Character (WCC) Bit Definitions

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

Figure 3-2. Write Control Character (WCC) Reset Actions (for Displays)

WRITE OPERATIONS

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

* Write (W)

* Erase/Write (EW)

* Erase/Write Alternate (EWA)

* Write Structured Field (WSF)

* Erase All Unprotected (EAU)

The first three 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 a display screen, restoring the keyboard, and sounding the audible alarm.

The Write Structured Field command sends structured fields to the display.

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

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

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

WRITE COMMAND

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

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


ERASE/WRITE COMMAND


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

1. Erases the character buffer by writing null characters into all buffer locations.

2. Sets all the associated character attributes to their default value (X'00').

3. Erases all field validation attributes.

4. 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).

5. If bit 1 of the WCC is set to b'1':

   a. Resets the inbound reply mode to Field.

   b. Resets to implicit partitioned 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.

6. Provides an acknowledgment of any outstanding read or enter if the keyboard-restore bit in the WCC is set to b'1'.

7. Provides a negative trigger reply.

8. Performs a write operation.

Bit 1 in the WCC carries reset information used by Erase/Write. If no WCC is defined following an Erase/Write, then no erasing or resetting occurs and it does not acknowledge any outstanding read or enter operation. However, it is treated as a negative reply to a trigger field AID.

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

ERASE/WRITE ALTERNATE COMMAND


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


WRITE STRUCTURED FIELD COMMAND


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

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

The format of a Write Structured Field data stream is:

| WSF | Command | Structured Field | Structured Field | ... |
| --- | --- | --- | --- | --- |

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

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

* Write

* Erase/write

* Erase/write alternate

* Erase all unprotected

The Write command can either be sent as the first byte of the data
stream to write to partition 0 or be enclosed in the Outbound 3270DS
structured field to write to any explicitly created partition.  See
"Create Partition" in Chapter 7 for information on creating a partition.

ERASE ALL UNPROTECTED (EAU) COMMAND


This command does the following:

* Clears all the unprotected character locations of the partition to
  nulls and resets 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 in the first
  unprotected field of the partition's character buffer.

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.


## READ OPERATION


The process of sending a read command and returning the requested
information is called a read operation. A read operation can be
initiated by an explicit read command sent by the application program or
by an operator action, for example, pressing the ENTER key. A read
operation sends an inbound data stream to the application program with
an AID byte as the first byte of the inbound data stream.


AID (ATTENTION IDENTIFICATION)


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 a chain. The
exception is a chain containing an Inbound 3270DS structured field that
itself may contain an AID byte.

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

* Pressing a program-function or program-attention key.

* Pressing the ENTER, CLEAR, or Clear Partition key.

* Reading a magnetic stripe with a magnetic reader.

* Detecting on an attention field with the selector pen.

The possible AID bytes are shown in Figure 3-3.  All AID bytes
transmited by the display are a result of operator actions except for:

* No AID generated

* No AID generated (printer)

* Structured field

* Read partition

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

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

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

* Any write command that has the keyboard-restore bit on in the WCC

* Erase All Unprotected


READ COMMANDS


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

| 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 | | / |
| Trigger action | 7F | | " |
| 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 | < |
| Magnetic Readers: | | | |
| Operator ID reader | E6 | 57 | W |
| Mag Reader Number | E7 | 58 | X |
| 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 | = |

[1]When structured fields are sent inbound, the 1st byte of the inbound data stream is always X'88'. Some structured fields also contain an AID, so the inbound data stream that contains structured fields may have more then 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.

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

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

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

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

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

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

- Field mode supports inbound data that contains field attributes. Start Field and Set Buffer Address orders, field attributes, characters, and graphic escape code X'08' may be included in the inbound Field Mode transmissions.

- Extended Field mode supports field attributes and extended field attributes. Start Field Extended and Set Buffer Address orders, field attributes, extended field attribute, characters, and the graphic escape code X'08' may be included in the inbound transmission.

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


READ BUFFER COMMAND

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

The first three bytes of the read data stream are always the AID code and the 2-byte cursor address.

## Read Buffer Field Mode

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

| AID | Cursor Address (2 bytes) | SF Order | Attribute | Data |
|-----|--------------------------|----------|-----------|------|

## Read Buffer Extended Field Mode

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

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

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

## Read Buffer Character Mode

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

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

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

READ MODIFIED COMMAND


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


Read-Modified Operation


During a read-modified operation, if an AID other than selector-pen attention, PA key, or CLEAR key is generated, all fields that have been modified by keyboard, selector-pen, or magnetic-reader activity are transferred to the application program.  A major feature of the read 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.

The first three bytes of the read data stream are always the AID code and the 2-byte cursor address; these bytes are called the read heading.

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

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

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

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

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

## Short-Read Operation

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

## Read Modified Field Mode

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

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

## Read Modified Extended Field Mode

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

## Read Modified Character Mode

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

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

READ MODIFIED ALL COMMAND

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

Read Modified All Field Mode

Same as for Read Modified.

Read Modified All Extended Field Mode

Same as for Read Modified.

Read Modified All Character Mode

Same as for Read Modified.

READ COMMANDS IN STRUCTURED FIELDS

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

READ OPERATIONS FROM PARTITIONS

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

OPERATOR ENTER ACTIONS

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

An enter action:

1.  Sets INOP and INPID so that operation and partition are known if a
    retry is needed.

2.  Raises the enter-inhibit condition so that the operator cannot
    generate an AID from any partition.

3.  Raises the PWAIT and system-lock conditions if the enter action is
    not a trigger action (AID = X'7F').

4.  Places the device in retry enter state.


APPLICATION-INITIATED READS


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

*   A Read Partition structured field.

*   A Read Buffer, Read Modified, or Read Modified All command received
    for implicit or explicit partition 0.


A READ PARTITION STRUCTURED FIELD


A Read Partition structured field:

1.  Sets INPID if the operation is RB, RM, or RMA.

2.  Sets INOP.

3.  Raises the enter-inhibit condition.

4.  Sends the data to the application program and places the device in
    retry-read state.

Each read operation starts with the application program's sending a Read
Partition structured field to the display. Explicit PID 0 may be read
with a read command and does not require a Read Partition structured
field operation.  The Read Partition structured field identifies a
partition through its partition identifier (PID).  The partition is

assigned a PID when it is created by the Create Partition structured field.

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

```
┌─────────────┬──────────────────┐
│AID = X'88'  │ Structured Fields │
└─────────────┴──────────────────┘
```

where the inbound control information is encoded into structured fields as defined in Chapter 7.

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

- After session activation but before the first Create Partition structured field.

- After all partitions created by Create Partition are deleted.

- After the CLEAR key is pressed.

- When an Erase/Write or Erase/Write Alternate command is received as the first byte of an outbound data stream and the WCC Reset bit is on.

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

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

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

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

INOP records the type of operation (read buffer, read modified, read modified all, query) that is being performed so that it can be repeated if a retry of a Read Partition is required. INOP is discussed in greater detail in Chapter 2.

INPID defines the partition from which the data will be retransmitted. INPID is discussed in greater detail in Chapter 2.

To ensure data integrity for retry of operator enter actions, an input-inhibit condition, called partition wait (PWAIT), prohibits the operator from working in the partition until the application program acknowledges that the transfer of data was successful. PWAIT is discussed in greater detail in Chapter 2.

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

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

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

Notes:

1. The X'80' identifies the structured field as an Inbound 3270DS.

2. From the AID X'61' to the end of the structured field, the data stream is the same as that sent by a corresponding RB, RM, or RMA command in implicit partitioned state.

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

a. Transmits data from partition zero.

b. Leaves the device in its current read state.


A HOST RETRY

When the device is in any of the retry states, the read commands perform a generic retry; that is, they operate on the inbound partition:

- RM performs the operation defined by INOP. This is normally a read-modified operation, unless the original inbound transmission was caused by a host-initiated Read Partition. In this case, INOP may specify RB, RM, RMA, or query.

- RB performs a read-buffer operation on the inbound partition.

- RMA performs a read-modified-all operation on the inbound partition.

READ STATES

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

- Normal-read state. Prior to initiation of a read operation, the device is in normal-read state. An operator enter action or a Read Partition structured field causes the device to transmit the data and to go into a retry state.

- Retry state. There are two forms of retry state.

  - Retry enter, when enter data has been transmitted to the application program.

  - Retry read, when the read data has been transmitted to the application program.

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

Normal Read State

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

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

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

## Retry Enter State

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

| Transition Event | Next State | Operations |
|---|---|---|
| Any read 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 generic retry.

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

| Transition Event | Next State | Operations |
|---|---|---|
| Any read command | Retry READ | Retransmit data from INPID. |
| Host acknowledge | Normal Read | Terminate read operation. |

## Read-State Transitions

The read-state transitions are summarized in Figure 3-4.

| Read States | Normal Read | Retry | |
|---|---|---|---|
| | | Enter | Read |
| Enter Action | 2 | R | R |
| Any Read Command | 1 | G | G |
| Read Partition | 3 | R(1) | R(1) |
| Host Acknowledge | 1 | 1 | - |

Key:

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

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

**Figure 3-4. Read-State Transitions**

## 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 raised for the device.

6. INPID is set to the PID of the active partition, or, for the selector pen, 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 raised 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

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

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

    a. The display is not in RCV or contention (CONT) state.

    b. The RU does not specify CD.

    c. The RU specifies EB; otherwise, step 2 or 3 is performed.

2.  If the device is in normal-read state, the command performs the
    read.  Data from partition 0 is transmitted inbound as defined by:

    a.  The command (RB, RM, RMA)

    b.  The AID(RM)

    c.  The inbound reply mode of partition 0.

        If partition 0 does not exist, the command is rejected.  The
        device remains in normal-read state.

3.  If the device is in retry state, then the command performs a retry.
    Data is transmitted inbound according to the following definition:

    a.  If the command is Read Modified, and INOP specifies Query, then
        the appropriate query replies are transmitted inbound.

    b.  If the command is Read Modified, and INOP specifies RB, RM, or
        RMA, then data is transmitted from the inbound partition
        (INPID) as defined by:

        (1)  The INOP (RB, RM, or RMA)
        (2)  The AID (RM)
        (3)  The inbound reply mode of the inbound partition

    c.  If the command is RB or RMA, then data is transmitted inbound
        from the inbound partition (INPID) as defined by:

        (1)  The command (RB or RMA)
        (2)  The inbound reply mode of the inbound partition


READ ACKNOWLEDGMENT


When the Read Partition structured field is used, an inbound
transmission must be acknowledged before a new inbound operation can be
performed.  (A generic retry is not considered a new inbound operation.)

Read Partition structured fields are acknowledged by any transmission
that, after processing, leaves the display in send or contention state.
This acknowledgment protocol avoids the potential deadlock that could
arise if the display were left in send state with an enter-inhibit
condition raised.  Thus, an enter-inhibit condition implies receive
state.

Notes:

1.  This applies even if there is a data stream error.

2.  The acknowledgment resets INPID to 0 and INOP to Read Modified.

A query operation is initiated by a Read Partition structured field that specifies query. Any of the following transmissions acknowledges a query:

1. A Write Structured Field (WSF) command with no data.

2. A WSF command containing at least one structured field.

3. An Erase/Write (EW), Erase/Write Alternate (EWA) , Write (W), or Erase All Unprotected (EAU) command with no Write Control Character (WCC) and no data.

4. An EW, EWA, or W command with a WCC and no data.

5. An EW, EWA, or W command with a WCC and data.

   Notes:

   1. In case 2, if there is an error in the first structured field, then the transmission is an acknowledgment. In case 5, if there is any error in the data, then the transmission is not an acknowledgment.

   2. The commands used to acknowledge queries are described under "Write Operations."

   3. The acknowledgment resets INPID to 0 and INOP to Read Modified.

A read-modified operation may be caused by:

• An enter action

• An RM command

• A Read Partition structured field specifying Read Modified

A read-buffer or read-modified-all operation may be caused by:

• An RB or RMA command

• A Read Partition structured field specifying RB or RMA

These read operations are acknowledged either by writing to the inbound partition, with keyboard restore, or by destroying the inbound partition.

These acknowledgments reset INPID to 0 and INOP to Read Modified.  Thus, the following transmissions constitute an acknowledgment of a read operation:

1.  A Write Structured Field (WSF) command containing an Outbound 3270DS structured field addressed to the inbound partition and (1) with its WCC specifying keyboard restore or (2) with a partition command of Erase All Unprotected.

2.  If INPID is 0:

    a.  An Erase All Unprotected command.

    b.  A Write, Erase/Write, or Erase/Write Alternate command with its WCC specifying keyboard restore.

3.  A WSF command containing a Destroy Partition structured field for the inbound partition.

4.  A WSF command containing a Create Partition structured field for the inbound partition.

5.  If the device is in explicit partitioned state, an EW or EWA command in which the WCC specifies reset.

6.  If the device is in implicit partitioned state, a WSF command containing a Create Partition structured field.  (This destroys implicit partition 0 and creates a new partition.)

Note: In case 2, if there is a data stream error, then the transmission is not an acknowledgment.  In cases 1, 3, 4, and 6, if there is a data stream error prior to, or within, the structured field providing the acknowledgment, then the transmission is rejected and no acknowledgment occurs.  In case 5, the transmission is always an acknowledgment.

CHAPTER 4. 3270 DATA STREAM ORDERS AND ATTRIBUTES


ORDERS


Orders are included in the inbound and outbound data streams to provide
additional control function. Unlike commands, a number of orders can
occur within a chain. Orders have unique 1-byte codes and can occur in
any position in the data stream. Many orders permit additional
information to follow the order code, for example, buffer addresses or
attributes. These longer sequences, called control sequences, 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.

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 with a sense
code of X'1003',


BUFFER CONTROL ORDERS


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

The order codes are as follows:

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

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


Start Field (SF)


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

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

The byte following the SF order in the write data stream is always treated as a field attribute, even when the byte is actually an order or alphanumeric data.

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


Start Field Extended (SFE)


This order is also used to indicate the start of a field. However, the SFE control sequence contains information on the field's properties that

are described in the extended field attribute.  The SFE order has the
following format:

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

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

Type     Value

| X'C0' | Field Attribute |
|---|---|

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

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

| SFE | 2 | 3270 | Value | Color | Red |
|---|---|---|---|---|---|

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

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

Attribute types and values that are unknown or cannot be maintained and
returned inbound by an implementation will be rejected (sense code
X'1003'.

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

| X'11' | Buffer Address (2 bytes) |
|-------|--------------------------|

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

If the SBA order specifies an invalid address (out of range), the write operation is terminated immediately (with a negative response X'1005').

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

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

The buffer address is relative to the origin of the buffer of the partition to which the orders and data are directed.

The partition may be defined to operate with either 16-bit addressing or 12- or 14-bit addressing. When 16-bit address mode is specified for a partition, outbound buffer addresses are interpreted as 16-bit binary, and inbound addresses are generated as 16-bit binary. Addresses that are not in the proper form are rejected.

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

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

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

If the flag bits are 01 or 11, the next 14 bits are to be interpreted as
a 2-character address (6 bits in each byte). The six 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 posiiton 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 buffer address
beginning with the flag bits 10 causes the data stream to be rejected.
(See Figure A-1.)

If 16-bit addressing mode is not 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 4,096
characters, then all addresses from that partition are in 14-bit form.
If the partition is less than 4,096 characters, then all addresses are
generated in 12-bit form.

## Set Attribute (SA)

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

| X'28' | Attribute Type | Attribute Value |
|-------|----------------|-----------------|

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

* A new SA order changes it.

* Another write-type command is sent.

* CLEAR key is pressed.

* Power is off.

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

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

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

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

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

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

Modify Field (MF)

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

The MF control sequence has the following format:

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

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

* Protected

* Alphanumeric

* Normal intensity

* Nonselectable

* Nonloadable character set (character set shipped with the device)

* Red

* No highlight

* No validation

Then, interpretation of the order sequence:

| SBAxx | MF | 2 | Color | Blue | Highlight | Blink |
|-------|----|---|-------|------|-----------|-------|

leaves the field with the following properties:

* Protected

* Alphanumeric

* Normal intensity

* Nonselectable

* Nonloadable character set

* Blue

* Blink

* No validation

If the current buffer address does not contain a field attribute, the MF order is rejected (sense code X'1005').

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

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

Invalid attribute types are rejected (sense code X'1003'). Attribute values that are unknown or cannot be maintained and returned inbound by an implementation must be rejected (sense code X'1003').

## Insert Cursor (IC)

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 an automatic scroll. The position of the cursor after an automatic scroll is described under "Scrolling" in Chapter 2.

## Program Tab (PT)

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 of each character nulled to its default value.

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.

4-8

## Repeat to Address (RA)

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

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

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

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

or

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

The 12/14-bit and 16-bit addressing considerations discussed under "Set
Buffer Address (SBA)" above apply to the RA order.  Also, if an invalid
stop address is specified (for example, greater than 1919 for a
1920-character screen), the write operation is terminated at this point
without storing the character.

The character to be repeated may be any 8-bit EBCDIC code or 2-byte
graphic-escape (GE) sequence.  If an EBCDIC code, including control
codes, has a valid graphic representation in the device, then that
graphic is displayed.  Otherwise, the RA order is rejected (sense code
X'1003').

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

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

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

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


## Erase Unprotected to Address (EUA)


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 stored into.

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

| X'12' | Stop Address (2 bytes) |
|-------|------------------------|

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

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

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

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

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


## Graphic Escape (GE)


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

Support of graphic escape is optional; when not supported, it is rejected (sense code X'1003').

When supported, its 2-byte format is:

```
┌──────┬────────────────┐
│X'08' │ Character Code │
└──────┴────────────────┘
```

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

1. The GE+Character sequence always takes precedence over any other character set specified in the extended field attribute or character attribute, and applies to both inbound and outbound data streams, independent of the inbound reply mode.

2. GE+Code points of X'00' through X'3F' and X'FF' are undefined and are rejected (sense code X'1003').

3. GE+Code points of X'40' through X'FE' are valid graphic character codes.

4. If there is no alternate character set, the X'40' to X'FE' code points are rejected (sense code X'0863').

FORMAT CONTROL ORDERS

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

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

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

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

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

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


CHARACTER SETS


There is a maximum of 190 language-dependent graphics in any character set. Typically, there are fewer than 128 unique graphics in most languages. Those codes not corresponding to defined graphics in a character set are displayed as undefined graphics; however, the integrity of the code point, including space (X'40' or X'0840'), is maintained. Any data stream attributes associated with an undefined graphic still apply. If a GE code (X'08') is encountered, then the next character set is defined to be from an alternate character set. The alternate character set, if used, is applied as the data stream is interpreted.

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' is a space.

- X'08nn' with nn = X'00' through X'3F' and X'FF' are undefined code points.

- 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'40' is a nonloadable space.

- X'41' through X'FE' are loadable graphics.


4-12

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


## ATTRIBUTES

Attributes are sent in the 3270 data stream to define field(s) or the
properties of the field(s) or character(s).

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.

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, and the 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 characteristics such as the color, extended
highlighting, and character set of the character. The extended field
attribute is subordinate to the character attribute; that is, if a
character attribute is present and defines a color, extended
highlighting, or character-set value other than default, it overrides
one or more of the attribute types identified in the extended field
attribute.

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

| Attribute Type | Attribute Value |
|----------------|-----------------|

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

FIELD ATTRIBUTES

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

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 of unprotected fields.

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

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

Figure 4-1. Bit Definitions for Field Attributes

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 field attribute. The remaining highlighting functions are in the extended field attribute and character attribute.

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

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

- When the operator changes an unprotected field, which includes pressing the erase end of file (EOF) key, the display sets the modified data tag (bit 7) of the field attribute to 1.

- When the operator uses the Clear key, field attributes and all other characters in a formatted buffer are set to nulls (X'00').

- When the application program writes data, field attributes are not protected from being overwritten by alphanumeric data in the data stream.

- When the ERASE INPUT key is pressed, it sets off the MDT bit on all unprotected fields.

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

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


EXTENDED FIELD ATTRIBUTES


Extended field attributes specify these field properties:

- Field validation

- Extended highlighting

- Extended color (referred to as <u>color</u> throughout this manual)

- Character set

The field-validation functions are mandatory-fill field, mandatory field, and trigger field. .

A mandatory-fill field 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.  A mandatory field requires the operator to enter data in the field.  A trigger field causes the data entered by the operator to be sent to the application program as soon as the operator completes each such field.

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

The color function allows selection of the color of characters in the field.

The character-set function 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.


CHARACTER ATTRIBUTES


Character attributes specify these character properties:

- Extended highlighting

- Extended color (referred to throughout this manual as <u>color)</u>

- Character set

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

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


CONFLICT RESOLUTION BETWEEN ATTRIBUTES


In field-oriented applications, extended field attributes may be used to define the color, highlighting, field validation, and character set of the field.  At times, however, there may be a conflict in the properties

of the field as specified in extended field attributes, if it exists, and the properties of individual characters as specified in 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 the 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 if it exists (in the extended field attribute). Otherwise, the character attribute overrides the value established for the field.

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 because of an insert or delete operation, the attribute moves with the character.

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

| If the screen is | And the character attribute is set to[1] | Then |
|---|---|---|
| Formatted | Default | The extended field attribute is used to display/print that character. |
| Formatted | Nondefault | The character attribute is used to display/print that character for each nondefault (specified) property, and the extended field attribute is used to display/print defaulted properties. |
| 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.

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

| Type Code | Attribute Type | May be used to define: Extended Field Attribute | Character Attribute |
|-----------|----------------|--------------------|---------------------|
| X'00' | Reset all character attributes to default value | No | Yes |
| X'C0' | Field attribute | NA | NA |
| X'C1' | Field validation | Yes | No |
| X'41' | Extended highlighting | Yes | Yes |
| X'42' | Color | Yes | Yes |
| X'43' | Character set | Yes | Yes |
| Other | Reserved | NA | NA |
| NA = Not Applicable | | | |

ATTRIBUTE VALUES AND SELECTION RULES

The value bytes for all defined attribute types are shown below.
Attribute values not shown are rejected (sense code X'1003'). Attribute
values that are unknown or cannot be maintained and returned inbound by
an implementation also must be rejected (sense code X'1003').

Reset All Character Attributes

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

```
Type       Value
┌──────────┬─────────────────────────┐
│          │ Reset All Attribute Types│
│ X'00'    │ (X'00')                  │
└──────────┴─────────────────────────┘
```

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


Field Attribute


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

```
Type       Value
┌──────────┬──────────────┐
│          │   Field      │
│ X'C0'    │   Attribute  │
└──────────┴──────────────┘
```

The definition of each bit position of the field attribute value byte is
shown in Figure 4-1.


Field Validation


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

```
Type       Value
┌──────────┬──────────────┐
│          │   Field      │
│ X'C1'    │   Validation │
└──────────┴──────────────┘
```

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

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

For detailed information, see "Field Validation."


Extended Highlighting


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

both).  A highlighting property specified by the extended field
attribute does not affect the intensify property specified by the field
attribute.

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

Type        Value

| X'41' | Highlighting |

Following are valid settings for the highlighting value byte:

Setting        Highlighting Property
X'00'          Default
Other          As indicated by Query Reply (Highlight)


Color


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

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

Type        Value

| X'42' | Color |

Following are valid settings for the color value byte:

Setting        Color Property
X'00'          Default color [as defined by Query Reply (Color))]
X'F7'          Multicolor
Other          As indicated by Query Reply (Color)

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

The settings X'00' and X'F7' have unique data stream meaning, but they
do not imply unique color assignment.  The X'00' value selects the
device default color indicated in the Query Reply (Color) structured
field.  The X'F7' value indicates 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'), only
the colors specified in Query Reply (Color) are used in the presentation
of data.

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


Character Set


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

| Type | Value |
|------|-------|
| X'43' | Character Set |

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

Setting          Character Set
X'00'            Default character set
X'01' to X'FE'   Local character set identifier (LCID)
                 (X'F1' cannot be used in SFE or MF order)

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

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


FIELD VALIDATION


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

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

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


MANDATORY FILL


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

* Entering data in every character position of the field.

* Not entering data in every position, but entering a DUP character or an error character in the last position filled.

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

* A data entry key

* The delete key

* The ERASE EOF key

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

* A write operation to that partition

* An operator erase-input keystroke

* Successful fill validation for that field

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

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

Fill validation checks the field for null characters. If the field contains any null characters, then:

* The mandatory-fill input-inhibit condition is raised.

* The cursor is not moved out of the field.

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

The RESET key must be used to remove the 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 key is not affected by the validation of a mandatory-fill field; that is, while the screen cursor is positioned within a mandatory-fill field, the operator may jump to another partition. The screen cursor is moved, but the current cursor position of the partition is not changed. Consequently, the jump key does not cause fill validation to occur.

2. Similarly, validation is not affected by the jump key. If the operator causes the MDT to be set to 1, and then jumps to another partition and back again, fill validation applies only at field exit time.

3. If a field has an extended field attribute with mandatory fill and trigger set on, fill validation occurs before the trigger action.

4. If the cursor is positioned in a primed mandatory-fill field, any enter action, such as a PF key, causes fill validation to occur.

Programming Notes:

1. If mandatory fill is specified for a "protected" field, it has no effect on the field.

2. If the host presents (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, the field, which may contain some nulls (that is, an "incomplete" fill field), is transmitted inbound.

3. Similarly, if a mandatory-fill field is "selectable", operator selection causes the MDT bit to be set on. Again, if the operator does not type any data into such a field, the incomplete fill field is transmitted inbound on a read-modified operation.


MANDATORY FIELD


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

• The ENTER key

- A PF key

- A selector-pen detect on a field with a designator of &

- A cursor select on a field with a designator of &

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-field 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-field 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 reenter the enter action for the data that was inhibited.


TRIGGER FIELD


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

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

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

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

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

cursor is moved to the next position unless it is overridden by an Insert Cursor order contained in the reply.

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

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

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

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

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

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

*   The RM command transmits the trigger field.

*   The RMA command transmits all fields that have the MDT bit set on.

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

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

Details of the possible replies are given below. The terms and concepts used here are explained under the heading "Read Operations from Partitions" in Chapter 3.

In summary, an affirmative reply is:

- A WSF command containing an Outbound 3270DS structured field, addressed to the inbound partition (INPID). The structured field specifies a write command and contains a WCC that specifies keyboard restore.

- If INPID is 0, a write command with a WCC that specifies keyboard restore.

- A null RU chain with CD or EB.

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

In summary, a negative reply is:

- A WSF command containing any of the following structured fields:

    - Outbound 3270DS, addressed to INPID, that specifies a write command and contains a WCC that does not specify keyboard restore, or that specifies an EAU command.

    - Outbound 3270DS, addressed to a partition where the PID is not INPID.

    - A control operation, namely, those operations requested by the Reset Partition, Load Programmed Symbols, Set Inbound Reply Mode, Create Partition, Destroy Partition, Activate Partition, and Set Window Origin structured fields.

- If INPID is 0, a write command with a WCC that does not specify keyboard restore, or an EAU command.

- If INPID is nonzero, a W, EW, EWA, or EAU command.

- A WSF command with no data included in the structured field.

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

- Read Modified, Read Modified All, or Read Buffer command

- A Read Partition structured field contained in a WSF command

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

If the reply from the host is a write operation that includes an IC order, then the cursor is positioned at the address specified in the data. If the reply is negative, the cursor is moved from its position

within the trigger field to where the host has placed it, and the invalid-field input-inhibit condition is raised.

Programming Note: To avoid operator confusion, the host 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 host has placed it, and the keystroke queue is processed. Thus any data entry keystrokes are placed at the host-specified position.

Programming Note: The ability to move the cursor allows "logical cursor movement" to be programmed; that is, the position of the cursor can be made dependent on the data content of a previous field. If such a function is implemented, you should ensure that the operator understands the protocol for that application.

Notes:

1. Trigger action occurs if scrolling action moves the cursor out of a primed trigger field.

2. If the operator performs an enter action when an unmodified mandatory-field input-inhibit condition is raised, the cursor is moved to the mandatory field. Thus, if this occurs while the cursor is positioned in a trigger field, the cursor movement causes a trigger action.

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

   In this situation, moreover, and only in this situation, if the reply from the host was a write operation containing an Insert Cursor order, the host-specified cursor position is overridden by the mandatory field cursor position; that is, after the reply from the host, the cursor is positioned in the mandatory field and the mandatory-field input-inhibit indicator is displayed.

3. A change of the active partition does not cause trigger action.

4. If the keystroke queue gets full before a write is received from the host, an input-inhibit condition is raised.

5. The MDT bit is not changed by the trigger action. The host must decide what to do when a field check fails and then issue the appropriate orders in the data stream.

6. Normally, the MDT bit is set on as a result of data entry. If the field is selectable, however, then 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 key is unaffected by the validation of a trigger field; that is, while the screen cursor is positioned within a trigger field, the operator may jump to another partition. The screen cursor is moved, but not before the current cursor position is recorded. Consequently, jump does not cause a trigger action. Similarly, validation is not affected by the jump key.


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' (multicolor), the processing is as follows:

1. If the character attribute contains an ID for a single-plane character, the character code point and the character-set-value byte are used to obtain the character definition.

2. If the character attribute contains an ID for a triple-plane character set, the character code point and the character-set value are used to obtain three character definitions, which are then ORed together to produce a single definition.

3. If the character attribute has a highlight value of reverse video, then the character definition obtained from step 1 or 2 is complemented; that is, all 0 bits are changed to 1's, and all 1 bits are changed to 0's (assuming the device supports reverse video).

4. If the character attribute has a highlight value of underscore, then an underline bar is effectively ORed into the character definition obtained from step 1 or 2.

5. The character definition is displayed or printed in the color specified. If the selected color is not supported by the device, then the device default color is used.

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 <u>reverse video</u>,
    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 <u>underscore</u>,
    then an underscore bar is effectively ORed into each of the three
    character definitions obtained from step 1.

4.  Each of the three character definitions is displayed in the color
    associated with its plane.

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

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)" in
   Chapter 7 for more information.)


<u>DEFAULTS</u> <u>FOR</u> <u>ATTRIBUTES</u>


The attribute defaults are given in Figure 4-2.

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

| | | Default Condition | |
|---|---|---|---|
| | | For a Character | |
| Attribute Type | For a Field | Formatted Screen | Unformatted Screen |
| Field attribute | Unprotected, A/N, display/nondetect, no intensify, MDT off | Not applicable | Not applicable |
| Color | 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 |

Figure 4-2. Attribute Default Conditions

CHAPTER 5.  MAGNETIC-READER, KEYBOARD, AND SELECTOR-PEN OPERATIONS

MAGNETIC-READER OPERATIONS *

This chapter describes the magnetic-reader operations for the magnetic slot reader (MSR).  This description also applies to the magnetic hand scanner (MHS).  Two modes of magnetic-reader operations are described: the 3275/3277-compatible mode and the numeric/alphanumeric mode.

3275/3277-COMPATIBLE OPERATION

Following is the numeric character set used for the 3275/3277-compatible mode operation.

| Stripe Codes | | Display-Generated Codes | | |
|---|---|---|---|---|
| Character | Hex | EBCDIC | ASCII | |
| 0 | 0 | F0 | 30 | |
| 1 | 1 | F1 | 31 | |
| 2 | 2 | F2 | 32 | |
| 3 | 3 | F3 | 33 | |
| 4 | 4 | F4 | 34 | |
| 5 | 5 | F5 | 35 | |
| 6 | 6 | F6 | 36 | |
| 7 | 7 | F7 | 37 | |
| 8 | 8 | F8 | 38 | |
| 9 | 9 | F9 | 39 | |
| | A | 7A | 3A | Operator ID (OID) |
| | B | 7B | 23 | Start of record (SOR) |
| | C | 7C | 40 | unassigned |
| | D | 7D | 27 | Field separator |
| | E | 7E | 3D | unassigned |
| | F | 7F | 22 | End of record (EOR) |

Magnetic stripes may contain up to 128 characters.  This includes the start of record (SOR), which does not appear in the input data stream.  Also included are the end of record (EOR) and the longitudinal redundancy check (LRC), which do appear in the input data stream.  Note that the application program must do the LRC verification.  (It is recommended that the application program verify that a X'F7' appears just prior to the LRC.)

The data portion of the stripe, the part between the initial start of record and the end of record (EOR), should contain only 0-9, D (field separator), or the OID (which is the A immediately following the start-of-record indicating secure data).  The secure data is for the

application program's use; the display treats all stripes as secure. Stripes containing A (other than special A), B, C, or E may be rejected. However, if accepted, the A, B, C, and E will be sent to the application program as 7A, 7B, 7C, and 7E, respectively. The graphics associated with A, B, C, and E are whatever the attaching device associates with the resulting EBCDIC or ASCII codepoint.

When the start-of-record character is read from the stripe, an attribute character is entered automatically into the cursor-identified location of the buffer, provided the cursor is at an unprotected character location. This attribute character defines the stripe data field following it as protected, alphanumeric, nondisplay/nonprint. As the rest of the stripe data is read into the buffer, it is stored, starting at the first character location after this attribute character. As each character is stored in the buffer, the cursor advances one buffer location. Note that the cursor may not be visible to the operator until the end of the operation. A magnetic-reader operation automatically causes the keyboard to lock and a magnetic-reader AID to become pending. At an appropriate time the AID character and the display buffer data are sent to the application program during a read-modified operation.

No additional keyboard data can follow a magnetic-reader operation. There are some differences in the data-stream content, depending on whether the device buffer is formatted or unformatted.

- Unformatted display. When a magnetic-reader input operation occurs, the device buffer becomes formatted because of the insertion of the protected, nondisplay attribute with the MDT bit set on. Any data that was previously displayed will no longer be displayed. (See Figure 5-1.)

- Formatted display. A magnetic-reader operation may be initiated only when the cursor is located in an unprotected field. The input data stream resulting from a magnetic reader operation will contain at least two fields with the MDT bit set on (new data field and the previous data field) because all the information from the stripe is treated as data until the information has been entered. Also, the MDT bit is set on in the attribute character created by the start of record. Figures 5-2 and 5-3 show how to prepare an unprotected field for magnetic-reader input. In Figure 5-2, the cursor follows the operator instructions. In Figure 5-3, the operator instructions are in a protected field and the cursor follows an unprotected field attribute character.

When the extended field and character attributes are supported, a magnetic-reader operation will cause an extended field attribute of X'00' to be generated in addition to the field attribute. Data characters will generate character attributes of X'00'.
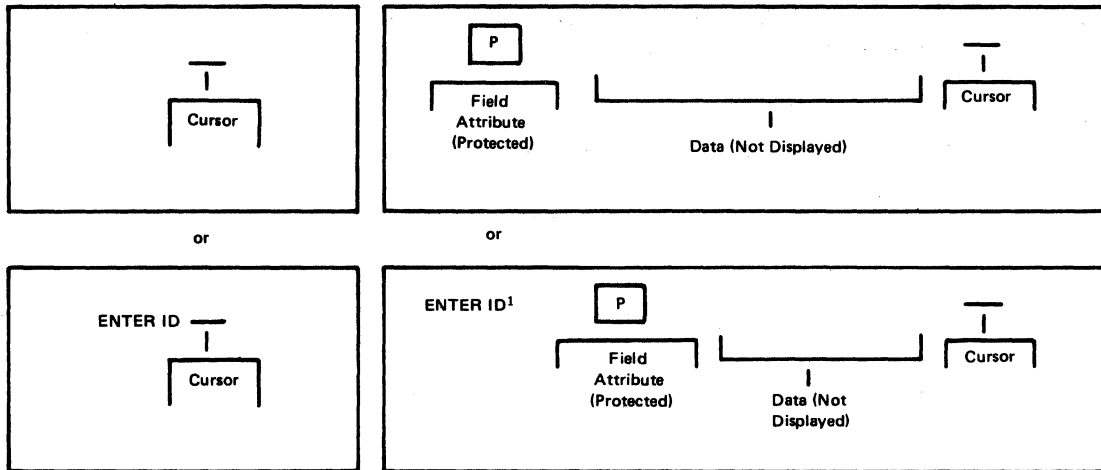
Display Screen before Magnetic-Stripe
Input Is Entered

Display Screen after Magnetic-Stripe Input Is Entered

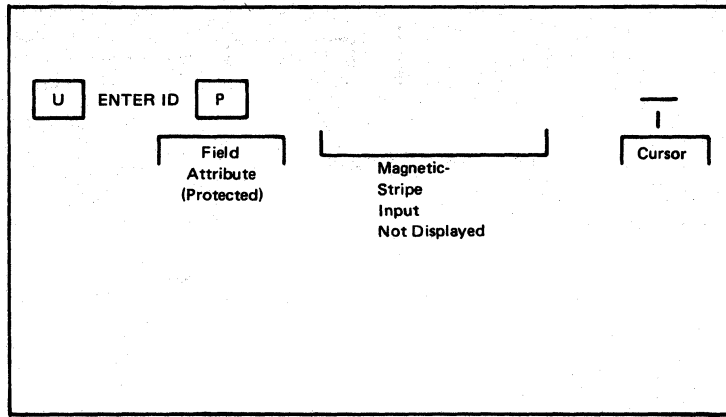| | |
|---|---|
| AID | Set to indicate magnetic-stripe input. |
| Cursor Address | Address of the cursor upon completion of the operation. |
| SBA | Address of the first data character following the field attribute. |
| Field Attribute Address + 1 | The magnetic-stripe data followed by any information present in the buffer. The additional information could have been initiated by the |
| Data | application program as ENTER ID, as shown in the example, or entered by the operator before the magnetic-stripe operation was started. |

Note that, with an unformatted screen, the magnetic-stripe data is the first text in the data stream presented to the application program.

[1] The ID is not displayed, because it is within a nondisplay field, defined by the magnetic-stripe field attribute.

**Figure 5-1. Display Screen Activity before and after Magnetic-Stripe Input (Unformatted Display)**
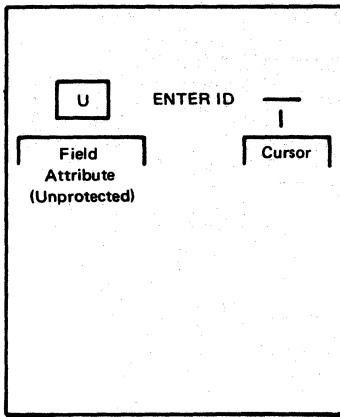
Display Screen before Magnetic-Stripe
Input Is Entered

Display Screen after Magnetic-Stripe Input Is Entered

| | |
|---|---|
| **AID** | Set to indicate magnetic-stripe input. |
| **Cursor** | |
| **Address** | Address of the cursor upon completion of the operation. |
| **SBA** | |
| **Field Attribute** | |
| **Address + 1** | Address of the unprotected (U) field attribute + 1. |
| **Data** | ENTER ID, in the example above. |
| **SBA** | |
| **Field Attribute** | Address of the protected field attribute + 1. In this case, the address of the first data character from the magnetic-stripe input following the field attribute. |
| **Address + 1** | |
| **Data** | The magnetic-stripe input (and any data between the cursor and the next field attribute). |

Figure 5-2. Display Screen Activity before and after Magnetic-Stripe Input (Formatted Display
with Unprotected Field Attribute)
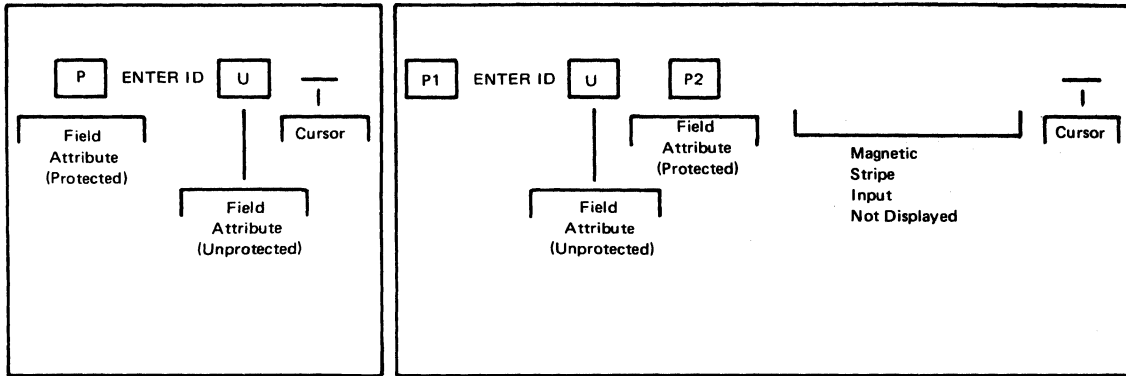
Display Screen before Magnetic-Stripe

| P | ENTER ID | U | | | ‾ |

Field Attribute (Protected)

Field Attribute (Unprotected)

Cursor

Display Screen after Magnetic-Stripe Input Is Entered

| P1 | ENTER ID | U | | P2 | | ‾ |

Field Attribute (Unprotected)

Field Attribute (Protected)

Magnetic Stripe Input Not Displayed

Cursor

Display Screen before Magnetic-Stripe
Input Is Entered

Display Screen after Magnetic-Stripe Input Is Entered

| AID | Set to indicate magnetic-stripe input. |
|---|---|
| Cursor<br>— Address — | Address of the cursor upon completion of the operation. |
| SBA | |
| Field Attribute<br>— Address + 1 — | Address of the unprotected (U) field attribute + 1. In the example above, it is the address of field attribute P2. |
| SBA | |
| Field Attribute<br>— Address + 1 — | Address of the P2 field attribute + 1. In this case, the address of the first data character from the magnetic-stripe input following the field attribute. |
| Data | The magnetic-stripe input (and any data between the cursor and the next field attribute). |

Figure 5-3. Display Screen Activity before and after Magnetic-Stripe Input (Formatted
Display with Protected Field Attribute)

## NUMERIC/ALPHANUMERIC OPERATION

For magnetic stripes using the numeric/alphanumeric character sets, the two character positions following the start sentinel (SS) perform a header function that designates to the display the data field contents, as follows:

*,* = numeric-character set, nonsecure data
'A',* = numeric-character set, secure data
'C','A' = numeric/alphanumeric-character set, nonsecure data
'A','C' = numeric/alphanumeric-character set, secure data

Note: * = 0 through 9 or D.

Any magnetic stripes not conforming to the above header information will be rejected, that is, the information will not be sent to the application. Also, if a display does not support a particular character set, cards with magnetic stripes using that character set will be rejected.

For the alphanumeric character set magnetic stripes, the data field starts after the 2-character header. For the secure numeric character set magnetic stripes, the data field starts after the A character. For nonsecure numeric character set magnetic stripes, the data field starts after the SS B. For the numeric/alphanumeric character set magnetic stripes, the data field ends with (but does not include) the EOC (F). Only the data field is sent to the application program. The LRC check is done by the display. The LRC and the end sentinel are not sent to the application program.

The data field, up to 125 characters less the header, is sent to the application program in conformance with the magnetic stripe character set designated by the header (that is, the numeric or alphanumeric character set).


## STRIPE CODES AND APPLICATION PROGRAM CODES

The following shows the relationship of the magnetic stripe code to the codepoint the display generates. This relationship is independent of the graphic character set supported by the attaching display, for example, the same codepoint is generated for the US EBCDIC character set displays as would be generated for a German character-set display. The graphic displayed when display is allowed depends on the graphic character set(s) supported by the attaching display.

Following is the magnetic stripe numeric character set:

| Stripe Codes | | Display-Generated Codes | |
|---|---|---|---|
| Character | Hex | EBCDIC | ASCII |
| 0 | 0 | F0 | 30 |
| 1 | 1 | F1 | 31 |
| 2 | 2 | F2 | 32 |
| 3 | 3 | F3 | 33 |
| 4 | 4 | F4 | 34 |
| 5 | 5 | F5 | 35 |
| 6 | 6 | F6 | 36 |
| 7 | 7 | F7 | 37 |
| 8 | 8 | F8 | 38 |
| 9 | 9 | F9 | 39 |

Control Data Characters
Hexadecimal

| | |
|---|---|
| A (secure data) | not sent in the data stream |
| B (start sentinel) | not sent in the data stream |
| C (reserved) | not sent in the data stream |
| D (space) | 40 |
| E (reserved) | not sent in the data stream |
| F (end sentinel) | not sent in the data stream |

The control characters A and B must appear in their respective positions on the magnetic stripe or they will be treated as errors. The control characters C And E will always be treated as an error. The A means secure data (protected, nondisplay, and nonprint) when located immediately after the SS. The B is SS. The F is the end sentinel (ES).

Another SS, called reverse start sentinel (RSS), follows the LRC on the magnetic stripe. The RSS is for magnetic stripe card readers that can scan the card in both directions. The B is also used for the reverse start sentinel. This type of operation is primarily for magnetic-hand-scanner operations.

Following is the alphanumeric character set:

| Stripe Codes | Display-Generated Codes | |
|---|---|---|
| Hex | EBCDIC | ASCII |
| 0A[1] | F0 | 30 |
| 1A[1] | F1 | 31 |
| 2A[1] | F2 | 32 |
| 3A[1] | F3 | 33 |
| 4A[1] | F4 | 34 |
| 5A[1] | F5 | 35 |
| 6A[1] | F6 | 36 |
| 7A[1] | F7 | 37 |
| 8A[1] | F8 | 38 |
| 9A[1] | F9 | 39 |

| Stripe Codes | Display-Generated Codes | |
| --- | --- | --- |
| Hex | EBCDIC | ASCII |
| C1 | C1 | 41 |
| C2 | C2 | 42 |
| C3 | C3 | 43 |
| C4 | C4 | 44 |
| C5 | C5 | 45 |
| C6 | C6 | 46 |
| C7 | C7 | 47 |
| C8 | C8 | 48 |
| C9 | C9 | 49 |
| D1 | D1 | 4A |
| D2 | D2 | 4B |
| D3 | D3 | 4C |
| D4 | D4 | 4D |
| D5 | D5 | 4E |
| D6 | D6 | 4F |
| D7 | D7 | 50 |
| D8 | D8 | 51 |
| D9 | D9 | 52 |
| E2 | E2 | 53 |
| E3 | E3 | 54 |
| E4 | E4 | 55 |
| E5 | E5 | 56 |
| E6 | E6 | 57 |
| E7 | E7 | 58 |
| E8 | E8 | 59 |
| E9 | E9 | 5A |
| 0C | 4A | 5B |
| 1C | 5A | 5D |
| 3C | 7A | 3A |
| 4C | 4C | 3C |
| 5C | 5C | 2A |
| 6C | 6C | 25 |
| 7C | 7C | 40 |
| 0D | 4B | 2E |
| 1D | 5B | 24 |
| 2D | 6B | 2C |
| 3D | 7B | 23 |
| 4D | 4D | 2B |
| 5D | 5D | 29 |
| 6D | 6D | 5F |
| 7D | 7D | 27 |
| 0E | 4F | 21 |
| 1E | 5F | 5E |
| 2E | 6F | 3F |
| 3E | 7F | 22 |
| 4E | 4E | 2B |
| 5E | 5E | 3B |
| 6E | 6E | 3B |
| 7E | 7E | 3D |

| Stripe Codes | Display-Generated Codes | |
| --- | --- | --- |
| Hex | EBCDIC | ASCII |
| | | |
| E0 | E0 | 5C |
| E1 | 61 | 2F |
| DA | 50 | 26 |
| EA | 60 | 2D |
| CA | 40 | 20 |
| 00 | F0,F0 | 30,30 |
| 01 | F0,F1 | 30,31 |
| | | | | | |
| 89 | F8,F9 | 38,34 |
| 99 | F9,F9 | 39,39 |

[1]Indicates that A̲ is a filler character when numerics are not paired.

When using the alphanumeric character set, the numerics are coded as two
4-bit numeric characters for each 8-bit byte.  As a result, there must
be either an even number of numerics in any continuous string of
numerics, or an odd number of numerics with a filler character.  The
code sequence (12XYZ) is represented on a card as (12E7E8E9).  The code
sequence (123XYZ) is represented on a card as (123AB7E8E9).  This limits
the number of characters on a card to 62 alpha and special characters
and to 124 numeric characters.  The hexadecimal A̲ (1010) is used as the
filler character.

The initial B̲ is start sentinel (SS).  The F̲ is the end sentinel (ES).
The reverse start sentinel (RSS) follows the LRC on the magnetic stripe
card.  The RSS is used by devices using a magnetic stripe card reader
that can scan the card in either direction.

Any card code not shown (except for CAEE--- the test card) will cause a
rejection of the card when that code appears between the header and the
end sentinel.  The first F̲ encountered after the start sentinel will be
treated as the end sentinel.  The code following the ES is considered
the LRC.


SECURE/NONSECURE MAGNETIC STRIPE CARDS


Display operations when using the numeric/alphanumeric character set
magnetic readers closely parallels the operation of the
3275/3277-compatible operation as far as the application program is
concerned.

There are differences in the handling of secure and nonsecure cards.

## Secure Magnetic Stripe Cards

Handling of the numeric/alphanumeric magnetic-stripe cards designated as secure follows that of the 3275/3277-compatible operation. That is, the same rules for the cursor being in an unprotected field and generation of a protected, nonprint/nondisplay field attribute and automatic entry all apply. An AID code (E7 for EBCDIC, 58 for ASCII) is generated indicating magnetic reader extended. This informs the application program that the magnetic stripe data conforms to the numeric/alphanumeric character set operation. Note that no indication is given on whether the data field is in the numeric or alphanumeric character set. No end sentinel or LRC is sent with the data. When the application receives the E7,(58), AID, the data is valid; the LRC check has already been done.

## Nonsecure Magnetic Stripe Cards

Nonsecure numeric/alphanumeric character-set magnetic-stripe cards appear to the application program as keyboard entered data. An exception is that the magnetic-stripe data does not follow the keyboard rules relative to numeric field/numeric lock/keyboard type. Nonsecure magnetic-stripe data may be entered in any unprotected field. The cursor must still be in an unprotected field for a successful magnetic-reader operation. However, no protected, nonprint/nondisplay field attribute is generated. No AID is generated, and no automatic entry will take place. Sending of the data relies on normal entry operation--the ENTER key, selector-pen attention, and so on. An implementation may optionally provide for a customer selected override that causes auto entry of nonsecure data. However, the AID generated will be coded as the ENTER key.

## TEST CARD

For displays, any nonsecure card of known content may be used as a test card. The alphanumeric test card (CAEE---) will be treated as a normal nonsecure card, except that the EE will be deleted. There will be no automatic verification of the test data. Visual verification will be done by the operator. All magnetic-reader data will be inhibited from being sent to the application program when the display is in subsystem test mode.

## KEYBOARD FUNCTIONS

The keyboard enables the operator to change, edit, or create character displays except within fields defined as protected from keyboard operations. As messages are being composed or modified by keyboard operations, the changes are inserted in the character buffer and displayed on the screen.

5-10

KEYS THAT AFFECT THE DATA STREAM


The data stream is oriented toward a keyboard display, but
implementations have the option of providing or not providing a
keyboard.  If supported,· the keys and key positions depend on the type
of keyboard.  The keyboard description that follows is limited to the
functions that have an effect on the data stream.


## Cursor


A special symbol, called a cursor, indicates where the next character
entered from the keyboard will be displayed on the display surface
(viewport).  For example, assume the cursor character is an underscore.
When the cursor is displayed under one character in a row of characters,
that character can be changed or deleted by keyboard action.  Also, if
the cursor is displayed under a position without a character, a
character can be placed in that position by keyboard action.  All these
operations, when performed on a field of a formatted display, cause the
MDT bit (bit 7) of the field attribute for the field to be set to 1.
However, when the cursor appears beneath a character in a protected
field or beneath a field attribute, that position cannot be modified by
keyboard action, and the MDT bit is not set.

One, and only one, cursor is displayed on the display surface.  When the
display is turned on, the cursor is in the first location on the screen.
The cursor can be repositioned by the keyboard operator and also by the
application program.  The cursor is not affected by field attributes; it
is displayed even when positioned in a nondisplay/nonprint field.


## Graphic Data


Alphabetic characters can be entered into the display buffer in either
uppercase or lowercase code, depending on the position of the shift key.
Only uppercase alphabetic codes can be entered from some keyboards.

Keyboard entry of an alphanumeric character into the buffer occurs at
the cursor location, provided that the cursor is located in a character
location within an unprotected field.  (An attempt to enter an
alphanumeric character into a protected field or into a field attribute
location is blocked.)  Successful keyboard entry of the alphanumeric
character causes the cursor to advance to the next character location
within the unprotected field.

## Automatic Skip

Upon entry of a character into the last character location of an unprotected field, the cursor is repositioned according to the field attribute describing the next field.

If the field attribute defines the next field as (1) alphanumeric and either unprotected or protected, or (2) numeric and unprotected, the field attribute is skipped and the cursor is positioned to the first character location in that field.

If the field attribute defines the field as numeric and protected, the cursor skips that field and is positioned to the first character location of the next unprotected field.

## Erase to End of Field (ERASE EOF) Key

If the cursor is located in an alphanumeric character location in an unprotected field, this key clears the character location occupied by the cursor and all remaining character locations in that field to nulls. Each character attribute associated with the nulled characters is set to its default value.

If the display is unformatted (that is, there are no field attributes on the display), the character buffer from the cursor address to the end of the screen is set to nulls and the corresponding character attributes are set to their default values.

The operation can wrap from the end of the last row of the display to the beginning of the top row. The cursor does not move as a result of operating this key, and the MDT bit is set to 1.

When the cursor is in a field attribute location or is within a protected field, pressing this key locks the keyboard and notifies the terminal operator of an input-inhibit condition. No character locations are cleared, the cursor is not moved, and the MDT bit is not set.

## ERASE INPUT Key

This key clears all unprotected character locations to nulls, resets the MDT bit to 0 in unprotected fields, sets all character attributes to their default value, and repositions the cursor to the first unprotected character location on the display.

In a buffer with only protected fields, no character locations are cleared and the cursor is repositioned to buffer address 0.

If the display contains no field, the entire character buffer is cleared to nulls, the cursor is repositioned to character location 0, and all character attributes are set to their default value.


## Insert Mode (INS MODE) Key


This key turns on the insert-mode indicator on the display (if one exists) and places the display in an insert mode of operation, regardless of the cursor location.

If the cursor is located in an unprotected field that has a null character either in the character location identified by the cursor or in any character location in the field beyond the cursor, pressing an alphanumeric key causes that alphanumeric character to be entered at the cursor position and the MDT bit to be set to 1. The character formerly occupying the cursor location and all remaining characters within the field (except for null characters or characters to the right of null characters) are shifted one character location to the right. If the location identified by the cursor at the time of the insert operation is a null, no character shifting occurs.

The character attributes of any shifted characters do not change; the shifted characters retain their character attributes. Each character attribute associated with the inserted character is set to its default value. In all cases, however, programmed symbol (PS), color, and highlighting select keys (if available on the keyboard, and allowed by the Set Reply Mode structured field) set the attribute values associated with that character as it is stored.

After all null characters at or beyond the cursor location in the field have been overwritten, or if there were no null characters, pressing an alphanumeric key locks the keyboard and notifies the terminal operator of an input-inhibit condition. The field attribute and the extended field attribute remain unchanged. If more than one row of characters is contained within the field, a character occupying the last character location in the row is shifted into the first character location of the next row.

Pressing an alphanumeric key while in insert mode when the cursor is located in a field attribute location, or is within a protected data field, locks the keyboard and notifies the operator of an input-inhibit condition. No character locations are cleared, the cursor is not moved, and the MDT bit is not set.

Operation of the RESET key returns the keyboard to normal mode.

## Delete (DEL) Key

If the cursor is located in an alphanumeric character location in an
unprotected field, pressing the delete key deletes the character from
the character location occupied by the cursor and sets the MDT bit to 1
(if it has not been set previously). The cursor does not move. All
remaining characters in the unprotected field (to the right of the
cursor and on the same row) are shifted one character location to the
left. Vacated character locations at the end of the row are filled with
nulls. If the unprotected field encompasses more than one row,
characters in rows other than the row identified by the cursor are not
affected.

The character attributes of any shifted characters do not change; the
characters retain their character attributes. Each character attribute
associated with a vacated character location is set to its default
value.

Operation of this key when the cursor is located in a field attribute
location or is within a protected field disables the keyboard; no
character locations are cleared, the cursor is not moved, and the MDT
bit is not set.

## Duplicate (DUP) Key

Operation of this key causes a unique code to be entered into the
buffer, a Tab key operation to be performed, and the MDT bit to be set
to 1. The operator uses the duplicate key to tell the application
program that a duplicate operation is indicated for the rest of the
field in which it is located. The character transferred to the
application program is a X'1C' (EBCDIC) and is sent when the data is
read. No duplicate operation is performed at the display. The
duplicate character, when stored in the character buffer, is displayed
as asterisk-overscore. Display devices operating in monocase mode
display this character as an asterisk.

Operation of this key when the cursor is positioned at a field attribute
location or is within a protected field disables the keyboard; no
character locations are cleared, the cursor is not moved, and the MDT
bit is not set.

## FIELD MARK Key

Operation of this key causes a unique code to be entered into the
character buffer and the MDT bit to be set to 1. The field mark
character informs the application program of the end of a field in an
unformatted buffer or subfield in a formatted buffer. The field mark
character transferred to the application program is X'1E' (EBCDIC) and

is sent when the data is read.  The field mark character, when stored in
the character buffer, is displayed as semicolon-overscore.  Display
devices operating in monocase mode may also display the FIELD MARK key
as a semicolon.

Operation of this key when the cursor is positioned at a field attribute
location or is within a protected field disables the keyboard; no
character locations are cleared, the cursor is not moved, and the MDT
bit is not set.


## Program Attention Keys


These keys solicit application program action by causing the sending of
an attention identification (AID) character to identify the key.  The
MDT bit is not affected.  The program attention keys are CLEAR, ENTER,
CANCEL, all program function (PF) keys, and the program access (PA)
keys.  Operation of the CLEAR key also causes the entire buffer to be
cleared to nulls and positions the cursor to character location 0.

Note: Not all program attention keys are available on all types of
keyboards.


## Clear Partition Key


If the display is in explicit partitioned state when the operator
presses the Clear Partition key, the viewport and the character buffer
(including field attributes) of the active partition are set to nulls.
This includes setting extended field attributes and character attributes
associated with the partition to their default values.

The AID for the Clear Partition key is different from the AID for the
CLEAR key.  This difference allows the host to distinguish between
CLEAR, which restores implicit partitioned state, and Clear Partition,
which keeps the device in explicit partitioned state.  If the
transmission is from explicit partition 0 (PID = 0), the AID is the only
way the application program can distinguish between them.


## CLEAR Key


If the display is in implicit partitioned state, the CLEAR key clears
the character buffer to nulls and resets partition 0 to default size.

If the display is in explicit partitioned state, the CLEAR key resets
the display to implicit partitioned state by (1) destroying all existing
partitions and (2) creating implicit partition 0 (PID = 0) with a
default size.  The character buffer of implicit partition 0 is cleared
to nulls.

## Jump Key

If the display is in explicit partitioned state, pressing the jump key causes the cursor to jump from the active partition to the next partition. If the PID of the active partition is N, the next partition is that partition with the smallest PID greater than N. If no such partition exists, the next partition is that partition with the lowest PID. Jumping from one partition to another causes:

- The active partition to be made inactive.

- The new partition to be made active.

- The cursor to be displayed at the current cursor position of the new active partition.

- Any "pending" input-inhibit conditions (that is, PWAIT or SYSTEM LOCK) that occurred while the partition was inactive are raised, and the appropriate indicator is displayed.

## KEYBOARD ACTIONS WITH ATTRIBUTE SELECTION KEYS

For those keyboards that do not have attribute selection keys for character sets, color, and highlighting, pressing an alphanumeric key causes the MDT bit to be set to 1 and the alphanumeric character to be entered into the display's character buffer at the position indicated by the cursor. The keyed character is assigned default values for programmed symbols, color, and highlighting.

For keyboards with keys that allow the operator to select or change the character attribute without application program interaction, the default settings are obtained as stated in the previous paragraph. Keying sequences are defined, however, that allow character sets, color, and highlighting attribute values to be selected by the operator when enabled by the application program.

Once these attribute values have been selected, pressing an alphanumeric key causes the alphanumeric character to be entered into the character buffer at the cursor position and the MDT bit to be set to 1. In addition, the character attribute associated with the entered character is modified by the attribute values selected by the operator (if operator alteration of the character attribute is allowed by the Set Reply Mode structured field). The operator's selections apply until changed, or until the keyboard is returned to the default setting.

KEYBOARD ACTIONS IN PARTITIONS

As the operator enters data, the current cursor position is incremented
by 1 for each character entered, and the cursor is displayed in the new
character location.  This cursor movement may cause automatic scrolling.

The operator may also move the screen cursor by using the Up, Down,
Left, Right, Backspace, Tab, Backtab, Skip, or New Line keys.  The Up,
Down, Left, Right, and Backspace keys move the screen cursor within the
viewport of the active partition and cause the current cursor position
to be updated.  Thus, the screen cursor wraps at the viewport
boundaries.

Data-entry keystrokes and the Tab, Backtab, Skip, and New Line keys move
the current cursor position within the character buffer of the active
partition.  Where the buffer is larger than the viewport, automatic
scrolling may occur.

If a character buffer contains at least one unprotected field, the first
unprotected character location in the buffer is termed the home
position. If the buffer contains no unprotected fields, the home
position is defined to be character location zero.

The Home key resets the current cursor position in the active partition
to the home position and causes the cursor to move to this position
within the viewport.  This may cause automatic scrolling to occur.

The effect of the Delete, ERASE EOF, and ERASE INPUT keys is constrained
to the active partition; that is, no deletions or storing of nulls
occurs outside the active partition.  The Clear Partition key clears the
active partition, which is active only when the device is in the
explicit partitioned state.  The CLEAR key clears the entire screen by
destroying all partitions, placing the device in implicit partitioned
state, and re-creating an implicit partition 0 of default size.


SCROLLING PARTITIONS


Different areas of the presentation space can be seen in the viewport by
moving the window on the presentation space.  This technique is called
scrolling. Scrolling within a partition is possible if the presentation
space is larger than the associated viewport.  Scrolling may be done by
the operator using operator keys or by the host using a structured
field.

When a partition is created, it can be set up as a scrollable partition
by making the buffer larger than the viewport.  The viewport seen by the
operator has a one-to-one relationship with a window on the
corresponding presentation space.  The position of the window on the
presentation space determines what data is seen in the viewport.  The
initial position of the window on the presentation space is specified by

the Create Partition structured field. The position of the window on the presentation space is re-set using the Set Window Origin structured field.

Row-by-row vertical scrolling within the presentation space is provided in response to the operator's pressing the Scroll Up and Scroll Down keys. Vertical scrolling is achieved by moving the position of the window within the presentation space by one line in response to the use of the scroll keys. (The action of the scroll keys is defined in more detail below.)

Data streaming is used to or from the character buffer, and, hence, addresses (such as in SBA) are relative to the origin of the buffer.


## Scroll Keys


The operator interacts with the system through information displayed in the viewport. The cursor is always displayed within the viewport. Whenever an operator keystroke (of data or the field-oriented keys) causes the cursor to leave the viewport, there is an automatic scroll. Conversely, when a scrolling operation moves the window so that the cursor would no longer be within the viewport, the cursor is dragged along at the edge of the window and hence remains within the viewport.

The cursor movement keys (Up, Down, Left, Right, Backspace) wrap at the boundary of the viewport. The actions for data keystroking and the field-oriented keys are not affected by scrolling, that is, they wrap at the boundary of the character buffer. The wrap at the buffer boundary may cause an automatic scroll.


## Vertical Scrolling


The Scroll Up and Scroll Down keys are used to control vertical scrolling.

Assuming that the window is not already positioned at the bottom of the presentation space, the action in response to the Scroll Up key is to move the window down the presentation space by one row. The effect, as seen by the operator, is to move the data up the viewport by one row.

The previous top row is lost from the window; all other rows are moved up by one row, and the next row below the window becomes the bottom row of the window.

If the current cursor position is in the top row of the window, the cursor is moved down the presentation space by one row so that it remains in the top row of the window. The effect seen by the operator is that the cursor remains on the edge of the viewport.

5-18

If the window is already positioned at the bottom of the presentation space, then Scroll Up has no effect. Note that there is no input-inhibit condition and no indicator.

Assuming that the window is not already positioned at the top of the presentation space, the action in response to the Scroll Down key is to move the window up the presentation space by one row. The effect, as seen by the operator, is to move the data down the viewport by one row.

The previous bottom row is lost from the window; all other rows are moved down by one row, and the next row above the window becomes the top row of the window.

If the current cursor position is in the bottom row of the window, the cursor is moved up the presentation space by one row so that it remains in the bottom row of the window. The effect seen by the operator is that the cursor remains on the edge of the viewport.

If the window is already positioned at the top of the presentation space, then Scroll Down has no effect. Note that there is no input-inhibit condition and no indicator.


Keyboard Actions and Scrolling


The character-oriented keys (Up, Down, Left, Right, and Backspace) wrap the cursor at the viewport boundary. The field-oriented keys (Tab, Backtab, Skip, and New Line) and automatic skip operate on the entire presentation space and wrap at its boundary. If the resulting current cursor position is outside the window, there is an automatic scroll.

The Home key positions the cursor at the home position within the presentation space. This key may cause an automatic scroll.


Action for Data Entry Keystrokes


All data entry keystrokes cause normal incrementing of the cursor position, including a wrap at the boundary of the presentation space. If the resulting cursor movement places the cursor outside the window, there is an automatic scroll.

The ERASE EOF, ERASE INPUT, Clear Partition, CLEAR, and Delete keys, and the use of the Insert Mode key, can cause changes to the character positions anywhere in the presentation space; in particular, such changes can be outside the viewport.

## Automatic Scroll

An automatic scroll occurs whenever a field-oriented key, a data entry keystroke, the Home key, or an IC order results in a current cursor position that is outside the window.

The window is moved by the minimum number of lines necessary to put the current cursor position in a peripheral row of the window. The effect is that the cursor always remains within the window, and hence within the viewport.

## SELECTOR-PEN OPERATION

The selector pen is a light-sensitive pen that can detect the light emitted from characters displayed on the display surface. With the selector pen, the operator can select from a list or table of displayed items and can then cause those selections to be identified to the application program.

## SELECTOR-PEN FIELD FORMAT

A field that is to be used for selector-pen operations must be defined in the following format:

```
                        ┌─────────────┐    The field attribute
                        │ Field       │    defines the field as displayed
                        │ Attribute   │    and selector-pen-detectable.
                        │             │    (A field may be protected or
                        │             │    unprotected, alphanumeric or numeric.)
                        │─────────────│
Selector-               │ Designator  │    The designator character
Pen-                    │ Character   │    defines the type of operation
Detectable              │             │    that will be performed by
Field                   │             │    detection on this field.
                        │─────────────│
                        │             │
                        │ Displayed   │    One or more displayed
                        │ Data        │    alphanumeric characters for
                        │             │    sensing by the selector pen.
                        └─────────────┘
```

## DESIGNATOR CHARACTERS

Designator characters are used to define two types of selector-pen fields: selection fields and attention fields. Each type of field performs a different selector-pen operation. If the field attribute

defines a field as selector-pen-detectable but the character in the designator character position is not a valid designator, a detect cannot be made on the field.


## Selection


The selection field is defined by a question mark (?) designator character. When the selector pen detects on a selection field, the MDT bit in the field attribute for the field is set to 1. Also, the designator character is automatically changed in the character buffer and on the display surface to show to the operator that the detection was successful. In summary:

| If Before Detect: | | Then After Detect: | |
|---|---|---|---|
| Designator | MDT | Designator | MDT |
| ? | 0 | > | 1 |
| ? | 1 | > | 1 |

If the designator is > and a detect is made on the field, the designator is changed to ? and the MDT bit is set to 0. In summary:

| If Before Detect: | | Then After Detect: | |
|---|---|---|---|
| Designator | MDT | Designator | MDT |
| > | 0 | ? | 0 |
| > | 1 | ? | 0 |


## Attention


The attention field is normally defined by a space or null designator character. In addition, an implementation may support the ampersand (&) designator character to provide an ENTER key simulation. Note that, when the selector pen is implemented, all the selector-pen functions are mandatory except the & designator.

A detection on an attention field designated by a space or null causes an AID to be sent to the application program which identifies the source of entry to be the selector pen. Also sent are the addresses of fields with the MDT bit set to 1 and the current cursor position for the partition that contains the selected field. The application program responds, usually by issuing a Read Modified All, to obtain the contents of modified fields and the addresses.

A detection on an attention field designated by an ampersand results in an AID's being sent which identifies the source of entry to be the Enter key. Also sent are the addresses and contents of all fields with the MDT set to 1 and the current cursor position for the partition that contains the selected field. If the application defines an attention field with an ampersand designator and the implementation does not

support the ampersand, the selector pen is inoperative on the field; the operator must then use the ENTER key.

Note: The application programmer should be aware that high-intensity/unprotected fields can be modified by the operator to become selector-pen-detectable fields.


SELECTING FIELDS IN PARTITIONS


The operator may use the selector pen to select fields in any partition, whether it is active or inactive. Selection of a field in an inactive partition causes no movement of the cursor on the display surface and the partition containing the selected field does not become active, but the operator may notice a change in the indicators. The operation performed depends on the designator character selected.

A selection outside the viewports is ignored.

Selection of an immediate detect field (designator of blank, null, or ampersand) causes data to be transmitted. This data includes the addresses of all the modified fields and the current cursor position for the partition that contains the selected field. Note that this need not be the active partition.

## CHAPTER 6. PRINTER CONSIDERATIONS

## PRINTERS

Although oriented toward a display with a keyboard, the preceding 3270 data stream descriptions apply, with certain exceptions, to a printer without a keyboard. The exceptions are the following:

- Commands. The read commands are not valid in the 3270 data stream received by the printer in an SNA environment. If received, the request is rejected (sense code X'1003'). See also the Structured Field exception. However, the read commands are valid if received in the 3270 data stream in a non-SNA environment for locally attached devices (3272 version) or in a BSC environment.

- Structured Fields. Structured fields that are not supported or supported structured fields with nonsupported values are rejected (sense code X'1003').

   - The Create Partition, Activate Partition, and Set Window Origin structured fields are not supported.

   - The Read Partition structured field is limited to query operations; the Read Modified command is supported only for retry of the query.

- Keyboards. The data stream sent to the application program is limited to SNA responses and to DFC requests, for example, logical unit status (LUSTAT).

- Selector pen. Not applicable to printer.

- WCC byte. The WCC byte definition for printer use is shown in Figure 6-1.

- Format-control orders. Unlike displays, the NL, CR, EM, and FF orders provide a print format function when received by a printer, as follows:

   - NL (New Line). Moves the print position horizontally to the left margin and vertically down to the next line.

   - CR (Carriage Return). Moves the print position horizontally to the left margin.

| Bit | Explanation |
|-----|-------------|
| 0&1 | The function of these bits is to make the WCC byte an EBCDIC/ASCII-translatable graphic. In an SNA/EBCDIC environment, the bits are ignored by the printer. In an SNA/ASCII environment, the bits are set in accordance with Figure D-1 in Appendix D. |
| 2&3 | Defines printout format as follows: 00 - The NL or CR orders in data stream determine print line length, and EM designates the end of the message. Provides a 132-character print line when orders are not present. 01 - Specifies 40-character print line 10 - Specifies 64-character print line 11 - Specifies 80-character print line |
| 4 | Start-printer bit. When set to 1, initiates a printout operation at the completion of the write operation. |
| 5 | Sound-alarm bit. When set to 1, sounds the audible alarm if an audible alarm provided. |
| 6 | Keyboard-restore bit. |
| 7 | Reset MDT bit. |

**Figure 6-1. The WCC Byte (As Defined for Use with Printers)**

- EM (End of Message). Terminates the print operation.

- FF (Form Feed). Moves the print position to the top and left margin of the next page.

In normal operation, when bits 2 and 3 of the WCC are 0, a new-line function is performed each time a valid NL character is encountered. In addition, if no valid NL is encountered before the printer reaches the end of a line (as determined by the maximum physical carriage length), the printer automatically performs NL and continues printing.

During a print operation, if line-length format is specified in bits 2 and 3 of the WCC (bits 2 and 3 not equal to 0), data characters in the printer buffer are scanned one line at a time before they are printed. A line feed is executed after each line is printed. If a line contains only null characters and one or more space characters, a line feed is performed to cause a blank line in the printout. When null characters, field attributes, or alphanumeric characters in a nonprint field are encountered, they are treated as follows:

- If in a line that contains another print field, they are printed as spaces.

- If they constitute an entire line, they are ignored, and the line feed is not performed; as a result, a blank line does not appear in the printout, and the data is compressed vertically one line.

When line-length format is not specified, printout of the buffer begins at buffer address 0 and continues until the last position of the character buffer is printed or until a valid EM character is encountered. Each print line is left-justified. At the end of each printout, a final New Line is executed so that the printer is ready to start the next printout. When the print-terminating EM order appears in the first print position of the print line, a final New Line is not executed, because the printer is already positioned at the left margin for the next printout.

The validity requirements for NL, CR, EM, and FF are as follows:

- NL, CR, and EM are valid only when encountered in a print field during a printout that does not have a line-length format specified by the WCC.

- FF is valid in any buffer position.

When the printer supports vertical forms control (VFC), an FF causes the form to index to a predetermined line, and the first print position (the buffer location containing the FF character) is printed as a space character. If VFC is not supported, the FF is invalid.

The rules for NL apply to CR. However, the printer must support CR at least to the extent of accepting and printing it as a space. Invalid NL, CR, EM, and FF are not executed and print as spaces.

- Data Integrity. Since printers in an SNA environment do not support a read operation, the data integrity requirement on returned data is not applicable. Where the requirement says must be returned as sent, for printers it becomes must be accepted.

Because printers in a BSC environment or in a non-SNA environment for locally attached devices can be read by the application program, these printers are subject to the same data integrity requirements as displays.


## LOCAL-COPY FUNCTION IN AN SNA ENVIRONMENT


A hard copy of the display screen or a portion of a display screen may be accomplished by "host" application use of the conventional display read function and the printer write function. However, to simplify the host application and reduce line traffic, a specific copy function is

provided that accomplishes a transfer of data to a printer without
routing through the host application.

Whether initiated by the host or by the operator, the resultant hard
copy is a replica of the screen or portion of the screen, with the
following exceptions;

*   Display lines containing all nulls are suppressed.

*   Mismatch of uppercase and lowercase may occur.  When the configured
    printer(s) has the capability for selection of Mono-Dual case
    without requiring operator action, the printed data must match the
    case of the displayed data.  It is not necessary for the printer to
    recognize any manual override by the display of uppercase and
    lowercase.

*   Color may not match.

The local-copy operation is described on the basis of a nonpartitioned
screen, that is, for a display that does not have partitioning
capability or for a display with partitioning capability that is in
implicit partitioned state (implicit partition 0).  Differences due to
partitioning are described under "Partition Mode Considerations."


COPY INITIATION


The host initiates a local copy by sending the display a write-type
command with the start-print (SP) bit set to 1 in the WCC byte.  If the
write-type command includes data, the screen will be updated prior to
print execution.  Once a copy request is accepted, the display keyboard
remains locked until the printer completes the print operation.  The
host is required to send the Copy request either as an RQD chain or as
an RQE, CD, not-EB chain; this prevents the host from following a copy
request immediately with another command.


PRINTER AVAILABILITY


When more than one printer is authorized for a display's use, a Copy
request could result in the printout on any of the authorized printers.
From the host view, however, there is only one logical printer,
regardless of how many physical printers are authorized for the
display's use; the host application has no control over the printer
selected.  The user must predefine (for example, IML) the printer(s)
configuration.  A printer(s) must be capable of being configured for
system use only (that is no copy use), for copy use only, or for shared
use between the system and copy.

A logical printer is considered available if one or more of the physical
printers, configured for the display's use, can immediately execute a

6-4

display printout. When none of the configured printers is available, the logical printer assumes the status of the most available physical printer(s). The following are the not-available categories listed in order of best to worst:

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

b.  Intervention-required condition.

c.  Allocation for session use or local use (not copy).

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

e.  No logical printer configured.

The display will reject a copy request whenever the logical printer is not available. The following are the sense codes used:

•  082E. The logical printer has an intervention-required condition, for example, out of paper, power off, and cover interlock open. This code is used if the printer is attended. In general, attended means the condition will likely receive quick attention; for example, the printer is located near a display. If the printer is unattended, the permanently unavailable code should be used.

•  0807. The logical printer(s) is busy for an indeterminate period of time. This may range from a relatively short time when the printer(s) is being used by another display for a display printout to many hours if all printers are in session. The display should mask short term busy from the host by withholding the -rsp (0807).

•  082F. Effectively, the logical printer is permanently not available; for example, a hardware failure.

•  0801. No printer configured.

After sending an 082E or 0807, an LUSTAT must be sent when the condition clears. However, regardless of how many times the request is repeated, only one LUSTAT is sent when the condition does clear. Sending an LUSTAT 0001 (with source=printer) indicates to the host that a Copy request will find either an available or short-term-busy logical printer. Exceptions are as follows:

•  The LUSTAT 0001 (source=printer) reporting the clear of a 082E or 0807 condition is not sent if either:

    After sending 082E/0807 -rsp, the Data Traffic Subtree is reset or the session is terminated.

Or

After sending 082E/0807 -rsp, the logical printer develops a
permanent error or becomes not configured.  An LUSTAT 081C/0801
(source=printer) is sent instead of LUSTAT 0001.

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

After sending 082E/0807 -rsp, the SLU receives any normal FM data
request chain other than a Copy request chain.

Or

After sending LUSTAT 0001 (source=printer), the logical printer
develops an intervention-required, permanent-error, or
not-configured condition.  The Copy request will be rejected with
the appropriate -rsp, that is, 082E, 082F, or 0801.

Or

It is not valid for an implementation to allow session contention
for a printer prior to honoring outstanding LUSTATs; that is, at
least one physical printer must be held after having sent the
LUSTAT(s) so that the logical printer is available to execute copy
sessions.  However, if as a result of an exception condition
developing on the held printer the logical printer status of in
session results, then 0807 -rsp may also be sent.

Or

After sending 082E/0807 -rsp, the display detects a nonprinter
exception condition which will cause a -rsp to a received request.

The sense codes 082E, 0807, 082F, and 0801 indicate to the host that, if
the copy request included screen update data, the screen update was
accomplished.  If a copy request (with update data) is re-sent, it
cannot be guaranteed that the screen will be unchanged.  An example
where the screen would be changed is if the data used a positioning
reference, for example, a Program Tab order that was revised later in
the data stream.


DISPLAY/PRINTER COMPATIBILITY


In general, a print operation will not be inhibited because of a
mismatch in display/printer capability.  However, a copy operation may
be inhibited if the printer is unable to accept all the display data,
for example, print buffer too small, printer maximum line length less
than display width.  When the copy is inhibited because the printer
cannot accept all the display data, the printer is considered to be not
authorized.

APL MISMATCH

Where the display has the APL feature, but the printer to which the copy
is directed does not have APL, the copy operation will occur using the
nonloadable character set. There will be no indication of this
mismatch. Such mismatches can be avoided by the proper configuration of
copy printers.


CHARACTER ATTRIBUTE/EXTENDED FIELD ATTRIBUTE (CA/EFA) MISMATCH

On a copy operation the CA/EFA must be sent to the printer if the
printer supports CA/EFA and if the display data to be copied references
nonzero values of CA/EFA. When not all the extended functions are
referenced in the copy data, an implementation may send only the
required information or may send all the CA/EFA information. For
example, if only extended color were referenced in the display data,
that is, no character set, or extended highlighting selected, an
implementation may send only the color information from the CA/EFA or
all the CA/EFA information.


PROGRAMMED SYMBOLS (PS) CONSIDERATIONS

A copy operation will not be inhibited because of a mismatch of
display/printer character-set capability or character-set load.
Whenever the printer cannot print the data against the same loadable
character set referenced by the display data, the print is executed
against the nonloadable character set of the printer.

The Load PS structured field contains a compare field of 1 bit. If set
to compare (bit=0), the LCID may be used in establishing a PS match in a
copy operation; for a valid match, the matching LCIDs must both be set
to compare. If set to no-compare (bit=1), the LCID cannot be used for
establishing a match.

The following summarizes the copy printout operation:

IF      The referenced display PS is set to compare, and the printer has a
        matching LCID also set to compare, then execute printout against
        the matched LCIDs.

ELSE    Execute the printout against the printer default character set.

Note that the printed data could have a mixed appearance, that is, some
matching the displayed data and some not matching the displayed data.

EXTENDED COLOR MISMATCH

A copy operation will not be inhibited because of a mismatch in color
capability between the display and the printer; that is, no check is
made for display/printer compatibility. If the printer does not support
color, display data in color is printed in monochrome. If the printer
supports color but a different set of colors than the display, the
colors not supported are printed in the default color defined by the
printer Query Reply color.


EXTENDED HIGHLIGHTING MISMATCH

Copy will not be inhibited because of a mismatch in extended
highlighting capability between the display and the printer; that is, no
check is made for display/printer extended highlighting capability. If
the display specifies a particular form of highlighting not supported by
the printer, the data is printed with defaults defined in the printer's
Query Reply highlight.


PARTITION MODE CONSIDERATIONS

When the display is in partition mode with or without scrolling, the
copy operation changes as follows:

> An Outbound 3270DS structured field with a WCC=SP initiates a
> printout of the window of the designated partition. If a Write
> Structured Field command contains multiple structured fields, the
> WCC=SP may be set only in the last structured field. If set in any
> other structured field, the Write Structured Field is rejected with
> sense code X'1001'. The printout of the partition viewport has the
> same width as the display partition and is left-justified. If the
> display partition width exceeds the printer line length, the copy
> operation is rejected with sense code X'0801', no printer
> configured.
>
> If a Write, EW (reset off), or an EWA (reset off) is sent with
> WCC=SP to a partitioned screen, a printout of partition 0 will
> occur. If partition 0 does not exist, the copy operation is
> rejected with sense code X'1005', parameter error. If an EW (reset
> on) or an EWA (reset on) is sent with WCC=SP to a partitioned
> screen, the screen is reset to implicit partition 0.


LOCAL-COPY COMMAND IN THE BSC ENVIRONMENT

In BSC, the local-copy function is accomplished by use of the Copy
command. This Copy command is used to transfer buffer data from one

terminal to another terminal attached to the same controlling device. The selected terminal is the to terminal, the one to which the buffer data will be transferred. The from terminal, the source of the buffer data to be copied, is identified in the second two bytes that follow the Copy command code; the first byte, called the copy control character (CCC), identifies the type of data to be copied. The CCC can also, at the to device, start print operations, specify the printout format for those operations, and, when the terminal is a display station, sound the audible alarm.

The copy data stream is shown in Figure 6-2.

Figure 6-3 describes the function of each CCC bit. A CCC and an address byte must always follow the command code. If they do not, the controlling device aborts the command and generates error status.

The from terminal buffer can be locked (made incapable of being copied) by writing a protected/alphanumeric attribute byte (bit 2=1 and 3=0) in address 0.

The Copy command can specify as the from device the same device that is selected (the to device). This procedure provides a means of programming selective device buffer "erase" operations as specified by CCC bits 6 and 7.

Programming Notes:

1.  Copy should not be chained from a Write, Erase/Write, Erase/Write Alternate, or Erase All Unprotected command, since it will copy the data as modified by the Write or Erase command.

2.  If the CCC start-print bit is set and commands are being chained, Copy should be the last command of the chain. If not, the control unit aborts the subsequent command.

3.  Copy can be executed from a smaller buffer to a larger buffer, but an attempt to copy from a larger to a smaller buffer will cause an operation check.

If the Copy command references a from terminal, in implicit partitioned state, that has not received an SFE, SA, MF, or GE, or on which the operator has not entered a character with an extended attribute, or a specific APL/TN character, since the last buffer clear (for example, EW, EWA commands), a copy action takes place. If the from terminal does not satisfy the above criteria, the Copy command will be rejected with an operation check (OC) and unit specify (US) status, unless all the following conditions are met:

1. The from terminal is a display.

2.  The destination terminal is a printer.

**Byte**

| | |
|---|---|
| 0 | STX |
| 1 | ESC |
| 2 | Copy Command Code |
| 3 | CCC ←———— See below and Figure 6-3. |
| 4 | *From* Device Address |
| 5 | ETX |
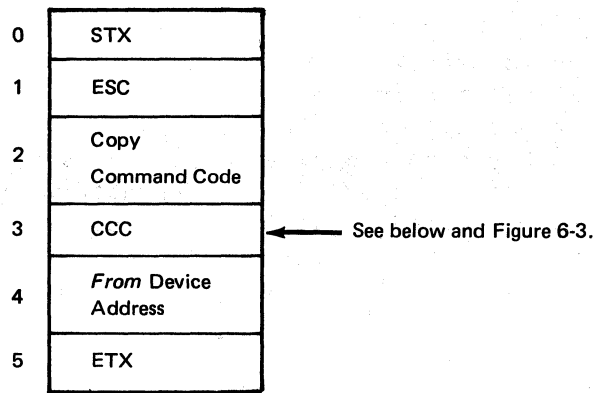
CCC-Byte Format

| * | 1 | Printout Format | | Start Print | Sound Alarm | Type of Data to Be Copied | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

*Determined by the configuration of bits 2 through 7. See Figure 6-3.

**Figure 6-2. The Copy Data Stream**

| Bit | Explanation |
|---|---|
| 0, 1 | Determined by the contents of bits 2 through 7. |
| 2, 3 | Define the printout format as follows: |
| | 00 — The NL, EM, and CR orders in the data stream determine print-line length. Provides a 132-print position line when the orders are not present. |
| | 01 — Specifies a 40-character print line. |
| | 10 — Specifies a 64-character print line. |
| | 11 — Specifies an 80-character print line. |
| 4 | The start-print bit. When set to 1, initiates a printout operation at the *to* device after buffer transfers are completed. |
| 5 | The sound-alarm bit. When set to 1, sounds the audible alarm at the *to* device after buffer transfers are completed if that device has an audible alarm. |
| 6, 7 | Define the type of data to be copied as follows: |
| | 00 — Only attribute characters are copied. |
| | 01 — Attribute characters and unprotected alphanumeric fields (including nulls) are copied. Nulls are transferred for the alphanumeric characters not copied from the protected fields. |
| | 10 — All attribute characters and protected alphanumeric fields (including nulls) are copied. Nulls are transferred for the alphanumeric characters not copied from the unprotected fields. |
| | 11 — The entire contents of the storage buffer (including nulls) are copied. |

**Figure 6-3. Copy Control Character (CCC)**

3. The <u>from</u> terminal does not have a protected/alphanumeric field attribute in the first buffer position.

4. The CCC has bits 4 (start print), and 6 and 7 (copy entire buffer) set to 1.

If these conditions are met, an attempt will be made to produce a local copy. Following print completion, the print buffer will be cleared and the appropriate completion status made available at the printer.

# CHAPTER 7. STRUCTURED FIELDS

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

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

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

 _____
|           |      |             |          |      |             |    |
| length 1  |  ID  | information | length 2  |  ID  | information |    |
|_____|_____|_____|_____|_____|_____|____|

|<--------length 1----------->|<---------length 2---------->|
```
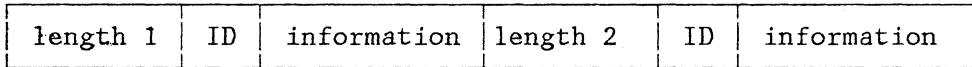
In the 3270 data stream, structured fields are introduced with the Write Structured Field (WSF) command. This command does not contain explicit control information as do other 3270 commands; it simply means all subsequent data is in a structured-field format. A WCC does not follow the WSF command as it does other write commands.

Following the WSF command, all data in the transmission must be in structured-field format. A structured field is of the form shown in Figure 7-1.
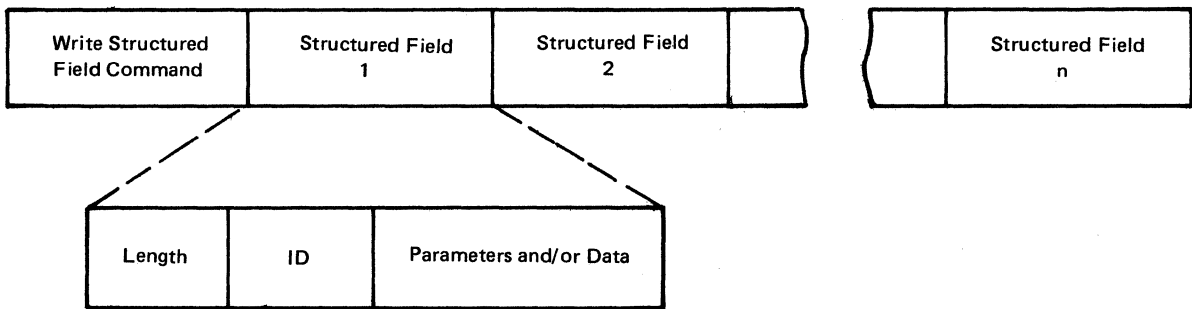


Figure 7-1. Structured Field Format

Each structured field contains a 2-byte-length field. This defines the length of the structured field (including the length bytes), in effect, pointing to the next structured field in the transmission. Next follows a 1-byte ID field that defines the function of the structured field, followed by parameters and data in the format defined by the ID code. A length of zero on an Outbound 3270DS or Inbound 3270DS indicates:

• The length of the structure was not determined.

• This is the last or only structured field in the transmission (for SNA, the transmission = chain).

• The length of the structured field should be determined using the end of the transmission (for SNA, transmission = chain).

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

| ID | Name |
|----|------|
| 00 | Reset Partition |
| 01 | Read Partition |
|    |    02 = Query |
|    |    6E = Read Modified All |
|    |    F2 = Read Buffer |
|    |    F6 = Read Modified |
| 06 | Load Programmed Symbols |
| 09 | Set Reply Mode |
| 0B | Set Window Origin |
| 0C | Create Partition |
| 0D | Destroy Partition |
| 0E | Activate Partition |
| 40 | Outbound 3270DS |
| 80 | Inbound 3270DS |
| 81 | Query Reply |
|    |    81 Query Reply (Usable Area) |
|    |    84 Query Reply (Partitions) |
|    |    85 Query Reply (Character Sets) |
|    |    86 Query Reply (Color) |
|    |    87 Query Reply (Highlight) |
|    |    88 Query Reply (Reply Modes) |
|    |    8A Query Reply (Field Validation) |

Structured-field type codes not listed are rejected with a sense code X'1003'. Unless specifically stated to the contrary, any bits/fields classified as reserved must be checked for zero value; nonzero values are rejected (sense code X'1003').

## ACTIVATE PARTITION

FUNCTION

Activates a partition.

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | X'0004' | Length of this structure |
| 2 | X'0E' | Activate partition |
| 3 | PID | Partition identifier (X'00' through X'7E') |

OPERATION

1. If the named partition does not exist, the structured field is rejected. (See Figure B-1 in Appendix B.)

2. Input-inhibit conditions are handled as follows:

   a. System-lock and PWAIT conditions are saved for the active partition. Their indicators are removed.

   b. Security-key, machine-check, program-check, and message-pending indicators remain displayed.

   c. All other input-inhibit conditions are reset.

3. The named partition is made active.

4. The screen cursor is displayed at the current cursor position of the named partition.

5. Any saved input inhibit conditions for the partition are raised, and the appropriate indicators displayed.

## CREATE PARTITION

FUNCTION

Creates a new partition.

FORMAT

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 0-1 | | L | Length of this structure |
| 2 | | X'0C' | Create partition |
| 3 | | PID | The identifier for the new partition (X'00' through X'7E') |
| 4 | 0-3 | RES | Reserved - must be zero |
| | 4-7<br>X'0'<br>X'1'<br>other | A-MODE | Address mode<br>Self-describing (12/14-bit)<br>16-bit binary<br>Reserved |
| 5 | | RES | Reserved - must be zero |
| 6-7 | | H | The height of the presentation space |
| 8-9 | | W | The width of the presentation space |
| 10-11 | | RV | The y-origin (row origin) of the viewport relative to the top edge of the usable area |
| 12-13 | | CV | The x-origin (column origin) of the viewport relative to the left side of the usable area |
| 14-15 | | HV | The height of the viewport in rows |
| 16-17 | | WV | The width of the viewport in columns |
| 18-19 | | RW | The y-origin (row origin) of the window relative to the top edge of the presentation space |
| 20-21 | | RES | Reserved - must be zero |
| 22-23 | | RS | The number of units to be scrolled in a vertical multiple scroll |

Notes: The above parameters may be progressively included by specifying the appropriate length of the structured field.

The defaults for parameters not included are:

| Parameter | Default |
|-----------|---------|
| PID | X'00' |
| Address mode | X'0' |
| H | Default screen height specified in BIND SESSION |
| W | Default screen width specified in BIND SESSION |
| RV | X'0000' (top screen row) |
| CV | X'0000' (leftmost screen column) |
| HV | H min of H and height of usable area |
| WV | W min of W and width of usable area |
| RW | X'0000' (top presentation space row) |
| RS | X'0001' |

OPERATION

The following steps define the operation:

1. If the display is in the implicit partitioned state, the implicit partition is destroyed and the screen size is set to the value defined in the Query Reply (usable area) structured field. (See "Destroy Partition.")

2. The partition with identifier equal to PID (if it exists) is destroyed. A new partition with identifier equal to PID is created.

3. A character buffer is allocated and cleared. The buffer is assigned WxH positions. If insufficient resources exist in the device, an error occurs. (See Figure B-1.)

4. The viewport of the new partition is assigned an origin (RV,CV) relative to the usable area, width WV and height HV. If any of the following conditions exist, the structured field is rejected (see Figure B-1):

   a. WV = 0

   b. HV = 0

c.  RV+HV >  screen depth

d.  CV+WV >  screen width

e.  The viewport overlaps another viewport

f.  The viewport cannot be contained within the usable area

5.  The current buffer address for the new partition is set to 0.

6.  The current cursor position for the new partition is set to 0.

Notes:

1.  If any of the errors identified above occurs during the processing, a sense code is returned.  (See Figure B-1.)

2.  Any explicitly created partitions are destroyed by BIND, UNBIND, ACTLU, DACTLU, ACTPU, and DACTPU.  (See "Destroy Partition.")

3.  Partition bytes other than 0 are reserved and will be rejected with a sense code of X'1003'.

## DESTROY PARTITION

FUNCTION

To destroy a named partition and, if this is the only existing partition, to create implicit partition zero.  Destroy means to (1) remove the PID from further accessibility, (2) prohibit any further display from the character buffer that had been assigned to the PID until that buffer space is erased or overwritten, and (3) clear the displayed viewport associated with the PID.

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | X'0004' | Length of this structure |
| 2 | X'0D' | Destroy partition |
| 3 | PID | The partition identifier |

OPERATION

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

2. If the named partition is the only existing partition:

   a. Reset any input-inhibit conditions.

   b. Destroy the named partition.

   c. Reset the display to implicit partitioned state and create the default partition 0.

3. If the named partition is not the only existing partition:

   a. If the partition is active, activate the next partition.

   b. Destroy the named partition.


INBOUND 3270DS


FUNCTION


Transmits inbound (from the terminal to the application program) orders and data from a named partition.


FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | L | Length of this structure |
| 2 | X'80' | Inbound 3270DS |
| 3 | PID | Partition identifier (X'00' through X'7E') |
| 4 | AID | Attention identifier |
| 5-6 | CURSOR | Cursor address |
| 7-n | DATA | Orders and data |

OPERATION

This structured field is transmitted inbound:

1. In reply to a Read Modified, Read Modified All, or Read Buffer command when the current partition has a nonzero PID.

2. In reply to a Read Partition structured field, type X'F6', X'6E', or X'F2', addressed to a partition with a nonzero PID.

3. For an operator enter action, when the active partition has a nonzero PID. Orders and data from both implicit and explicit partition zero are transmitted in 3270 format, not in this structured field.


LOAD PROGRAMMED SYMBOLS (LOAD PS)


FUNCTION


Defines additional coded graphic character sets, coded graphic symbols, or type fonts, and then loads that symbol and the character-definition data into a specified PS Read/Write storage (RWS) area in the device.

This structured field causes one or more characters to be loaded into contiguously addressable slots in the specified RWS. An RWS contains 191 contiguously addressable slots, which are associated with data stream code points X'40' through X'FE'. The slot associated with X'40' cannot be loaded and contains a blank.

FORMAT

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 0-1 | | L | Length of this structure |
| 2 | | X'06' | Load Programmed Symbols (Load PS) |
| 3 | | FLAGS | |
| | 0 | BASIC/EXT.<br>B'0'<br>B'1' | Basic or extended form<br>   Basic form: only bytes 0 through 6 present<br>   Extended form: length determined by byte 7 |
| | 1 | CLEAR<br>B'0'<br>B'1' | Clear PS RWS<br>   Do not clear PS RWS.<br>   Clear the specified PS RWS prior to<br>   loading the character data in this<br>   structured field. |
| | 2 | SKIP<br>B'0'<br>B'1' | Skip Suppress (See Note 2)<br>   Suppression not required.<br>   Suppression required. |
| | 3 | RES | Reserved - must be zero |
| | 4-7 | <br>X'0'<br>X'1'<br><br><br><br><br><br><br>X'2'<br>X'5'<br>X'6'<br>others | PS data format type:<br>   Reserved<br>   18-byte form: the first 2 bytes contain a<br>   16-bit vertical slice: the following 16<br>   bytes contain 8-bit horizontal slices.<br>   For a 9x12 cell, the last 4 bytes contain<br>   binary zeros, as do the last 4 bits of<br>   the 16-bit vertical slice.<br>   Type X'1' compressed<br>   Column loading (from left to right)<br>   Type X'5' compressed<br>   Reserved |
| 4 | | LCID | Local character set ID<br>   Use X'40' to X'EF'.<br>   Use X'FF' to indicate that RWS associated<br>   with this LCID is free (not assigned).<br>   Other values are reserved. |
| 5 | | CHAR | Beginning code point<br>   X'41 through  X'FE' |
| 6 | | RWS | PS set RWS number (X'02' through X'07') |

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

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 7 | | P LENGTH | Length of parameters for extended form, including the length parameter. The parameters defined below are progressively included by specifying the appropriate length. Omitted parameters are assumed to have the defined default indicated by the zero value for that parameter. |
| 8 | 0 | APA<br>B'0'<br>B'1' | All points available (APA)<br>   APA<br>   Not APA |
| | 1 | CB<br>B'0'<br>B'1' | LCID compare bit<br>   LCID compare<br>   No LCID compare |
| | 2 | OB<br>B'0'<br>B'1' | Operator-selectable by PS key<br>   operator-selectable<br>   not operator-selectable |
| | 3-7 | RES | Reserved - Must be zero |
| 9 | | LW | Number of X-units in character cell |
| 10 | | LH | Number of Y-units in character cell |
| 11 | | RES | Reserved - must be zero |
| 12 | 0-4 | Res | Ressrved - must be zero |
| | 5-7 | COLOR<br>B'000'<br>B'001'<br>B'010'<br>B'100'<br>other | Color planes<br>   All planes<br>   Blue plane<br>   Red plane<br>   Green plane<br>   Reserved |

Notes:

1. Values for parameters other than those shown above are rejected invalid parameter values are also rejected. (See Figure B-1.)

2. The "lost" space resulting from setting of the skip suppress to B'1' is made up following the last display/print line; that is, the last display/print line moves up from the bottom of the physical presentation space. For a printer, setting of skip suppress to

b'1' must not result in loss of forms sync following a form feed (FF) or forms overflow.

The PS data follows in the format described in bits 3-7 of byte 3.

| Byte | Bit | | Content |
|------|-----|------|---------|
| m-n | | Data | PS data (character-definition data) |

OPERATION

1.  The extended-form bit in byte 3 indicates whether this structured field is of the extended form.  The extended form contains additional information associated with copy operations and color. If the device does not support the extended form and this bit is set to 1, the data stream is rejected with a sense code of X'1005'.

2.  If skip suppression is on, any row on the usable area containing characters from this character set will have the skip suppressed that would normally follow that row.  The characters on the following row will then be vertically adjacent to characters on the current row.  If the skip suppress flag is not set, then no skip suppression is required for characters from this character set.

3.  The RWS number indicates the physical RWS to be loaded.  There is a fixed relationship between the physical RWS number and the attribute selection keys defined for PS.  The value in CHAR indicates the first slot to be loaded, and must be in the range X'41' through X'FE'   If CHAR < X'41' or CHAR > X'FE', the data stream is rejected (invalid parameter, sense code X'1005').

    If the CLEAR flag is set on, all slots in the specified RWS are cleared; then the character definitions in the data portion of the structured field, interpreted according to the data format type (byte 3), are loaded into contiguously addressable slots in the PS RWS, starting at the position defined by CHAR.

    If the data type specifies compressed data (type 2 or 6), the data is first decompressed. (See the Compression Function.)

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

    For data of type 5 or (decompressed) type 6, each set of LH contiguous bits defines one column of a character cell.  The number of LW contiguous columns defines a character as an LW-by-LH matrix of dots, where LW is the width of the character cell and LH is the height of the character cell.  If LW and LH are not supplied in the extended form, or if they are supplied set to zero, the values of LW and LH are determined by the device.

Loading of character definitions continues until either:

a.  The data is exhausted.  In this case the last complete cell
    definition in the data is loaded.  If there are any excess
    bits, they are ignored and a -RSP (with a sense code of
    X'1005') is returned.

    or

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

4.  The LCID specified in this structured field is released from any
    previously associated PS RWS number.  The PS RWS number released is
    assigned LCID = X'FF'.  This is not done until the LCID specified
    in this structured field is assigned to the associated RWS.

5.  The PS RWS number specified in this structured field is released
    from any previously assigned LCID.  Any subsequent reference to
    this released LCID will be an error condition.

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

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

7.  The APA bit, when set to b'1' (not APA), implies that fewer than
    all points may be printed to allow a performance gain for specific
    devices.  For example, 3287 not APA attempts to print all

8.  When the CB bit is set to 0, and a local copy is initiated, the
    LCID of this character set is compared with character set LCIDs in
    the printer to determine whether there is a match.  If the LCIDs
    match and the CB bits are both zero, the copy operation is
    performed using the corresponding LCID in the printer.  If not,
    characters from the nonloadable character set of the printer are
    used.

    When the CB bit is set to 1, the LCID of this character set is not
    compared with LCIDs of character sets in the printer.  The copy
    operation is performed using the base character set of the printer.

9.  When the OB bit is set to 1, it signifies that this character set
    is intended for output only.  Thus, the PS selection key normally
    characters in one head sweep across the print line.  associated
    with the RWS containing this character set cannot be enabled by the
    terminal operator while the key is connected to this LCID.

Selection of individual character sets can thus be disabled, even though the Set Reply Mode structured field allows character set selection. When the Set Reply Mode structured field disables character set selection, selection is disabled for all character sets, independent of the setting of the OB bit for each character set.

10. A character set with triple-plane capability has three color planes into which characters may be loaded. For any code point (X'41' to X'FE') within the character set, each plane may be loaded independently; that is, a different character definition may be loaded into each of the color planes in the character set for that code point.

   For a triple-plane character set, if the COLOR field is B'001', B'010', or B'100', the character-set data is loaded only into the specified character slots in that plane. Other color values are reserved and rejected.

   For a triple-plane character set, if the COLOR field is B'000', the character-set data is loaded into the specified character slots in all three planes.

   For a single-plane character set, if the COLOR field is B'001', B'010', or B'100', the data stream is rejected.

THE COMPRESSION FUNCTION

Symbol definition bit strings can be transmitted by the Load PS structured field function in uncompressed or compressed form. The compression function proceeds in four parts. First, the cell is divided into appropriate pieces called slices, and the slices are further subdivided into 4-bit elements. Second, the elements are compared to determine pattern matching. Third, the elements are grouped into bunches. Fourth, the element comparisons are encoded as a compressed cell definition.

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

An uncompressed symbol definition is created by dividing the character cell within which a symbol is formed into bytes (slices), as shown under the next heading, "Character Cell Divison." The symbol is defined by encoding the bits (dots) in each byte (slice) as B'1' if the dot is to be on, or B'0' if the dot is to be off. The dot pattern representing the symbol is thus formed. Byte (slice) 1 is understood to represent the leftmost upper vertical 8 dots in the display matrix or the leftmost vertical 8 dots in the printer matrix. The string of 144 bits (display) or 80 bits (printer) thus encoded represents the uncompressed symbol

definition. A comparison process, comparing elements (4 bits) in the uncompressed bit string with reference elements selected from the same bit string, is used to compress the data.

## Character Cell Division

The character cell for a display or printer character position is divided into slices as shown in Figure 7-2. A slice corresponds to a byte, the bits to dots.

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

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

As noted, each group of 4 bits is termed an element. The bit string forming the symbol definition is compressed by comparing each element with the corresponding element in a preceding slice or 0, and the compressed bit string is generated according to the matches and mismatches that occur in the comparsion process.

## The Compression Process

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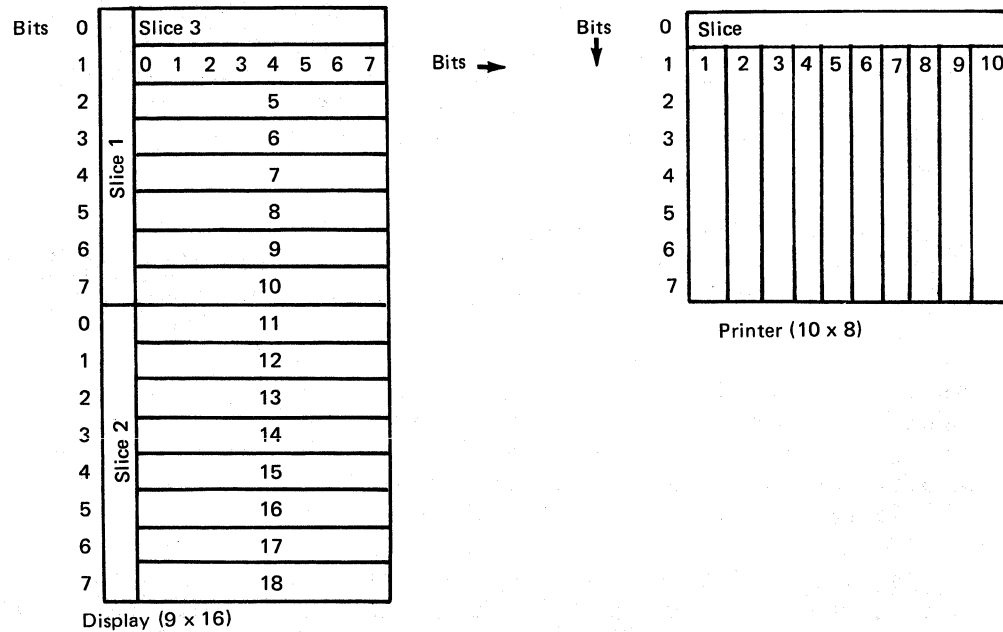


Figure 7-2. Vertical and Horizontal Slicing of a Character Cell

7-14

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

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

H = header bit(s)
SD = symbol defintion bit(s)
T = terminator bit(s)

The following text describes the comparison rules and header bits, the
creating of the compressed bit string, the terminator bits, and examples
of compressing the symbol definition bit strings for the symbol in
Figure 7-3.
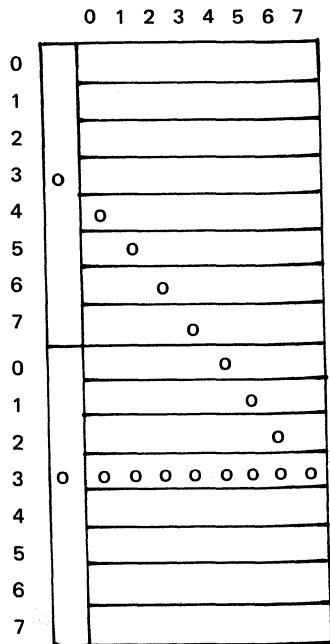


In Type 1 data format, the dot pattern for this example
symbol would be transmitted in 18 bytes having the
following values:

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

Figure 7-3. Type 1 Data Format—An Example Dot Pattern Encoded

## The Comparison Rules and Header Bits

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

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

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

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

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

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

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

- Comparison Rule 4 (Header bits = B'1110")

  No comparison required.  A blank symbol.  The symbol definition
  consisted of 0 bits only.

## Creating the Compressed Bit String

The element comparisons are encoded by taking the elements 4 at a time
(2 slices - the slice-pair referred to in following discussons) and
comparing them with their corresponding elements in a reference
slice-pair that has been created (following the comparison rules).
Because the elements are compared 4 at a time, it is convenient to
regard the 18 slices of the Type 1 data string or the 10 slices of a

Type 5 data string symbol definition as being made up of 9 or 5 slice-pairs.  The elements of the slice-pairs are compressed as follows:

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

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

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

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

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

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

- Step 3
  Repeat steps 1 and 2 in a similar manner through to the 9th slice-pair of a Type 1 string or the 5th slice-pair of a Type 5 string.

## Terminator Bits

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

## Examples of the Compression Algorithm in Use

The following three examples show how a Type 1 data string for a particular symbol is compressed into a Type 2 data string.  In these examples, the symbol whose Type 1 data string is being compressed is the

one shown in Figure 7-3. Here is the data string for that symbol presented as nine slice-pairs:

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

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


Example of Algorithm Using Comparison Rule 1

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

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

| Reference Slice-Pair | Type 1 Slice-Pair | | Element 1 | Element 2 | Element 3 | Element 4 |
|---|---|---|---|---|---|---|
| X'0000' | X'1010' | 1 | 1 0001 | 0 | 1 0001 | 0 |
| X'0000' | X'0000' | 0 | | | | |
| X'0000' | X'0000' | 0 | | | | |
| X'0000' | X'8040' | 1 | 1 1000 | 0 | 1 0100 | 0 |
| X'0000' | X'2010' | 1 | 1 0010 | 0 | 1 0001 | 0 |
| X'0000' | X'0804' | 1 | 0 | 1 1000 | 0 | 1 0100 |
| X'0000' | X'02FF' | 1 | 0 | 1 0010 | 1 1111 | 1 1111 |
| X'0000' | X'0000' | 0 | | | | |
| X'0000' | X'0000' | 0 | | | | |

(Slice-Pairs Being Compared according to Rule 1:)

Figure 7-4. Example of Compression Algorithm Using Comparison Rule 1

With comparison rule 1, the header is B'0' and the symbol-definition bit string is created by comparing each Type 1 slice-pair with an all-zeros reference slice-pair as shown in Figure 7-4. The resultant bit string, including the header, is:

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

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


Example of Algorithm Using Comparison Rule 2

| | | Step 1: Compare slice-pairs.<br>For a match, generate B'0'.<br>For a mismatch, generate B'1' and do step 2. | | | |
|---|---|---|---|---|---|
| Slice-Pairs Being Compared according to Rule 2: | | Step 2: Compare elements.<br>For a match, generate B'0'.<br>For a mismatch, generate B'1' followed by a copy of the bits in the nonmatching Type 1 element. | | | |
| Reference Slice-Pair | Type 1 Slice-Pair | | Element 1 | Element 2 | Element 3 | Element 4 |
| X'0010' | X'1010' | 1 | 1 0001 | 0 | 0 | 0 |
| X'1000' | X'0000' | 1 | 1 0000 | 0 | 0 | 0 |
| X'0000' | X'0000' | 0 | | | | |
| X'0080' | X'8040' | 1 | 1 1000 | 0 | 1 0100 | 0 |
| X'4020' | X'2010' | 1 | 1 0010 | 0 | 1 0001 | 0 |
| X'1008' | X'0804' | 1 | 1 0000 | 1 1000 | 0 | 1 0100 |
| X'0402' | X'02FF' | 1 | 0 | 1 0010 | 1 1111 | 1 1111 |
| X'FF00' | X'0000' | 1 | 1 0000 | 1 0000 | 0 | 0 |
| X'0000' | X'0000' | 0 | | | | |

Figure 7-5. Example of Compression Algorithm Using Comparison Rule 2


With comparison rule 2, the header is B'10' and the symbol-definition bit string is created by comparing each Type 1 slice-pair with a reference slice-pair composed of the previous slices, as shown in Figure 7-5. The resultant bit string including the header is:

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

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

Example of Algorithm Using Comparison Rule 3

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

Figure 7-6. Example of Compression Algorithm Using Comparison Rule 3

With comparison rule 3, the header is B'110' and the symbol-definition bit string is created by comparing each Type 1 slice-pair with a reference slice-pair composed of the next-to-previous slices, as shown in Figure 7-6. The resultant bit string, including the header, is:

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

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


OUTBOUND 3270DS


FUNCTION


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

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | L | Length of this structure |
| 2 | X'40' | Outbound 3270DS |
| 3 | PID | Partition identifier (X'00' through X'7E') |
| 4 | CMND | Partition command |
| | X'F1' | Write - The associated data is written to the buffers of the specified partition. |
| | X'F5' | Erase/Write - The viewport and buffers of the specified partition are cleared, and the data is written to the buffers as in a normal Write. |
| | X'7E' | Erase/Write Alternate - Same action as Erase/Write. |
| | X'6F' | Erase All Unprotected - All the unprotected fields in the named partition are set to nulls. |
| 5 | WCC | WCC (write control character) |
| 6-n | DATA | Orders and data |

OPERATION

PID specifies which partition the data in this structured field is to be directed to. The specified partition must be defined or a data stream error occurs. (See Figure B-1.)

CMND contains control information for the partition. Note that, when CMND = X'6F', no WCC appears in the structured field. CMND values not shown are reserved and are rejected (sense code X'1003').

WCC contains the write control character as defined for the 3270 data stream in "Write Commands." When CMND + X'F1', bits 0 to 3 of the WCC byte are ignored for displays and bits 0 and 1 are ignored for printers.

See Figure B-1 for other error conditions and the sense codes that are returned.

## QUERY REPLY

FUNCTION

Transmits a response to a Query request.   Query requests are contained
in the Read Partition structured field.

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | L | Length of this structure |
| 2 | X'81' | Query Reply |
| 3 | REPLY CODE | Reply identifying code:<br>    X'81' = Usable Area<br>    X'84' = Partitions<br>    X'85' = Character set<br>    X'86' = Color<br>    X'87' = Highlight<br>    X'88' = Reply modes<br>    X'8A' = Field validation |
| 4-n | PLIST | Parameter list |

OPERATION

In reply to a Query, the device transmits inbound a set of structured
fields that describe the device features.  Only the query replies that
reflect the functions/features supported by the device are transmitted
inbound.

Each structured field has an ID of X'81nn', where nn is the reply code
identifying the feature.

PLIST is a variable-length parameter list.  For the parameter list for
each reply code, see the specific Query Reply in the definitions that
follow.

Note: When this structured field is used in the 3270 data stream, it is
preceded by an AID of X'88'.  If the structured field is one of a set of
Query Reply structured fields, only the first is preceded by an AID of
X'88'

## QUERY REPLY (CHARACTER SETS)

FUNCTION

Transmits information about (1) each character set supported and (2) the ability of the device to support the Load PS structured field and the graphic-escape character.

The display sends this form of Query Reply when responding to a Read Partition structured field indicating Query.

FORMAT

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 0-1 | | L | Length of this structure |
| 2 | | X'81' | Query reply |
| 3 | | X'85' | Character sets |
| 4 | | FLAGS | |
| | 0 | ALT<br>B'0'<br>B'1' | Graphic escape (GE):<br>  GE not supported<br>  GE supported |
| | 1 | RES | Reserved - must be zero |
| | 2 | MULTI<br>B'0'<br>B'1' | Multiple-character set capability:<br>  Load PS not supported<br>  Load PS supported |
| | 3 | EXT<br>B'0'<br>B'1' | Extended form of Load PS:<br>  Load PS extended form not supported<br>  Load PS extended form supported |
| | 4 | RES | Reserved - must be zero |
| | 5-7 | RES | Reserved - must be zero |
| 5 | | RES | Reserved - must be zero |
| 6 | | SDW | Matrix width for Load PS |
| 7 | | SDH | Matrix height for Load PS |
| 8-11 | | FORM | Supported Load PS format types; bit-encoded (if bit i = 1, then type i is supported). |
| 12 | | DL | Length of each descriptor |

The base part is followed by one or more character set descriptors, each defining the characteristics of one character set (ROS or RWS). The length of each descriptor is contained in DL. At least one descriptor must be included, the one where SET X ='00'. The descriptor format is:

| 1 | | | SET | Device-specific character set ID (RWS or ROS number) |
|---|---|---|---|---|
| 2 | | | FLAGS | |
| | | 0 | LOAD<br>B'0'<br>B'1' | Loadable:<br>    Nonloadable character set<br>    Loadable character set |
| | | 1 | TRIPLE<br>B'0'<br>B'1' | Triple plane:<br>    Single-plane character set<br>    Triple-plane character set |
| | | 2 | RES | Reserved - must be zero |
| | | 3 | CB<br>B'0'<br>B'1' | Compare bit:<br>    LCID compare<br>    No LCID compare |
| | | 4-7 | RES | Reserved |
| 3 | | | LCID | Local character set ID |

OPERATION

The parameters describe the device's ability to support character sets, as follows:

• For a loadable character set, SDW and SDH define the matrix size to be used by the Load PS structured field.

• FORM defines the Load PS format types supported by the device. This field is a 32-bit field. Each bit corresponds to a format type that can be specified in Load PS (see TYPE field of Load PS). Thus if bit i = 1 in FORM, the device supports format type i in Load PS.

Currently, Load PS format types 1, 2 , 5, and 6 are defined. The remaining values in the first byte of the field are reserved, as are bytes 9 through 11.

Thus bits 0, 3, 4, and 7 through 31 are reserved; the remaining bits may be set, depending on the formats supported.

• DL defines the length of each descriptor (in bytes).

The parameters in each descriptor define the characteristics of a character set, as follows:

• SET defines the device specific ID. For a loadable character set, this is the value specified in the RWS parameter of the Load PS. SET also defines the key that can be used by the operator to select this character set.

• LOAD indicates that this character set is loadable.

• TRIPLE indicates that this character set has one or three planes.

• CB indicates whether this character set can be compared for copy (see also the CB field and the discussions in Load PS).

• LCID identifies the local character set ID currently connected to this character set. The ID may not identify the contents uniquely. A value of X'FF' indicates that this RWS is free (not assigned). Load PS allows an LCID to be connected to a loadable character set. The data stream value used in SA, SFE, or MF orders to identify LCID is then characters from this set. Nonloadable character sets have a default LCID assigned to them from X'F0' through X'FE'.


QUERY REPLY (COLOR)


FUNCTION


Transmits information about the color features of the device. The structured field is sent if the device is capable of interpreting 3270 field attribute as color attributes or accepts some subset of the color attribute values.

The device sends this form of Query Reply when responding to a Read Partition structured field indicating Query.

FORMAT

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 0-1 | | L | Length of this structure |
| 2 | | X'81' | Query reply |
| 3 | | X'86' | Color |
| 4 | | FLAGS | |
| | 0 | RES | Reserved |
| | 1 | B'0'<br>B'1' | Printer Ribbon<br>  Printer only - Black<br>  ribbon is not loaded<br>  Printer only - Black<br>  ribbon is loaded |
| | 2-7 | RES | Reserved - must be zero |
| 5 | | N | Length of color attribute list<br>(N = number of CAV/COLOR pairs) |
| n | | CAV(n) | Color attribute value accepted by<br>the device |
| n+1 | | CID(n)<br><br>X'00'<br>X'F1'<br>X'F2'<br>X'F3'<br>X'F4'<br>X'F5'<br>X'F6'<br>X'F7' | Color identifier of the color<br>displayed/printed for CAV(n).<br>Color identifiers are:<br>  Default color<br>  Blue<br>  Red<br>  Pink<br>  Green<br>  Turquoise<br>  Yellow<br>  Neutral |

OPERATION


The parameters CAV(n) are all those color attribute values that are
accepted by the device without causing a negative response.  Bytes n and
n+1 are repeated for each of the data stream values accepted by the
device.

The parameters CID(n) identify the colors that are displayed or printed
by the device for each of the accepted color attribute values.  Note

that the device must either display the color whose color identifier is the same as the color attribute value or display the device default color.

The color associated with the CAV(n) value of X'F7' defines the default color that is displayed/printed when a single plane character set is referenced; the associated CID(n) value may be any of the values in CAV(n), including X'00'. The value X'F7' for CID(n) indicates the default color is neutral.

The CAV(n) value of X'00' may have an associated CID(n) value of any of the defined values except X'00'.

All devices that send Query Reply (Color) are required to have the values CAV1 = X'00', CID1 = value associated with the the device default color, as the first entry in the CAV/CID pairs list.

Query Reply (Color) Example:

| CID(n) Color Identifier | | | |
|---|---|---|---|
| CAV(n) Attribute Value | Color Display | Mono Display | Color Printer |
| X'00' | X'F4' | X'F4' | X'F7' |
| X'F1' | X'F1' | X'00' | X'F1' |
| X'F2' | X'F2' | X'00' | X'F2' |
| X'F3' | X'F3' | X'00' | X'00' |
| X'F4' | X'F4' | X'00' | X'F4' |
| X'F5' | X'F5' | X'00' | X'00' |
| X'F6' | X'F6' | X'00' | X'00' |
| X'F7' | X'F7' | X'00' | X'00' |

## QUERY REPLY (FIELD VALIDATION)

### FUNCTION

Transmits the field validation capabilities of the device.

The device sends this form of Query Reply when responding to a Read Partition structured field indicating Query.

### FORMAT

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 0-1 | | X'0005' | Length of this structure |
| 2 | | X'81' | Query reply |
| 3 | | X'8A' | Field Validation |
| 4 | | | Types supported |
| | 0-4 | | Reserved |
| | 5 | B'0'<br>B'1' | Mandatory fill not supported<br>Mandatory fill supported |
| | 6 | B'0'<br>B'1' | Mandatory field not supported<br>Mandatory field supported |
| | 7 | B'0'<br>B'1' | Trigger field not supported<br>Trigger field supported |

### OPERATION

The structured field specifies that the device supports field validation and indicates the types of validation the device supports.

QUERY REPLY (HIGHLIGHT)

FUNCTION

Transmits the types of highlighting supported by the device.

The device sends this form of Query Reply when responding to a Read
Partition structured field indicating Query.

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | L | Length of this structure |
| 2 | X'81' | Query reply |
| 3 | X'87' | Highlight |
| 4 | N | Number of attribute-value/action pairs |
| n | Vi | Data stream attribute value accepted |
| n+1 | Ai | Data stream action |

OPERATION

If a device accepts the highlight attribute type, then it must accept
attribute value X'00' (default specification). It may optionally accept
other attribute values. All accepted values are listed in Query Reply
(Highlight). For each accepted value, Query Reply (Highlight) lists the
device action in replies of 1-byte action codes. The action codes (Ai)
are:

• X'F0' Field attribute

• X'F1' Blink

• X'F2' Reverse video

• X'F4' Underscore

In addition to the above, the action code may specify X'00'. The code
X'00' indicates that the device action for the corresponding attribute
value is the same as the action for the attribute value X'00', that is,
the default action of the device.

An example of a Query Reply (Highlight) for a display is:

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | X'000C' | Length |
| 2 | X'81' | Query reply |
| 3 | X'87' | Highlight |
| 4 | X'04' | Number of pairs |
| 5 | X'00' | Attribute value (default) |
| 6 | X'F0' | Normal action |
| 7 | X'F1' | Attribute value |
| 8 | X'F1' | Action - blink |
| 9 | X'F2' | Attribute value |
| 10 | X'F2' | Action - reverse video |
| 11 | X'F4' | Attribute value |
| 12 | X'F4' | Action - underscore |

## QUERY REPLY (PARTITIONS)

### FUNCTION

Transmits the maximum number of partitions that can be supported on this device and the device's ability to support partition-related functions.

The device sends this form of Query Reply when responding to a Read Partition structured field indicating Query.

### FORMAT

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 0-1 | | L | Length of this structure |
| 2 | | X'81' | Query reply |
| 3 | | X'84' | Partitions |
| 4 | | N | Maximum number of partitions supported |
| 5-6 | | M | Maximum interactive partition size |
| 7 | | FLAGS | |
| | 0 | B'0'<br>B'1' | Vertical scrolling not supported<br>Vertical scrolling supported |
| | 1-7 | RES | Reserved - must be zero |

### OPERATION

N specifies the maximum number of partitions supported. Partitions are named 0 through N-1. N = 0 is never returned. Support of partitions implies that they can be created explicitly.

M defines the number of bytes available for storing character data for the maximum interactive partition size. The maximum size is achievable only when a single partition exists, and, in this case, the size is the product of the width and depth of the one partition.

When multiple partitions exist, the sum of the partition sizes must be less than or equal to M.

QUERY REPLY (REPLY MODES)

FUNCTION

Transmits the reply modes supported by the device.

The device sends this form of Query Reply when responding to a Read
Partition structured field indicating Query.

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | L | Length of this structure |
| 2 | X'81' | Query reply |
| 3 | X'88' | Reply modes |
| 4-(n-1) | MODES<br>X'00'<br>X'01'<br>X'02' | Modes supported:<br>Field mode<br>Extended Field mode<br>Character mode |
| other | RES | Reserved - must be zero |

OPERATION

For a description of reply modes, see "Set Reply Mode."

QUERY REPLY (USABLE AREA)

FUNCTION

Transmits the size and characteristics of the screen.

The display sends this form of Query Reply when responding to a Read
Partition structured field indicating Query.

The screen comprises a specific number of cells.  The number of these
cells and their size are specified in this query reply.

A character may be placed in each cell.  This character may be taken
from any of the character sets defined in Query Reply (Character Sets).

FORMAT

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 0-1 | | L | Length of this structure |
| 2 | | X'81' | Query reply |
| 3 | | X'81' | Usable area |
| 4 | | FLAGS | |
| | 0-1 | RES | Reserved. |
| | 2 | 3270<br>B'0'<br>B'1' | Outbound 3270DS structured field:<br>  Outbound 3270DS supported<br>  Outbound 3270DS not supported |
| | 3 | HC<br>B'0'<br>B'1' | Hard copy characteristics:<br>  Not a hard-copy device<br>  A hard-copy device |
| | 4-7 | ADDR<br><br>X'0'<br>X'1'<br>X'2'<br>X'3'<br>X'F' | Allowable addressing modes for alphanumeric partitions:<br>  12-bit addressing only<br>  12/14-bit addressing allowed<br>  16-bit addressing allowed<br>  12/14- or 16-bit addressing allowed<br>  Unmapped - no explicit addressing |
| 5 | | FLAGS | |
| | 0 | RES | Reserved - must be zero |
| | 1 | CHAR<br>B'0'<br>B'1' | Characters<br>  Matrix characters<br>  Nonmatrix characters<br>  (for example, a belt printer) |
| | 2-7 | RES | Reserved - must be zero |
| 6-7 | | W | Width of usable area in cells |
| 8-9 | | H | Height of usable area in cells |
| 10 | | UNITS<br>X'00' | Unit of measure:<br>  Inches |

| Byte | Bit | Content | Meaning |
|------|-----|---------|---------|
| 11-14 | | Xr | Distance between points in x-direction as a fraction, measured in units:<br>   2-byte numerator<br>   2-byte denominator |
| 15-18 | | Yr | Distance between points in y-direction as a fraction, measured in units:<br>   2-byte numerator<br>   2-byte denominator |
| 19 | | AW | No of x-units in default cell |
| 20 | | AH | No of y-units in default cell |
| 21-22 | | BUFFSZ | Character buffer size (in bytes) |

OPERATION

3270 indicates that the Outbound 3270DS structured field is supported.

HC indicates that this is a hard-copy device, that is, a printer.

CHAR indicates matrix or nonmatrix characters. If CHAR = B'1' (nonmatrix characters), bytes 10-20 are not applicable and must be set to zeros.

W and H define the size of the usable area in <u>cells</u> of size AW x AH (see also AW, AH). Thus, the total size of the usable area in points is W x AW by H x AH. The primary unit of measure is always character cells.

For a printer, W and H are the maximum print position (MPP) and the maximum print line (MPL) supported by the hardware. These values correspond to the MPP and MPL values in the SNA Character String (SCS). On some devices, MPP and MPL may be set by the operator. The values returned in this query reply refer to the hardware's capability, not to the current setting.

UNITS, Xr, and Yr define the spacing granularity. For example, if the device has 72.5 points per inch horizontally and 69 points per inch vertically, the values would be:

```
    UNITS - X'00'
    Xr    - X'00020091' (that is, 2/145 decimal)
    Yr    - X'00010045' (that is, 1/69 decimal)
```

In combination with Load PS, the application program can use these parameters to present data where precise dimensions are critical. Alternatively, if the relative dimension (x/y aspect ratio) is critical, the ratio Xr/Yr allows this relationship to be controlled.

AW and AH define the default cell size for the device.

BUFFSZ defines the amount of character buffer available. It applies only to a device that does not support partitions. If the device supports partitions, Query Reply (Partitions) defines the buffer size, and the BUFFSZ parameter will be zero or, optionally, omitted.

BUFFSZ is required for printer operation, where the buffer is loaded prior to initiating the print.


READ PARTITION


FUNCTION


Reads data from a specific partition or queries the device.


FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | L | Length of this structure |
| 2 | X'01' | Read partition |
| 3 | PID | Partition identifier:<br>X'00' - X'7E' (read operations), or<br>X'FF' (query operations). |
| 4 | TYPE<br>X'02'<br>X'6E'<br>X'F2'<br>X'F6' | The type of operation to be performed<br>Query<br>Read Modified All (RMA)<br>Read Buffer (RB)<br>Read Modified (RM) |

OPERATION

If any of the following conditions exist, Read Partition is rejected (see Figure B-1):

* The device is in RETRY state (sense code X'0871').

* The terminal is not in RCV (receive) or CONT (contention) state.

* Read Partition is not the last structured field in the RU chain.

* The RU chain does not specify CD.

* The operation type is Query and the PID is not X'FF'.

Otherwise, the following steps are performed:

1. The enter-inhibit condition is raised.

2. If the PID specified in Read Partition is not X'FF', then INPID is set to the specified PID.

3. INOP is set to the operation specified in Read Partition.

4. For a read-modified, read-modified-all, or read-buffer operation, the AID is set to X'61'.

5. The data is transmitted inbound.

6. The display is placed in RCV state.

7. The device is placed in retry-read state.

8. For read operations with type code X'F6', X'6E', or X'F2', the data is transmitted inbound in the format specified by the operation and the inbound reply mode with an AID of X'61'.

9. For a query operation (type code X'02'), a set of query replies is transmitted inbound. These replies describe the features on that device.

The inbound data stream in reply to Read Partition (Query) is:

| X'88' | Query Reply | Query Reply |
| --- | --- | --- |

where the reply is sent as a sequence of Query Reply structured fields.

## RESET PARTITION

FUNCTION

Resets definable partition characteristics.

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1  | X'0004' | Length of this structure |
| 2    | X'00'   | Reset partition |
| 3    | PID     | Partition identifier (X'00' - X'7E') |

OPERATION

The partition characteristics for the specified partition (PID) are
reset to their default values:

Partition Characteristic                Default

Inbound Reply mode                      Field

## SET REPLY MODE

FUNCTION

Defines the format of any subsequent inbound data streams.

FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | L | Length of this structure |
| 2 | X'09' | Set Reply mode |
| 3 | PID | Partition identifier (X'00' through X'7F') |
| 4 | MODE<br>X'00'<br>X'01'<br>X'02'<br>other | <br>Field mode<br>Extended Field mode<br>Character mode<br>Reserved |
| 5-n | ATTR LIST<br>(Present only if mode is X'02') | The type codes for the attribute types that:<br>• Will be transmitted using SA orders if mode X'02' is specified.<br><br>• The operator may select from the keyboard. (The operator may select which attribute value is to be associated with the keyed data.) |

OPERATION

Subsequent inbound data streams from the named partition sent inbound in response to a read command are in the format defined by the Set Reply Mode structured field. This reply mode remains in effect until set by another Set Reply Mode, or until reset by a reset action.

In Field mode or Extended Field mode, attribute selection by the operator is not allowed. Any operator selection allowed by a previous Set Reply Mode structured field is reset to the default value.

In Character mode, the listed attribute types may have values selected by the operator. Any operator selection current when Set Reply Mode is interpreted is not reset.

## SET WINDOW ORIGIN

### FUNCTION

Changes the position of the window within the presentation space of the specified partition.

### FORMAT

| Byte | Content | Meaning |
|------|---------|---------|
| 0-1 | X'0008' | Length of this structure |
| 2 | X'0B' | Set window origin |
| 3 | PID | Partition identifier (X'00' through X'7E') |
| 4-5 | RW | Row position of the window origin |

### OPERATION

1. If the named partition does not exist, an error situation arises. (See Figure B-1.)

2. The window of the specified partition is repositioned to the new origin. If the following condition occurs, the window is not moved and a sense code is returned.

    $$RW + HW > H$$

    where:  HW = Height of window (viewport)
            H  = Height of presentation space

3. The current cursor position for the named partition is not changed unless the new window origin results in the current cursor position's being outside the window. If outside the window, the current cursor postion is moved by the minimum number of row and columns needed to place it in the nearest peripheral row or column of the window.

# CHAPTER 8. BINARY SYNCHRONOUS COMMUNICATIONS (BSC) ENVIRONMENT

The 3270 data stream operates the same in a binary synchronous communication (BSC) environment as it does in the SNA environment, except for the differences noted in this chapter.

## TRANSPARENT MODE

The SNA environment provides inherent transparency on the data line. This mode of operation permits greater versatility in the range of coded data that can be transmitted. This is because all data, including the normally restricted data link line control characters, is treated only as bit patterns when transmitted in transparent mode-operation. This is required when transmitting binary counts and addresses that may appear in the data stream for the functions of extended highlighting, color, and programmed symbols.

BSC data links may be either nontransparent or transparent, however, when the SF, SFE, and MF orders and the WSF command are supported; the BSC link must be in the transparent mode. The format of the data stream for BSC is:

- For nontransparent mode
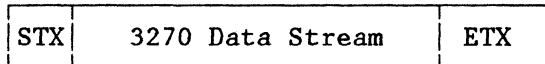
| STX | 3270 Data Stream | ETX |
|-----|------------------|-----|

- For transparent mode
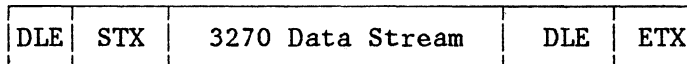
| DLE | STX | 3270 Data Stream | DLE | ETX |
|-----|-----|------------------|-----|-----|

## WRITE COMMANDS

The Erase/Write and Erase/Write Alternate commands operate the same in a BSC environment as in the SNA enviornment. The Write command operates the same in a BSC environment as in the SNA environment except that the starting buffer location depends upon the following considerations:

- The starting location may be specified by a Set Buffer Address order that follows the WCC.

- The starting location will be the buffer address containing the cursor if the Write command is not chained from a Copy or Erase All Unprotected.

- The starting location will be the current buffer address if the Write command is chained from a Read or another Write command.

## READ COMMANDS

The read commands operate the same in a BSC environment as in the SNA environment except for the differences that follow.

## READ BUFFER COMMAND

Execution of the Read Buffer command causes all the data in the addressed device's buffer, from the buffer location at which reading starts through the last buffer location, to be transferred to the application program's storage. The transfer of data begins:

1. From buffer address 0 if the Read Buffer command is unchained.

2. From the current buffer address if the Read Buffer command is chained from either a Write, Erase/Write, Erase/Write Alternate, Read Modified, or another Read Buffer command. Regardless of where the transfer of data begins, data transfer from the buffer will terminate when the last character location in the buffer has been transferred, or when the last character of a text block has been transferred.

## READ MODIFIED COMMAND

Read Modified initiates one of three operations as determined by operator actions at the display: (1) read modified, (2) short read, or (3) test request read.

Read Modified functions the same as in an SNA environment except that the buffer location at which the search begins for field attributes that define modified fields is a function of command chaining. This location is:

1. Buffer address 0 if the Read Modified command is unchained or is chained from a Copy.

2.  The current buffer address if the Read Modified command is chained
    from a Write, Erase/Write, Read Modified, Read Modified All, or a
    Read Buffer command.

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

The transfer of read data is terminated as follows:

1.  If the last modified field is wrapped from the last buffer location
    (for example, 479 or 1919) to the first location , the operation is
    terminated after all the data in the field is transferred (nulls
    are suppressed).  The buffer address at the end of the operation is
    the address of the next field attribute byte in the buffer.  For
    example, if a modified field extends from address 1900 (the field
    attribute) to address 79 (wrapped field), the data from address
    1901 through 79 is transferred (nulls suppressed); in this case,
    the read operation is terminated with the buffer address set to 80
    (the field attribute byte of the next field).

2.  If the buffer does not contain a wrapped modified field, the
    modified data stream is terminated when the last modified field is
    transferred.  At the end of the operation, the buffer address is
    set to 0.

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

    If the buffer is unformatted, the read data stream consists of the
    3-byte read heading followed by all the alphanumeric data in the
    buffer (nulls suppressed), even when part or all of the data has
    not been modified.  Since an unformatted buffer contains no field
    attribute bytes, no SBA codes with associated addresses or address
    characters are included in the data stream, and the modification of
    data cannot be determined.  Data transfer starts at address 0,
    regardless of command chaining, and continues to the end of the
    buffer.  At the end of the operation, the buffer address is set to
    0.  This read operation can also be terminated by the channel byte
    count's reaching 0 before all data is read.  In this case, the
    buffer address after termination is undefined.


TEST REQUEST READ


The Read Modified command causes a test-request-read operation if the
TEST REQ or SYS REQ key has been pressed at the selected device.  The
Test request Read data stream sent inbound to the application program is
as follows:

```
 _____
|                |
|   SOH          |
|----------------|
|   %            |
|----------------|
|   /            |
|----------------|
|   STX          |
|----------------|
|   Input Data   |
|_____|
```

The Test Request Read heading is generated by the display.  The
remainder of the data stream is the same as described previously for
read-modified operations, excluding the 3-byte read heading (AID and
cursor address).  If the buffer is unformatted, all the alphanumeric
data in the buffer is included in the data stream (nulls suppressed),
starting at address 0.  If the buffer is formatted, each field-attribute
byte is examined for a set MDT bit.  Each time a set MDT bit is found,
the alphanumeric data in the field associated with that bit is sent to
main storage (nulls suppressed).  If no MDT bits are set, the read data
stream consists of the Test Request Read heading only.  The buffer
location at which the search for the MDT bits begins and the transfer of
data ends is the same as described for the read-modified operations.

Note: Usage of the test-request-read function is determined by the
application program.  Normally, the operator would  (1) clear the
display, (2) enter test-request data in a predefined format, and then
(3) press the TEST REQ or SYS REQ key.


INBOUND TRANSMISSIONS


Inbound transmissions result from an operator enter action, an
application-program-initiated (unsolicited) read request, or a host
retry of an inbound transmission.

An operator enter action is one that causes an attention identifier to
be transmitted inbound.  The application program responds with a read
request.  The application program must acknowledge the inbound
transmission before a new inbound operation can be performed.

An application-program-initiated read operation is an inbound
transmission not caused by an operator enter action.  No application
program acknowledgment is required before a new inbound transmission can
occur.

Host retry is a retransmission of the last unacknowleged inbound
transmission from the device.  The application program must acknowledge
receipt of an inbound transmission before a new inbound transmission can
take place.  A host retry transmission does not cause read-state
transitions and is not considered a new inbound transmission requiring

8-4

acknowledgment by the application program. Host retry occurs until an application-program acknowledgment takes place.

The type of inbound transmission is either a Query Reply structured field (the response to the Read Partition Query structured field) or data from the device buffer (for example, modified fields of the display image). An inbound operation device characteristic (INOP), set by the controller, defines the type.


INBOUND OPERATION (INOP)


The INOP determines the operation to be performed when data is transmitted inbound on a retry transmission.

INOP is set by any of the following:

•   An operator enter action sets INOP to Read Modified.

•   Reception of a Read Partition Query structured field sets INOP to Query.

•   Application-program acknowledgment of an inbound transmission sets INOP to Read Modified.


READ STATES


While powered on, a device is in one of seven states with respect to read operations. The three primary states are:

•   Normal read

•   Data pending read

•   Read retry

The data-pending and read-retry states have three substates:

•   Enter

•   Read

•   Stacked enter

The events that cause transitions between states are shown in Figure 8-1.

## NORMAL READ STATE

A device is in normal-read state when powered on, or prior to initiation
of a new read operation, or after use of the RESET key in certain
instances. (See "Read-State Transitions.")

When in normal-read state an operator enter action or the reception of
a Read Partition Query structured field causes the device to prepare to
generate the inbound data stream and to go into a data-pending state.

An application-program-initiated read operation causes the data to be
transmitted with no state transitions occurring. The device remains in
normal-read state.

## DATA PENDING STATES

There are three data pending states

* Data pending enter - the device state after an operator enter
  action.

* Data pending read - the device state after reception of a Read
  Partition Query structured field.

* Data pending stacked enter - the device state after a Read Partition
  Query structured field was received while the device was in
  data-pending-enter read state or retry-enter state (the enter is
  stacked).

A poll received while the device is in a data-pending state causes the
data to be transmitted and the device to be placed in the corresponding
retry state.

## READ-STATE TRANSITIONS

The read-state transitions for BSC are summarized in Figure 8-1.

| Read States / Events | Normal | Data Pending | | | Retry | | |
|---|---|---|---|---|---|---|---|
| | | Enter | Read | Stacked Enter | Enter | Read | Stacked Enter |
| Enter Action | 2 | R | R | R | R | R | R |
| Read Command | 1 | 1 | 1 | 1 | G | G | G |
| Read Partition Query | 3 | 4 | – | – | 4 | 3 | 4 |
| Poll | – | 5 | 6 | 7 | – | – | – |
| Host Acknowledge | – | 1 | 1 | 2 | 1 | 1 | 2 |
| RESET key | – | 1 | – | – | 1 | – | – |

Key:

R – Reject, no state transition
G – Generic retry, no state change
– – No action or state change
1 – Normal-read state
2 – Data-pending-enter state
3 – Data-pending-read state
4 – Data-pending-stacked-enter state
5 – Retry-enter state
6 – Retry-read State
7 – Retry-stacked-enter state

Figure 8-1. Read-State Transitions for BSC

RETRY STATES

There are three retry states:

- Retry enter - the device state after enter data was transmitted to the application program.

- Retry read - the device state after Query Reply data was transmitted to the application program.

- Retry stacked enter - the device state after enter data was stacked and the Query Reply data transmitted to the application program.

While in retry state, the last inbound transmission can be retried by means of a Read Modified command.

A host acknowledgment causes the device to revert from a retry state to the normal-read state, or, in the case of retry stacked enter, to the data-pending-stacked-enter read state.


INDICATORS


With reference to Figure 8-1 the indicators displayed in the operator information area of a display are as follows:

| State | Indicator |
|---|---|
| Normal read | a. No indicator or |
|  | b. System Lock |
| Data pending enter | TWAIT |
| Data pending read | TWAIT |
| Data pending stacked enter | TWAIT |
| Retry enter | System Lock |
| Retry read | TWAIT |
| Retry stacked enter | TWAIT |


HOST ACKNOWLEDGMENTS


After inbound transmissions resulting form operator enter actions, or after transmissions of the reply to a Read Partion Query structured field, the transmission must be acknowledged before a new inbound operation can be performed.

For inbound transmissions generated by operator enter actions, the following are host acknowledgments:

* An outbound transmission containing a Write, Erase/Write, or Erase/Write Alternate command followed by a WCC with keyboard restore bit set to 1, or an Erase All Unprotected command.

* A copy command.

For inbound Query Reply transmissions the acknowledgment is:

* Any valid outbound data stream transmission other than a read command. For purposes of Query Reply acknowledgment, write commands without a write control character are considered an acknowledgment. The reception of a Write Structured Field command is also an acknowledgment.

Host acknowledgment resets INOP to Read Modified.

## PROCESSING OF READ COMMANDS

In a BSC environment, a read command (Read Buffer, Read Modified, or Read Modified All) as the 1st byte of the data stream is processed follows:

1. If the device is in normal-read state, then the command performs a read and the display data is transmitted inbound as defined by;

   a. The Read Modified, Read Modified All, or Read Buffer command.

   b. The AID (Read Modified command only)

   c. The reply mode

   For items a, b, and c above the device remains in normal-read state.

2. If the device is in a data-pending state, then the command performs a read and data is transmitted as defined by;

   a. The Read Modified, Read Modified All, or Read Buffer command.

   b. The AID (Read Modified command only)

   c. The inbound reply mode

   The device is placed in normal-read state.

3. If the device is in a retry state, then the command performs a retry as follows:

   a. If the command is Read Modified, and INOP specifies Query, then the appropriate query replies are transmitted inbound.

   b. If the command is Read Modified, and INOP specifies Read Modified, then data is transmitted as defined by;

      (1) The Read Modified command
      (2) The AID
      (3) The inbound reply mode

   c. If the command is Read Buffer, then data is transmitted inbound as defined by;

      (1) The Read Buffer command
      (2) The inbound reply

   For items a, b, and c the device remains in the original retry state.

## PROCESSING OF READ PARTITION QUERY STRUCTURED FIELDS

Read Partition Query and the Query Reply are processed as follows:

1.  If the device is in normal-read state, then

    a.  The TWAIT indicator is displayed.

    b.  INOP is set to Query.

    c.  The device prepares to generate the required inbound data stream.

    d.  The device is placed in data-pending-read state.

    e.  A later poll causes the data to be transmitted and the device to be placed in retry-read state.

2.  If the device is in data-pending-enter or retry-enter state, then

    a.  The outstanding data is stacked.

    b.  The TWAIT condition remains in effect.

    c.  INOP is set to Query.

    d.  The device prepares to generate the required inbound data stream.

    e.  The device is placed in data-pending-stacked-enter state.

    f.  A later poll causes the Query Reply data to be transmitted inbound and the device to be placed in retry-stacked-enter state.

## BSC COPY COMMAND

BSC uses the Copy command to accomplish a local-copy function.  Refer to Chapter 6 for a detailed description of the operation of the Copy command.

CHAPTER 9. NON-SNA ENVIRONMENT (LOCALLY ATTACHED DEVICES--3272 VERSION)


This chapter explains those data stream operations that differ from the
SNA version, for locally attached devices in a non-SNA environment.
Most of the explanations cover only the differences between SNA and
non-SNA for locally attached devices.


COMMANDS


Except for the Read Modified All command, the commands valid for an SNA
environment are also valid for the non-SNA locally attached environment.

The additional commands of Select, No Operation (No-op), and Sense are
used in the non-SNA locally attached environment to improve device
utilization, to retrieve pending status, and to obtain unit check
definition, respectively.  These commands are not 3270 data stream
commands and are not discussed in this manual.  See the applicable
product manuals for an explanation of these commands.


WRITE COMMANDS


The Erase/Write and Erase/Write Alternate commands operate the same as
in an SNA environment.  The Write command operates the same as in the
SNA environment except that the starting buffer location depends upon
the following considerations:

•   The starting location may be specified by a Set Buffer Address order
    that follows the WCC.

•   The starting location will be the buffer address containing the
    cursor if the Write command is not chained or if it is chained from
    a Select, Erase All Unprotected, No Operation, or Sense command.

•   The starting location will be the current buffer address if the
    Write command is chained from a Read or another Write command.


READ COMMANDS


The read commands for devices in this non-SNA environment operate the
same as for an SNA environment except for the differences noted below.

Read Buffer Command


Execution of the Read Buffer command causes all data in the addressed
device buffer location to be transferred to main storage  The transfer
of data begins:

* From buffer address 0 if the Read Buffer command is unchained.

* From the current buffer address if the Read Buffer command is
  chained from either a Write, Erase/Write, Erase/Write Alternate,
  Read Modified, or another Read Buffer command.  Regardless of where
  the transfer of data begins, data transfer from the buffer will
  terminate when the last character location in the buffer has been
  transferred or before the last character location has been
  transferred when the channel byte count reaches 0 (in this case, the
  buffer address after termination is undefined).


Read Modified Command


Read Modified initiates one of three operations as determined by
operator actions at the display: (1) Read Modified,  (2) Short Read  or
(3) Test Request Read.

READ-MODIFIED OPERATION: The Read Modified command functions the same as
in an SNA environment except that the buffer location at which the
search begins for the field attribute bytes that define modified fields
is a function of command chaining.  This location is:

* Buffer address 0 if the Read Modified command is unchained or is
  chained from a Select, Sense, or No Operation command.

* The current buffer address if the Read Modified command is chained
  from a Write, Erase/Write, Read Modified, or Read Buffer command.

The search for modified fields ends when the channel byte count reaches
0.  The transfer of data is terminated as follows:

  1.  If the last modified field is wrapped from the last buffer
      location, the operation is terminated after all the data in the
      field has been transferred (nulls are suppressed).  The buffer
      address at the end of the transfer is the address of the next field
      attribute byte in the buffer.  For example, if a modified field
      extends from address 1900 (the field attribute byte) to address 79
      (wrapped field), the data from addresses 1901 through 79 is
      transferred (nulls suppressed).  In this case, the read operation
      is terminated with the buffer address set to 80 (the field
      attribute byte of the next field).

2.  If the buffer does not contain a wrapped modified field and if the channel byte count has not reached 0, the modified data stream is terminated when the last modified field is transferred.  At the end of the operation, the buffer address is set to 0.

3.  If the channel byte count reaches 0 before all modified data is transferred, read operations are terminated and the remaining modified data is not transferred.  The buffer address after termination is undefined.

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

If the buffer is unformatted, the read data stream consists of the 3-byte read heading followed by all alphanumeric data in the buffer (nulls suppressed), even when part or all of the data has not been modified.  Since an unformatted buffer contains no field attributes, no SBA codes with associated addresses or address characters are included in the data stream, and the modification of the data cannot be determined.  Data transfer starts at address 0, regardless of command chaining, and continues to the end of the buffer.  At the end of the operation, the buffer address is set to 0.  This read operation can also be terminated by the channel byte count's reaching 0 before all data is read.  In this case, the buffer address after termination is undefined.

TEST REQUEST READ OPERATION: The Read Modified command causes a Test Request Read operation if the TEST REQ or SYS REQ key has been pressed at the selected device.  The Test Request Read data stream sent inbound to the application program is the same as for the BSC environment.


INBOUND TRANSMISSIONS


Inbound transmissions result from an operator enter action, a host initiated (unsolicited) read request, or a host retry of an inbound transmission.

An operator enter action is one that causes the attention identifier to be transmitted inbound.  The application program responds with a read request.  The application program must acknowledge the inbound transmission before a new inbound operation can be performed.  (See "Host Acknowledgments.")

A host-initiated read operation is an inbound transmission not caused by an operator enter action.  No host acknowledgment is required before a new inbound transmission can occur.

Host retry is a retransmission of the last unacknowledged inbound transmission from the device.  The application program must acknowledge reception of an inbound transmission before a new inbound transmission can take place.  A host retry transmission does not cause read-state transitions and is not considered a new inbound transmission requiring

host acknowledgment. Host retry occurs until a host acknowledgment takes place.

The type of inbound transmission is either a Query Reply structured field (the reply to the Read Partition Query structured field) or data from the device buffer (for example, modified fields of the display image). An inbound operation (INOP), set by the controller, defines the type.


## INBOUND OPERATION (INOP)

The INOP determines the operation to be performed when data is transmitted inbound, or when the device is in a data-pending state.

INOP is set by any of the following:

* An operator enter action sets INOP to Read Modified.

* Reception of a Read Partition Query structured field sets INOP to query.

* Host acknowledgment of an inbound transmission sets INOP to Read Modified.


## READ STATES

While powered on, a device is in one of seven states with respect to read operations. The three primary states are:

* Normal read

* Data pending read

* Retry read

The data-pending read and retry read have three substates: enter, read, and stacked enter.

The events that cause transitions between the states are shown in Figure 9-1.


## NORMAL READ STATE

A device is in normal-read state when powered on, or prior to initiation of a read operation, or after use of the RESET key in certain instances. (See Figure 9-1.)

When in normal-read state, an operator enter action or the reception of a Read Partition Query structured field causes the device to prepare to generate the inbound data stream, and then to go into a data-pending state.

An application-program-initiated read operation causes the data to be transmitted with no state transitions occurring.  The device remains in normal-read state.

DATA PENDING STATES

There are three forms of data-pending states:

*   Data pending enter:  when enter data has been transmitted to the application program.

*   Data pending read:  the device state after reception of a read-partition-query structured field.

*   Data pending stacked enter:  when a read-partition structured field has been received while the device is in data-pending-enter state or retry-enter state (the enter data is stacked).

A read command received while the device is in data-pending state causes the data to be transmitted inbound and the device to be placed in the corresponding retry state.

An operator enter action will not be processed, it will be rejected.  A host-initiated read partition will be rejected.

RETRY STATE

There are three forms of retry state.

*   Retry enter, when enter data has been transmitted to the host.

*   Retry read, when the read data has been transmitted to the host.

*   Retry stacked enter, when enter data has been stacked and the query-reply data has been transmitted to the host.

While in retry state, the last inbound transmission can be retried by means of a Read Modified command.

A host acknowledgment causes the device to revert from a retry state to the normal-read state or, in the case of retry stacked enter, to the data-pending-stacked-enter-read state.

The read-state transitions for non-SNA locally attached devices are summarized in Figure 9-1.

| Event | Read States / Normal | Data Pending Enter | Read | Stacked Enter | Retry Enter | Read | Stacked Enter |
|---|---|---|---|---|---|---|---|
| Enter Action | 2 | R | R | R | R | R | R |
| Read Command | 1 | 5 | 6 | 7 | G | G | G |
| Read Partition Query | 3 | 4 | – | – | 4 | 3 | 4 |
| Host Acknowledge | – | 1 | 1 | 2 | 1 | 1 | 2 |
| RESET key | – | 1 | – | – | 1 | – | – |

Key:

R — Reject, no state transition
G — Generic Retry, no state transition
– — No action or state change
1 — Normal Read state
2 — Data Pending Enter state
3 — Data Pending Read state
4 — Data Pending Stacked Enter state
5 — Retry Enter state
6 — Retry Read State
7 — Retry Stacked Enter state

Figure 9-1. Read-State Transitions for Non-SNA Locally Attached Devices--3272 Version

INDICATORS

With reference to Figure 9-1, the indicators displayed in the operator information area of a display are as follows:

| State | Indicators |
|---|---|
| Normal read state | a. No indicator or b. System Lock |
| Data pending enter | System Lock |
| Data pending read | TWAIT |
| Data pending stacked enter | TWAIT |
| Retry enter | TWAIT |
| Retry read | TWAIT |
| Retry stacked enter | TWAIT |

HOST ACKNOWLEDGMENTS


After inbound transmissions resulting from operator enter actions, or
after transmissions of the reply to a Read Partion Query structured
field, the transmission must be acknowledged before a new inbound
operation can be performed.

For inbound transmissions generated by operator enter actions, the host
acknowledgment is:

•   An outbound transmission containing a Write, Erase/Write, or
    Erase/Write Alternate command followed by a WCC with the keyboard
    restore bit set to 1, or an Erase All Unprotected command.

For inbound Query Reply transmissions, the acknowledgment is:

•   Any valid outbound data stream transmission other than a read
    command.  For purposes of Query Reply acknowledgment, write commands
    without a write control character are considered an acknowledgment.
    The reception of a Write Structured Field command is also an
    acknowledgment.

Host acknowledgment resets INOP to Read Modified.


PROCESSING OF READ COMMANDS


In a non-SNA (locally attached devices--3272 Version) environment, a
read command (Read Buffer or Read Modified) as the 1st byte of the data
stream is processed as follows:

  1.  If the device is in normal-read state, then the command performs a
      read and the display data is transmitted inbound as defined by;

      a.  The Read Modified, or Read Buffer command

      b.  The AID (Read Modified command only)

      c.  The reply mode

The device remains in normal-read state.

  1.  If the device is in a data-pending state, then:

      a.  If the command is Read Modified and INOP specifies Query, the
          appropriate query replies are transmitted.

b. If the command is Read Modified and INOP specifies a Read Modified, then data is transmitted as defined by:

(1) The Read Modified command
(2) The AID
(3) The inbound reply mode

c. If the command is Read Buffer, then data is transmitted as defined by:

(1) The command
(2) The reply mode

Note: For items a, b, and c, the device is placed in the corresponding retry state (Enter, Read, or Stacked Enter).

1. If the device is in a retry state, then the command performs a retry as follows:

a. If the command is Read Modified, and INOP specifies Query, then the appropriate query replies are transmitted inbound.

b. If the command is Read Modified, and INOP specifies Read Modified, then data is transmitted as defined by;

(1) The Read Modified command
(2) The AID
(3) The inbound reply mode

c. If the command is Read Buffer, then data is transmitted inbound as defined by;

(1) The Read Buffer command
(2) The inbound reply

For items a, b, and c, the device remains in the original retry state.


## PROCESSING OF READ PARTITION QUERY STRUCTURED FIELDS


Read Partition Query and the Query Reply are processed as follows:

1. If the device is in normal read state, then

a. The TWAIT indicator is displayed.

b. INOP is set to Query.

c. A channel attention occurs.

d. The device is placed in data-pending-read state.

e. A later read command causes the data to be transmitted and the device to be placed in retry-read state.

2. If the device is in data-pending-enter or retry-enter state, then

a. The outstanding data is stacked.

b. The TWAIT condition remains in effect.

c. INOP is set to Query.

d. A channel attention occurs.

e. The device is placed in data-pending-stacked-enter state.

f. A later Read Modified command causes the data to be transmitted inbound and the device to be placed in retry-stacked-enter state.

# APPENDIX A. SNA SENSE CODES

The SNA sense codes for 3270 data stream errors are listed in Figure A-1. For non-SNA environments, all 3270 data stream errors are rejected with an Op-Check (OC) except for invalid commands. Unsupported commands are rejected with Command Reject (CR). Supported commands that cannot be executed are rejected with an Op-Check.

| | SNA Sense Codes (Hex) | | | |
|---|---|---|---|---|
| Data Stream Error | 1003 | 1005 | 0863 | Other |
| Commands | | | | |
|   Invalid command code | X | | | |
|   Data following read command or EAU | X | | | |
| Orders (valid for all orders) | | | | |
|   Invalid data stream order | X | | | |
|   Incomplete parameter list or parameter missing | | X | | |
|   Invalid code point to be repeated by RA | X | | | |
|   Invalid address | | X | | |
| Start Field Extended | | | | |
|   Invalid attribute type | X | | | |
|   Invalid color or highlighting attribute value | X | | | |
|   Unknown character set attribute value in range X'01'-X'FE' | | | X | |
|   Character set attribute value = X'FF' | X | | | |
| Modify Field | | | | |
|   Current buffer location does not contain field-attribute character | | X | | |
|   Invalid attribute type | X | | | |
|   Invalid color or highlighting attribute value | X | | | |
|   Unknown character set attribute value in range X'01'-X'FE' | | | X | |
|   Character set attribute value = X'FF' | X | | | |

**Figure A-1 (Part 1 of 2). SNA Sense Codes for 3270 Data Stream Errors**

| Data Stream Error | SNA Sense Codes (Hex) | | | |
|---|---|---|---|---|
| | 1003 | 1005 | 0863 | Other |
| Set Attribute<br>  Invalid attribute type<br>  Invalid color or highlighting<br>    attribute value<br>  Unknown character set attribute<br>    value in range X'01'-X'FE'<br>  Character set attribute value =<br>    X'FF' | X<br><br>X<br><br><br><br>X | | <br><br><br><br>X | |
| Set Buffer Address<br>  Address not in form specified for<br>    partition (12-, 14-bit)<br>  Invalid address flag in 12-,<br>    14-bit mode (high-order bits =<br>    B'10') | | X<br><br><br>X | | |
| WCC<br>  Start-printer bit on in WCC in a<br>  structured field that is not the<br>  last structured field of the RU<br>  chain | | | | 1001 |
| Graphic Escape<br>  No alternate character set | | | X | |

Figure A-1 (Part 2 of 2). SNA Sense Codes for 3270 Data Stream Errors

# APPENDIX B. SNA SENSE CODES FOR STRUCTURED FIELDS

The SNA sense codes for structured-field errors are listed in Figure B-1.

The following are included for clarification:

*   The validity of the field attribute is not checked.

*   Write, Erase/Write, Erase/Write Alternate, and Write Structured Field commands without data are accepted without error, but are treated as a no-op.

| | SNA Sense Codes (Hex) | | | |
|---|---|---|---|---|
| Structured-Field Error | 1003 | 1005 | 0863 | Other |
| Structured Fields (All)<br> Invalid structured field type.<br> Missing structured field parameter.<br> Reserved field is not zero. | X<br><br>X | <br>X | | |
| Activate Partition<br> Incorrect length.<br> PID is unknown partition. | | X<br>X | | |
| Create Partition<br> Incorrect length.<br> PID > maximum allowed.<br> Invalid A-mode.<br> Specification error in viewport/<br>  window.<br> Insufficient resource. | <br><br>X | X<br>X<br><br>X<br>X | | |
| Destroy Partition<br> Incorrect length.<br> PID > maximum allowed. | | X<br>X | | |

Figure B-1 (Part 1 of 3). SNA Sense Codes for Structured-Field Errors

| Structured-Field Error | SNA Sense Codes (Hex) | | | |
| --- | --- | --- | --- | --- |
| | 1003 | 1005 | 0863 | Other |
| Load Programmed Symbols | | | | |
|   Incorrect length. | | X | | |
|   Invalid data type. | X | | | |
|   Invalid LCID. (Values X'40' through X'EF' are not considered invalid.) | X | | | |
|   CHAR < X'41' or X'FE'. | | X | | |
|   RWS not in range X'02'-X'07'. | | | | 084C |
|   RWS is valid, but not installed. | | | | 084C |
|   EXTN not supported. | | X | | |
|   P-length incorrect. | | X | | |
|   Byte 8, bits 3-7 not zero. | X | | | |
|   Invalid LH or LW. | | X | | |
|   Byte 12, bits 5-7 invalid. | | | | 084C |
|   Excess bits in data. | | X | | |
|   Compressed data terminator incorrectly specified. | | X | | |
|   PS resource not available. | | | | 084C |
| Outbound 3270DS | | | | |
|   Incorrect length. | | X | | |
|   PID is unknown partition. | | X | | |
|   Invalid CMND byte. | X | | | |
|   Violation of presentation space integrity. | | | | 082B |
|   Data following EAU partition command. | X | | | |
|   WCC print bit = 1, but not last structured field. | | | | 1001 |
| Read Partition | | | | |
|   Incorrect length. | | X | | |
|   PID is unknown partition. | | X | | |
|   Invalid read operation code. | X | | | |
|   PID not equal to X'FF' for Query. | | | | |
|   Read Partition not last structured field in chain. | | X | | |
|   Device in Retry state. | | | | 0871 |
|   Chain containing Read Partition does not specify CD. | | | | 0829 |
|   Chain containing Read Partition does specify EB. | | | | 0829 |

Figure B-1 (Part 2 of 3). SNA Sense Codes for Structured-Field Errors

| Structured-Field Error | SNA Sense Codes (Hex) | | | |
|---|---|---|---|---|
| | 1003 | 1005 | 0863 | Other |
| Reset Partition | | | | |
|   Incorrect length. | | X | | |
|   PID is unknown partition. | | X | | |
| Set Reply Mode | | | | |
|   Incorrect length. | | X | | |
|   PID is unknown partition. | | X | | |
|   Invalid reply mode. | X | | | |
|   Invalid attribute type in A-list. | X | | | |
| Set Window Origin | | | | |
|   Incorrect length. | | X | | |
|   PID is unknown partition. | | X | | |
|   Window specification outside | | | | |
|    presentation space. | | X | | |
|   CW > 0 | | X | | |

Figure B-1 (Part 3 of 3). SNA Sense Codes for Structured-Field Errors

# APPENDIX C. RESET ACTIONS

Figure C-1 summarizes the reset actions performed as a result of the actions listed.

| Action | Partitions | Inbound Reply Mode | Highlighting Color Character Set | | INPID INOP | PS Content PS LCID |
|---|---|---|---|---|---|---|
| | | | Selection | Indicators | | |
| Jump Key | 6 | 1, 3 | NC | 7 | NC | NC |
| CLEAR Key SSCP | 9 | 9 | 9 | 9 | NC | NC |
| CLEAR Key Unowned | R | ID | R | 7 | NC | NC |
| CLEAR Key LULU | R | R | ID | 7 | 9 | NC |
| System Request Key SSCP (Unowned) | 9 | 13 | 9 | 9 | NR | NC |
| System Request Key SSCP (LULU) | 9 | 9 | 9 | 9 | R | NC |
| System Request Key Unowned | R | R | R | 7 | NR | NC |
| Receipt of RU | R | R | R | 7 | NR | NC |
| System Request Key LULU | R | R | ID | 7 | NR | NC |
| Test Key ENTER | R | R | R | 7 | NR | NR |
| Test Key EXIT | 9 | 9 | R | 7 | NR | 15 |
| WCC in EW/EWA | R | R | ID | 7 | R | NC |
| WCC Reset in 3270DS EW/EWA Only | NC | 12 | NC | 11 | 16 | NC |
| Reset Partition | NC | 12 | NC | 11 | 16 | NC |
| Power On | R | 13 | NR | R | NR | R |
| Clear LULU | 10 | 10 | | NA, 10 | 10 | 10 |
| DACTLU LULU | ID | ID | ID | ID | NR | NC |
| DACTLU SSCP | ID | ID | ID | ID | NR | NC |
| ACTLU SSCP | 9 | 9 | 9 | 9 | NR | NC |
| DACTLU Unowned | ID | ID | ID | ID | NR | NC |
| ACTLU Unowned | ID | ID | ID | ID | NR | NC |
| Unbind LULU (8) | 17 | 13 | NC | 7 | NR | NC |
| Unbind SSCP | 9 | 9 | 9 | 9 | NR | NC |
| Bind SSCP | 9 | 9 | 9 | 9 | R | NC |
| Bind Unowned | R | R | R | R | R | NC |
| Set Inbound Reply Mode | NC | 2 | NC | 7 | NC | NC |
| Clear Partition | 4 | 12 | NC | 7 | 16 | NC |
| Destroy Active Partition | 6 | 1 | ID | 7 | 16 | NC |
| Destroy Not-Active Partition | 5 | 5 | NC | NC | 16 | NC |
| Clear Partition Key LULU | NC | NC | NC | NC | 9 | NC |
| CD/EB Write Acknowledgment | NC | NC | NC | NC | R | NC |

ID = Implementation defined (effect visible on external interface).
NA = Not Applicable
NC = No change.
NR = Not relevant (will be changed by subsequent action).
R = Reset (effect visible on external interface).

Figure C-1 (Part 1 of 2). Reset Actions

<u>Notes:</u>

1. The inbound reply mode is set to the mode of the newly activated partition.

2. The inbound reply mode is changed to the mode described in the structured field.

3. The alternate character set is changed to the mode of the newly activated partition.

4. Add the partition name to the list along with its attributes.

5. Delete the partition name from the list along with its attributes. Reset if it is the last partition.

6. Change the active partition to the next in the list. Reset if it is the last partition.

7. Display exactly those attribute selection indicators that are honored as a result of the inbound reply mode in the current partition. If the inbound reply mode is reset, no operator selection is displayed.

8. In the unowned state, the last application write is still displayed (that is, the operator may be able to add data and perform a local copy).

9. The state will already be set.

10. No change. SNA does not permit Clear in the FM layer.

11. Reset if the active partition; otherwise, make no change.

12. Reset the referenced partition only.

13. Change the inbound reply mode to allow all selections.

14. Reset when the test mode alters the programmed symbols

15. Reset if directed to INPID.

16. Screen remains--operator interaction restricted to the active partition.

**Figure C-1 (Part 2 of 2). Reset Actions**

Note that a response of 082B also causes resetting of all the functions shown in the figure, except for PS LCID and PS CONTENT.

## APPENDIX D.   12-, 14-, AND 16-BIT ADDRESSING


The SNA 3270 data stream allows 12-, 14-, and 16-bit addressing.  With
12-bit addressing, an address is created from two bytes of binary
information.  The six low-order bits of each byte are joined to provide
a 12-bit address.  The address specifies the buffer position, not the
line and column position on the display surface.  For example, on a
480-character display, the buffer addresses are 0 to 479.  To specify a
12-bit buffer address of 160 (binary 000010100000), bits 2-7 of the
first byte are set to 000010; bits 2-7 of the second byte are set to
100000:


```
| xx000010 | xx100000 |
```


The 12-bit binary value is a combination of the two 6-bit values of the
first and second bytes.

With 14-bit addressing, an address is created from two bytes of binary
information, but the 14 bits of the address are contiguous.  For
example, an address of 800 decimal (X'320') would be represented as
follows:


```
| xx000011 | 0010000000 |
```


With 16-bit addressing, all bits in both bytes are used.  For example,
an address of 3100 decimal (X'C1C') would be represented as follows:


```
| 00001100 | 000111000 |
```


SNA products (displays) that support only 12-bit addressing ignore bits
0 and 1 of each address byte in outbound data streams.  For inbound data
streams, they should set these bits in accordance with Figure D-1,
although the bits provide no function other than retaining compatibility
with previously written applications.

When a product supports both 12- and 14-bit addressing, bits 0 and 1 of the first address byte are flag bits and have the following significance:

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

For inbound data streams that contain 12-bit addresses, the display generates the B'01' or B'11' setting, using Figure D-1.

A partition may be defined (using the Create Partition structured field) to operate with either 16-bit addressing or 12- and 14-bit addressing. When 16-bit address mode is specified in Create Partition, outbound buffer addresses are interpreted as 16-bit binary and inbound addresses are generated as 16-bit binary. If no partitions are defined, 12- and 14-bit addressing is assumed.

The SNA 3270 data stream can be EBCDIC or ASCII.  With ASCII, each of the allowable characters is a graphic symbol;  that is, there are no unprintable characters in the data stream.  In addition, only 12-bit addressing is used with ASCII.  Figure D-1 shows the 64 binary values permitted, using bits 2 through 7, and defines how they are transformed into ASCII values.  Using the example of 160 decimal as shown above, the 12-bit address becomes X'422D' when bits 0 and 1 of each byte are set in accordance with Figure D-1.

Field attributes and write control characters also require special processing for conversion from binary values to graphic symbols when ASCII or 12-bit addressing is used.

| Bits 2—7 | EBCDIC | ASCII | Bits 2—7 | EBCDIC | ASCII |
|---|---|---|---|---|---|
| 00 0000 | 40 | 20 | 10 0000 | 60 | 2D |
| 00 0001 | C1 | 41 | 10 0001 | 61 | 2F |
| 00 0010 | C2 | 42 | 10 0010 | E2 | 53 |
| 00 0011 | C3 | 43 | 10 0011 | E3 | 54 |
| 00 0100 | C4 | 44 | 10 0100 | E4 | 55 |
| 00 0101 | C5 | 45 | 10 0101 | E5 | 56 |
| 00 0110 | C6 | 46 | 10 0110 | E6 | 57 |
| 00 0111 | C7 | 47 | 10 0111 | E7 | 58 |
| 00 1000 | C8 | 48 | 10 1000 | E8 | 59 |
| 00 1001 | C9 | 49 | 10 1001 | E9 | 5A |
| 00 1010 | 4A | 5B | 10 1010 | 6A | 7C |
| 00 1011 | 4B | 2E | 10 1011 | 6B | 2C |
| 00 1100 | 4C | 3C | 10 1100 | 6C | 25 |
| 00 1101 | 4D | 28 | 10 1101 | 6D | 5F |
| 00 1110 | 4E | 2B | 10 1110 | 6E | 3E |
| 00 1111 | 4F | 21 | 10 1111 | 6F | 3F |
| | | | | | |
| 01 0000 | 50 | 26 | 11 0000 | F0 | 30 |
| 01 0001 | D1 | 4A | 11 0001 | F1 | 31 |
| 01 0010 | D2 | 4B | 11 0010 | F2 | 32 |
| 01 0011 | D3 | 4C | 11 0011 | F3 | 33 |
| 01 0100 | D4 | 4D | 11 0100 | F4 | 34 |
| 01 0101 | D5 | 4E | 11 0101 | F5 | 35 |
| 01 0110 | D6 | 4F | 11 0110 | F6 | 36 |
| 01 0111 | D7 | 50 | 11 0111 | F7 | 37 |
| 01 1000 | D8 | 51 | 11 1000 | F8 | 38 |
| 01 1001 | D9 | 52 | 11 1001 | F9 | 39 |
| 01 1010 | 5A | 5D | 11 1010 | 7A | 3A |
| 01 1011 | 5B | 24 | 11 1011 | 7B | 23 |
| 01 1100 | 5C | 2A | 11 1100 | 7C | 40 |
| 01 1101 | 5D | 29 | 11 1101 | 7D | 27 |
| 01 1110 | 5E | 3B | 11 1110 | 7E | 3D |
| 01 1111 | 5F | 5E | 11 1111 | 7F | 22 |

Figure D-1. Conversion of Binary Values to Hexadecimal Values That Obtain
Graphic Symbols

# GLOSSARY

## A:

activate partition: A structured field used to make a previously defined partition become active. See also destroy partition and create partition.

active partition: The partition in which the cursor is currently located.

AID: Attention identifier

alphanumeric field: A field that can contain any alphabetic, numeric, or special characters.

alphanumeric keyboard: A keyboard used to enter letters, numbers, and special characters into the character buffer of a display device. It is also used to perform special functions (such as backspacing) and to produce special control signals.

attention (ATTN): An occurrence, external to an operation, that could cause an interruption of the operation.

attention identifier (AID): A code that the terminal sends in the inbound data stream to identify the source or type of data that follows. An AID is always sent as the first byte of the inbound data stream. Structured fields in the data stream may also contain an AID.

attribute: A characteristic; for example, attributes of data may include record length, record format, data set name, associated device type and volume identification, use, and creation date.

attribute type: A code that identifies the properties from which an associated set of attribute values can be selected. See also extended color, extended highlighting, character set, and field validation.

attribute value: A code immediately following the attribute type in the data stream that specifies a particular property from the set defined by the attribute type.

automatic skip: On an IBM 3270 Display Station, a feature that, after entry of a character into the last character position of an unprotected display field, repositions the cursor to the first character position of the next unprotected display field.

## B:

base color: The capability to display or print all characters in a field, in one of four colors, on a color terminal by using combinations to the field protection and the field intensify bits of the field attribute.

Binary Synchronous Communications (BSC): See Data Processing Glossary.

blink: An extended highlighting attribute value (for emphasis) of a field or character. See blinking.

blinking: (SC1) In computer graphics, an intentional regular change in the intensity of a display element on a display surface. Synonymous with flashing.

BSC: Binary synchronous communications

buffer: A routine or storage used to compensate for a difference in the rate of flow of data, or time of occurrence of events, when transferring data form one device to another.

buffer address: The address of a location in the character buffer.

**C:**

CCC: Copy control character.

CCP: Current cursor position.

character addressing: Gaining access to any character position by using an address.

character attribute: Defines a single property (for example, extended color, character set, or extended highlighting) of a character(s). A character can have more than one character attribute defined.

character buffer: The Read/Write Storage used by a partition for storing character or graphic data for display or printing on a terminal.

character cell: An addressable location on a display surface used to display or print characters or graphic symbols. For example the 9 x 16 dot matrix on a 3278 Display.

character matrix: Vertical and horizontal representation of a character in a terminal. See character cell.

character set: (1) A defined collection of characters in a loadable or nonloadable set selected by means of a local character set identifier. (2) An attribute type. (3) An attribute passed between session partners in the Start Field Extended, Modify Field, and Set Attribute orders.

codepoint: The specific hexadecimal value selected from an interface code chart.

command: A request from a terminal for the performance of an operation or the execution of a particular program.

control character: (ISO) A character whose occurrence in a particular context initiates, modifies, or stops a control operation. A control

character may be recorded for use in a subsequent action. A control character is not a graphic character, but may have a graphic representation in some circumstances.

control codes: The hexadecimal values "hex 00" through "hex 3F", and "hex FF' in the 3270 data stream.

copy control character (CCC): A character used in conjunction with the Copy command to specify the data to be copied and the type of copy operation to be performed.

copy operation: An operation that copies the contents of the character buffer from one display station or printer to another display station or printer.

create partition: A structured field used to define a new partition. See also activate partition and destroy partition

current cursor position (CCP): The position of the cursor on the presentation space.

cursor: (SC1) In computer graphics, a movable, visible, mark used to indicate a position on a display surface. (2) A movable spot of light on the screen of the display device, usually indicating where the next character will be entered.

**D:**

data-entry keyboard: A keyboard on which the numeric keys are grouped in a format similar to the numeric keys on a keypunch keyboard (to facilitate entry of numeric data).

data stream: All data transmitted through a channel in a single read or write operation.

designator character: A character that immediately follows the field attribute byte in a detectable field.

The designator character controls whether a detection on the field causes an attention. For a nonattention-producing field, the designator character also determines whether the modified data tag for the field is to be set or reset as the result of a detect.

destroy partition: A structured field used to erase a partition and its associated viewport. If the partition being destroyed is the only existing partition, then implicit partition 0 is created. See also Activate Partition and Create Partition.

device: See Data Processing Glossary.

display field: (SC1) An area in the display buffer, or on a screen, that contains a set of characters manipulated or operated upon as a unit.

display surface: In computer graphics, that part of a display device that includes a display space; for example, the entire screen of a cathode ray tube.

do not enter: A symbol (X) in the operator information area that indicates to the operator that the operator should not enter data. See input inhibit and enter inhibit.

dot matrix: (SC1) A matrix of dots used for constructing a display.

E:

EAU: Erase All Unprotected

End of Message: A control code in the 3270 data stream transmitted to a printer marking the end of the data to be printed.

enter inhibit: A condition that prevents the operator from changing the inbound operation or the inbound partition identifier until the terminal receives acknowledgment of

successful receipt of previously transmitted data. Contrast with input inhibit.

Erase/Write: A command that clears the entire character buffer to nulls, positions the cursor to position 0, resets the buffer address to 0, sets character attributes to their default values, and then performs a write operation. It also returns the screen size to the default size.

Erase/Write Alternate (EWA): A command that causes an Erase/Write operation and also switches the size of a display screen or the print capacity of a printer to the alternate size. See also Erase/Write.

Erase All Unprotected (EAU): A command that erases all unprotected fields (inserts nulls),resets the modified data tags in all the unprotected fields, unlocks the keyboard, resets the attention identifier, and repositions the cursor to the first character position in the first unprotected field.

Erase Unprotected to Address (EUA): An order that erases all unprotected character locations (inserts nulls) from the current character buffer address up to but not including, the specified stop address.

ETX: The end-of-text character.

EUA: Erase Unprotected to Address.

EWA: Erase/Write Alternate.

explicit partition: A partition that has been defined by the Create Partition structured field and assigned a partition identifier to distinguish it from other partitions. Contrast with implicit partition 0.

explicit partitioned state: A device state with at least one partition having been explicitly defined using the Create Partiton structured field.

extended color: (1). A capability that allows color terminals to display or print fields or characters in colors using extended field and character attributes. (2). An attribute type.

extended field attribute: Additional field definition to the field attribute that controls defining additional properties such as extended color, character set, extended highlighting and field validation. The extended field attribute is altered by information passed in the Start Field Extended and Modify Field orders.

extended highlighting: (1). A function that provides blink, reverse video, and underscore for emphasizing fields or characters on devices supporting extended field attributes and character attributes. (2). An attribute type. (3) An attribute passed between session partners in the Start Field Extended, Modify Field, and Set Attribute orders.


F:

field: See display field.

field attribute: A control character stored in the character buffer in the first character position of a field. For those devices supporting the 3270 data stream a field attribute defines protected/unprotected, alphanumeric/numeric, detectable/nondetectable, display/nondisplay, intensity, and modified data tag (MDT) on.

field validation: (1). A function that provides mandatory field, mandatory fill, and trigger field capability for checking of data when the data is entered by the operator from a terminal. (2). An extended field attribute type. (3) An attribute passed between session partners in the Start Field Extended, Modify Field, and Set Attribute orders.

fill validation: The process of checking the field to determine if the field is filled with data. Fill validation is performed when the field validation attribute specifies mandatory fill.

formatted display: On a display device, a display in which the attributes of one or more display fields have been defined by the user. Contrast with unformatted display.


G:

graphic escape character: A control code "hex 08" in the data stream used to introduce a graphic character "hex 40" through "hex FE' from an alternate character set.


H:

highlight: In computer graphics, to emphasize a portion of a display image, for example, by blinking.

highlighting: See highlight

home position: The first unprotected character position in the partition's character buffer if the character buffer contains at least one unprotected field. The partition's character buffer address 0 if this buffer contains no fields or no unprotected fields.


I:

implicit partitioned state: The mode of the display when powered on; when it has a single implicitly created partition with the default screen size and a partition identifier of zero.

implicit partition 0: A partition automatically created at power-on or BIND time, or created when all explicitly defined partitions are destroyed and assigned the partition

identifier of 0. Contrast with explicit partition.

inbound operation (INOP): The operation determined by the device characteristics when data is transmitted from the device to the PLU or application program.

inbound partition identifier (INPID): The partition from which data is being transmitted from the device to the PLU or application program.

inbound 3270DS: A structured field used to transmit Start Field, Start Field Extended, Set Buffer Address or Set Attribute orders, and data from a named partition.

INOP: Inbound operation.

input inhibit: (1). A device indicator that informs the operator that further input will not be accepted. (2). A condition whereby operator input from the keyboard or other input devices is not accepted until the input inhibit condition has been reset by the operator or controlling device. Contrast with enter inhibit and do not enter.

INPUT INHIBITED: An indicator on the 3277 and 3275 displays that informs the operator that further input will not be accepted.

INPID: Inbound partition identifier.

insert cursor: An order that moves, if necessary, the cursor to the current buffer address.

intensified field: Data in a field displayed at a brighter level than nonintensified data.

L:

LCID: Local character set identifier.

line control characters: Characters that regulate the transmission of data over a line.

load programmed symbol (Load PS): An outbound structured field used to transmit character definition data for a loadable character set.

Load PS: Load Programmed Symbols

loadable character set: A character set that is not permanently stored in the device. Contrast with nonloadable character set.

local character set identifier (LCID): A value (between "hex 40" and "hex FE") used to identify a character set. The LCID is used by the application program to select a character set for the displaying or printing of data.

local copy operation: The copying of the contents of a display buffer to a printer when the action is initiated by the print key or from the application program.

M:

magnetic slot reader (MSR): A device that reads precoded information from a magnetic stripe as it passes through the slot in the reader.

magnetic hand scanner (MHS): A hand-held device that reads precoded magnetic information from a magnetic stripe.

mandatory field: (1). A field in which the operator must enter one or more characters before an enter action will be permitted. (2). An attribute value controlled by a bit setting in the extended field attribute. (3). An attribute value passed between session partners in the Start Field Extended and Modify Field orders.

mandatory fill: (1) An attribute value controlled by a bit setting in the extended field attribute. (2) An attribute value passed between session partners in the Start Field Extended and Modify Field orders.

mandatory-fill field: A field that the operator must completely fill with data, or press the DUP (duplicate) key before the cursor can be moved from this field.

MDT: Modified data tag.

MF: modify field.

modified data tag (MDT): A bit in the field attribute of a display field, which when set to 1 causes that field to be transferred to the application program during a read modified operation.

Modify Field (MF): An order that specifies the field and extended field attributes to be modified without having to respecify all of the attributes of the field.

multicolored symbol: A programmed symbol of more than one color defined in a triple plane symbol set.

N:

new line character: (1) (ISO) A format effector that causes the print or display position to move to the first position of the next line. (2) Contrast with carriage return.

NL: The new line character.

nonloadable character set: One or more character sets installed in the device that must be used as they are. These character sets cannot be extended or altered by the user. Contrast with loadable character set.

NUL: The null character.

null: (1) Empty. (2) Having no meaning. (3) Not usable.

null character (NUL): (1) (ISO) A control character that is used to accomplish media-fill or time-fill, and that may be inserted into or removed from, a sequence of characters

without affecting the meaning of the sequence; however, the control of equipment or the format may be affected by this character.

null suppression: The bypassing of all null characters in order to reduce the amount of data to be transmitted.

numeric field: A display field that accepts only numeric (0-9), minus, decimal sign, and DUP key entries from a keyboard with the numeric lock feature.

O:

OC: Operation check.

order: A one-byte code in the 3270 data stream instructing a device how to format and define data for display or printing.

order sequence: A string in the data stream that starts with an order and includes one or more data characters associated with the order code.

outbound 3270DS: A structured field used for those portions of the 3270 data stream explicitly directed to a partition.

P:

PA: Program access key.

partition: All or a portion of the usable area of a display surface and its character buffer. Data is presented within the partition boundary through a viewport that is defined when the partition is created. Data is logically stored in the presentation space and that portion of the data that is within the window may be displayed in the viewport.

partition identifier (PID): A one-byte code assigned to a partition by the Create Partition structured field to identify the partition. (It

identifies the presentation space, window, and viewport from which the partition is constructed).

partition wait (PWAIT): A partition related do not enter condition that prevents the operator from keystroking data into that particular partition.

PID: Partition identifier.

program access (PA) key: A key on the keyboard of a display device that causes a program attention without sending any input data.

PF: Program function key.

presentation space: A conceptual two dimensional surface in storage on which data for the partition is represented. See also window. Contrast with partition, viewport, and usable area.

primed field: A trigger field that has had its contents modified by the operator.

program attention key: On a display device keyboard, a key that produces an interruption to solicit program action; for example, a program access key.

program function (PF) key: A key on the keyboard of a display device that passes a signal to a program to call for a particular program operation.

Program Tab (PT): An order that advances the current buffer address to the address of the first character location of the next unprotected field , if it exits, and resets the character attributes of all characters in the field that are replaced by nulls.

programmed symbol: A user-defined character or symbol that can be loaded into the terminal and used for display or printing by the terminal.

programmed symbols (PS): A feature allowing one or more character sets containing up to 190 user-defined characters or symbols.

protected field: On a display device, a display field in which the user cannot enter, modify, or erase data from the keyboard. Contrast with protected field.

PWAIT: Partition wait.

Q:

query reply: An inbound structured field sent in response to a read partition query and used to define the characteristics of the terminal.

R:

RA: Repeat to Address.

Read Buffer (RB): A command which causes all the data in the addressed display buffer to be transmitted to the application program.

Read Modified (RM): A command which causes all of the modified fields (MDT = 1) to be transmitted to the application program.

Read Modified All (RMA): A command which causes all of the modified fields (MDT = 1) and their addresses to be transmitted to the application program.

read modified operation: An operation in which only those display fields that have the modified data tag on (MDT=1) are transferred to the application program.

read partition: An outbound structured field used to read data from a specific partition or to query the device.

read partition query: An outbound structured field that provides the capability for an application program to inquire about the definition of the terminal as to what functions are present and how they are defined.

Repeat to Address (RA): An order that stores a specified alphanumeric or null character and its associated character attributes in character buffer locations, starting at the current buffer address and ending at, but not including, the specified stop address.

reset partition: An outbound structured field that allows resetting the definable parameters of a partition.

reverse video: A form of highlighting a character, field, or cursor by reversing the character, field, or cursor with its background.

RH: Request/response header.

RM: Read Modified

RMA: Read Modified All

RU: Request/response unit.

RWS: Read/write storage.

S:

SA: Set Attribute.

SBA: Set Buffer Address.

scrollable partition: A partition that has a presentation space larger than the viewport.

scrolling: Within a partiton, the continuous vertical movement of the display elements within a viewport in a manner such that the new data appears at one edge of the viewport as old data disappears at the opposite edge. The viewport may include the entire display surface.

selector pen: In computer graphics, a stylus that detects or projects light within the aiming symbol on a display surface.

selector pen attention: An interruption generated when a light pen detect occurs on a display field that has a null or space designator character. The attention concludes the light pen operation.

selector pen detect: The sensing by the light pen of the presence of light from data in a display field that has the detectable attribute.

Set Attribute (SA): An order that specifies an attribute type-value pair defining the property to be applied to subsequent character(s) in the data stream. An SA order is required for each property assigned.

Set Buffer Address (SBA): An order that sets the buffer address to a specified location for the operation that follows.

set reply mode: A structured field that controls operator selectability and sets the parameters for data streams returned.

set window origin: A outbound structured field that changes the position of the window within the presentation plane of the specified partition.

SF: Start Field.

SFE: Start Field Extended.

single plane: A function that restricts the user to defining a symbol using a single buffer. Contrast with triple plane.

Start Field Extended (SFE): An order that defines the start of a data field, with a field attribute, and may include extended field attribute type/value pairs.

structured field: A syntax that permits variable length data to be encoded for transmission in such a way so that the device or program can process a sequence of related parameters without having to scan every byte.

STX: The start-of-text character.

system copy: In the 8775, the copying of the contents of all or a portion of the display surface to a printer that is not attached to the display or display control unit.

system lock: An indicator (X SYSTEM) displayed in the operator information area of the display that indicates to the operator that he cannot enter data from the keyboard into that partition until the condition is reset by the application program or the operator.

T:

terminal wait (TWAIT): A device related do-not-enter condition that prevents the operator from entering data from the keyboard.

TH: The transmission header.

trigger field: (1). An input field which, when modified by the operator, provides an automatic enter operation as the operator attempts to move the cursor out of this field. (2). An attribute value controlled by a bit setting in the extended field attribute. (3). An attribute value passed between session partners in the Start Field Extended and Modify Field orders.

triple plane: A function that allows the user to define a whole symbol or portions of a symbol in the same address location in three different buffers. See also triple plane symbol set.

triple plane symbol set: A Programmed Symbol set that has a portion of a symbol or the whole symbol defined in each of the primary color planes, (red, blue, and green) allowing the user to display or print a whole symbol in one color, multiple colors, or a blend of colors.

TWAIT: Terminal wait.

U:

unformatted display: On a display device, a display in which no display field has been defined by the user. Contrast with formatted display.

unprotected field: On a display device, a display field in which the user can enter, modify, or erase data from the keyboard. Contrast with protected field.

Usable Area: (1). The area on a display surface that can be used to display data. (2). Data in a query reply structured field that defines the size and characteristics of the screen available for defining viewports.

V:

viewport: A rectangular area on the usable area of the display surface through which the operator views all or a portion of the data outlined by the window on the presentation plane.

W:

WCC: The write control character.

window: That portion of the data in the presentation space that may be viewed in a viewport on the display surface.

wraparound: (SC1) In computer graphics, the continuation of an operation, such as a read or a cursor movement, from the last character position in a display buffer to the first position in the display buffer.

write control character (WCC): A control character that follows a write command in the 3270 data stream providing control information to the control unit for executing display and printer functions.

write control character reset (WCC reset): Bit one of the write control character is the reset bit and when set to one it resets the characteristics of the operation to their system defined values.

Write Structured Field (WSF): A command used to transmit data in structured field format.

WSF: Write Structured Field

3270 data stream: The commands, control codes, orders, attributes, and data or structured fields transmitted in s fixed sequence inbound to an application program or outbound to a terminal.

link trailer (LT) 1-1
loadable character set 4-12
load programmed symbol set
 structured
 field 1-5,7-1,7-8,7-13
Load PS (see load programmed
 symbol set structured field)
local character set identifier
 (LCID)
    general 7-10
    in Load PS 7-10,7-12
    printers 6-7
local copy function
    in BSC 6-8 - 6-11,8-10
    in non-SNA 6-3
logical screen 2-1
longitudinal redundancy check
 (LRC) 5-1,5-9,5-10
LRC (see longitudinal redundancy
 check)
LT (see link trailer)


magnetic readers 5-1
Magnetic Reader
 Operations 5-1,5-2
magnetic stripe 5-1
magnetic stripe input 5-1
Management of Presentation
 Spaces 2-7
mandatory field 4-23,7-28
mandatory fill 4-22,7-28
mandatory fill-field 4-22
mapping of the character
 buffer 1-2
modified data tag (MDT)
    generally 3-13
    resetting 3-3
Modify Field (MF) order 4-6 - 4-7
multicolor 4-20,4-28
multiple partitions 2-4


neutral 7-26
new line control character
 (NL) 4-11,6-1,6-2
NL (see new line)
nondetectable 1-8,4-14
Nondisplay or Dispaly/Intensified
 Display 1-8
nonsecure data 5-6,5-7,5-9
Nonsecure Magnetic Stripe
 Cards 5-10

non-SNA 3270
    environment 9-1
    generally 1-1,9-1
nonloadable character set 4-12
nontransparent mode 8-1
normal read state 3-19,8-6,9-4
NUL 4-11
nulls 4-10
Numeric/Alphanumeric
 Operation 5-6
numeric character set 5-6
numeric/alphanumeric character
 set 5-6,5-9
numeric field
    unprotected 5-12


OC (see operation check)
OID (see operator ID)
operation check (OC) A-1,B-1
Operator Enter Actions 3-16
operator ID (OID) 5-1
orders, 3270
    Buffer control 4-1,4-2
    codes for 4-2
    formats 4-3
    generally 1-3 - 1-4,4-1
    Erase Unprotected to Address
     (EUA) 1-4
    Insert Cursor (IC) 1-4,4-8
    Modify Field
     (MF) 1-4,4-6 - 4-7
    Program Tab (PT) 1-4,4-8
    Repeat to Address (RA) 1-4,4-9
    Set Attribute (SA) 1-4,4-5
    Set Buffer Address
     (SBA) 1-4,4-4
    Start Field (SF) 1-4
    Start Field Extended (SFE) 1-4
outbound data stream 1-2
outbound transmissions 8-8
Outbound 3270DS structured
 field 1-5,3-2,7-1,7-20,7-21
overview 1-1


PA (see program attention key)
paging (see Program Tab (PT)
 order)
parameter list (PLIST) 7-22
partition identifier (PID) 2-4
partitions
    generally 2-1 - 2-10
    scrolling in 5-17

IBM 3270 Information Display System
Data Stream
Programmer's Reference

READER'S
COMMENT
FORM

Order No. GA23-0059-0

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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

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

How did you use this publication?

[ ]   As an introduction          [ ]   As a text (student)
[ ]   As a reference manual        [ ]   As a text (instructor)
[ ]   For another purpose (explain)   _____

_____

Is there anything you especially like or dislike about the organization, presentation, or writing in this manual? Helpful comments include general usefulness of the book; possible additions, deletions, and clarifications; specific errors and omissions.

   Page Number:                    Comment:

What is your occupation? _____

Newsletter number of latest Technical Newsletter (if any) concerning this publication: _____

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

IBM branch office serving you _____

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

Note:   Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

**Reader's Comment Form**

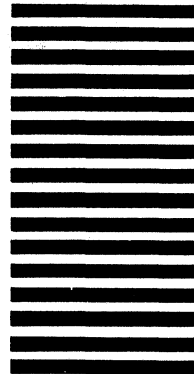Fold and tape                        **Please Do Not Staple**                    Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

# BUSINESS REPLY MAIL

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

POSTAGE WILL BE PAID BY ADDRESSEE:

**International Business Machines Corporation**
**Department 52Q**
**Neighborhood Road**
**Kingston, New York 12401**

Fold and tape                        **Please Do Not Staple**                    Fold and tape
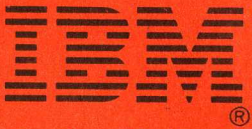
# IBM ®

**International Business Machines Corporation**
**Data Processing Division**
**1133 Westchester Avenue, White Plains, N.Y. 10604**

**IBM World Trade Americas/Far East Corporation**
**Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591**

**IBM World Trade Europe/Middle East/Africa Corporation**
**360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601**

Cut or Fold Along Line

IBM 3270 Information Display System Data Stream Programmer's Reference (File No. S360/S370/S3/4300/8100-30)  Printed in U.S.A.  GA23-0059-0

GA23-0059-0

**IBM**

®