Chapter 7. Magnetic Slot Reader, Keyboard, and Selector Pen Operations

Introduction 7-2
Magnetic Slot Reader Operations 7-2
  3275/3277-Compatible Operation 7-2
  Numeric/Alphanumeric Operation 7-6
    Stripe Codes and Application Program Codes 7-7
    Secure/Nonsecure Magnetic Stripe Cards 7-9
  Test Card 7-9
Keyboard Operations 7-10
  Keys that Affect the Data Stream 7-10
    Cursor 7-10
    Alphanumeric Data 7-10
    Automatic Skip 7-10
    Erase to End of Field (Erase EOF) Key 7-11
    Erase Input Key 7-11
    Insert Mode Key 7-11
    Delete Key 7-12
    Duplicate (DUP) Key 7-12
    Field Mark Key 7-13
    Program Attention Keys 7-13
  Clear Partition 7-13
  Clear 7-13
  Error Override 7-14
  Keyboard Actions with Attribute Selection Keys 7-15
  Keyboard Actions in Partitions 7-15
  Scrolling Partitions 7-16
    Vertical Scrolling 7-16
    Keyboard Actions and Scrolling 7-17
    Action for Data Entry Keystrokes 7-17
    Automatic Scrolling 7-17
Selector Pen Operation 7-17
  Selector Pen Field Format 7-18
  Designator Characters 7-18
    Selection 7-18
    Attention 7-19
  Selecting Fields in Partitions 7-19

Introduction

This chapter describes the magnetic slot reader (MSR), keyboard, and selector pen operations. The discussion of MSR also applies to the magnetic hand scanner (MHS).

Magnetic Slot Reader Operations

This section describes two modes of MSR 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 can contain up to 128 characters. These characters include the start of record (SOR), which does not appear in the input data stream. Also included are the EOR and the longitudinal redundancy check (LRC), which appear in the input data stream. The application program must verify that X'F7' appear just prior to the LRC.

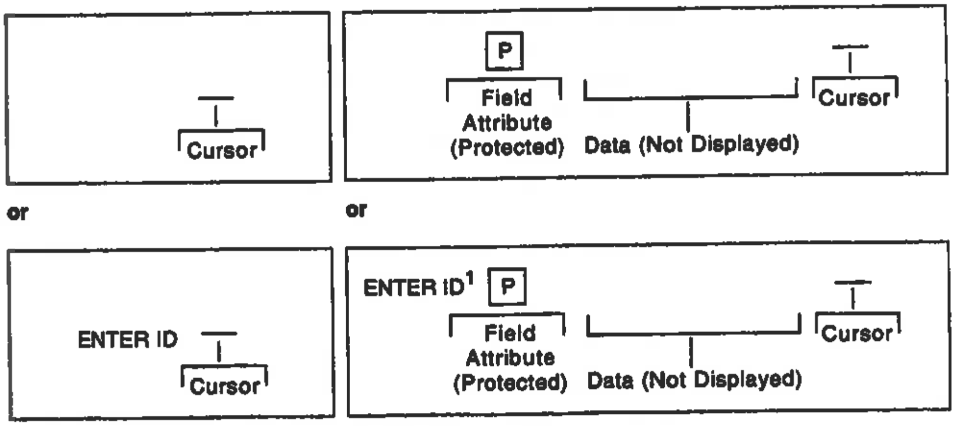
The data portion of the stripe, the part between the initial start of record and the EOR, should contain only 0-9, *D* (field separator), or the OID (which is the *A* immediately following the SOR indicating secure data). The secure data is for the application program's use; the display treats all stripes 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 SOR 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. The cursor may not be visible to the operator until the end of the operation. An MSR operation automatically causes the keyboard to lock and an MSR 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 an MSR operation. There are some differences in the data-stream content, depending on whether the device buffer is formatted or unformatted:

- Unformatted display. When an MSR input operation occurs, the device buffer becomes formatted because of the insertion of the protected, nondisplay attribute with the MDT bit set on. Any data that was previously displayed will no longer be displayed. (See Figure 7-1 on page 7-4.)
- Formatted display. An MSR operation may be initiated only when the cursor is located in an unprotected field. The input data stream resulting from an MSR 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 SOR. Figure 7-2 on page 7-5 and Figure 7-3 on page 7-6 show how to prepare an unprotected field for MSR input. In Figure 7-2, the cursor follows the operator instructions. In Figure 7-3, the instructions are in a protected field and the cursor follows an unprotected field attribute character.

When the extended field and character attributes are supported, an MSR operation causes an extended field attribute of X'00' to be generated in addition to the field attribute. Data characters generates character attributes of X'00'.



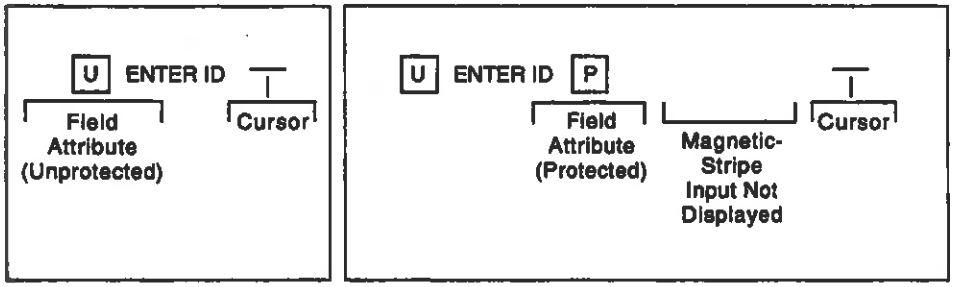
Display Screen before Magnetic-Stripe Input is Entered

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	
Data	The magnetic-stripe data followed by any information present in the buffer. The additional information could have been initiated by the application program as ENTER ID, as shown in the example, or entered by the operator before the magnetic-stripe operation was started. Note that, with an unformatted screen, the magnetic-stripe data is the first text in the data stream presented to the application program.

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

Figure 7-1. Display Screen Activity Before and After Magnetic-Stripe Input (Unformatted Display)

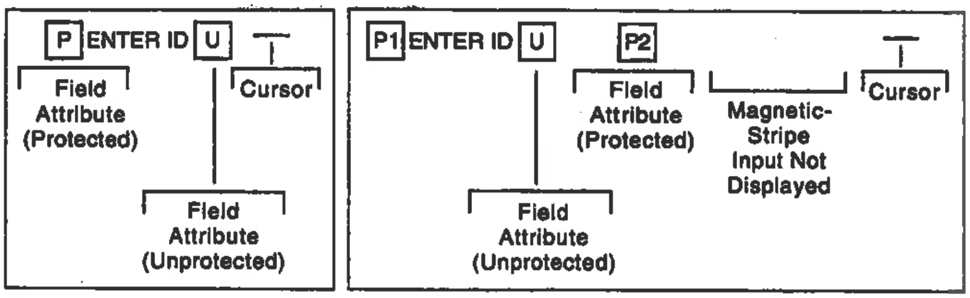

U
ENTER ID
Field Attribute (Unprotected)
Cursor
U
ENTER ID
P
Field Attribute (Protected)
Magnetic-Stripe Input Not Displayed
Cursor

Display Screen before Magnetic-Stripe Input is Entered

Display Screen after Magnetic-Stripe Input is Entered

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

Figure 7-2. Display Screen Activity Before and After Magnetic-Stripe Input (Formatted Display with Unprotected Field Attribute)


P
ENTER ID
U
Field Attribute (Protected)
Cursor
Field Attribute (Unprotected)
P1
ENTER ID
U
P2
Field Attribute (Protected)
Magnetic-Stripe Input Not Displayed
Cursor
Field Attribute (Unprotected)

Display Screen before Magnetic-Stripe Input is Entered

Display Screen after Magnetic-Stripe Input is Entered

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

Figure 7-3. Display Screen Activity Before and After Magnetic-Stripe Input (Formatted Display with Protected Field Attribute)

Numeric/Alphanumeric Operation

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

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

Note: * = 0 through 9 or D.

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

For the alphanumeric character set magnetic stripes, the data field starts after the two-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 (ES) 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 table shows the relationship of the magnetic stripe code to the codepoint that 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 U.S. EBCDIC character set displays as would be generated for a German character-set display. The graphic displayed when display is allowed depends on the graphic character sets supported by the attaching display.

Following is the magnetic-stripe numeric character set:

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

Control Data Characters (Hex)	
A (secure data)	Not sent in the data stream
B (SS)	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 (ES)	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 the SS. The *F* is the ES.

Another SS, called the *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 RSS. This type of operation is primarily for MHS operations.

Following is the alphanumeric character set:

Stripe Codes	Display-Generated Codes	
Character	**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
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
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 *A* is a filler character when numerics are not paired.

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

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

The initial *B* is SS. The *F* is the ES. The RSS follows the LRC on the magnetic stripe card. The RSS is used by devices using a magnetic stripe card reader that can scan the card in either direction.

Any stripe code not shown (except for CAEE—the test card) causes a rejection of the card when that code appears between the header and the ES. The first *F* encountered after the SS is treated as the ES. The code following the ES is considered the LRC.

Secure/Nonsecure Magnetic Stripe Cards

With regard to the application program, display operations that use the numeric/alphanumeric character set MSRs closely parallel the 3275/3277-compatible operation. However, secure and nonsecure cards are handled differently.

Secure Magnetic Stripe Cards: Handling of secure numeric/alphanumeric magnetic stripe cards is similar to the handling of cards during the 3275/3277-compatible operation. The same rules apply to the cursor when it is in an unprotected field, to the generation of a protected, nonprint/nondisplay field attribute, and to automatic entry. An AID code (E7 for EBCDIC, 58 for ASCII) is generated that indicates that the MSR is extended. This code informs the application program that the magnetic stripe data conforms to the numeric/alphanumeric character set operation. Nothing indicates whether the data field is in the numeric or alphanumeric character set. No ES or LRC is sent with the data. When the application receives the E7,(58), AID, the data is valid; the LRC check has been completed.

Nonsecure Magnetic Stripe Cards: Nonsecure numeric/alphanumeric character set magnetic stripe cards appear to the application program as data entered using the keyboard. 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 can be entered in any unprotected field. The cursor must still be in an unprotected field for a successful MSR operation. However, no protected, nonprint/nondisplay field attribute is generated. No AID is generated, and no automatic entry takes place. The sending of the data relies on normal entry operation—for example, the Enter key or selector pen attention. An implementation may optionally provide for a customer selected override that causes auto entry of nonsecure data. However, the AID generated is coded as the Enter key.

Test Card

For displays, any nonsecure card of known content can be used as a test card. The alphanumeric test card (CAEE—) is treated as a normal nonsecure card, except that the EE is deleted. There is no automatic verification of the test data. The operator performs visual verification. All MSR data is inhibited from being sent to the application program when the display is in subsystem test mode.

Keyboard Operations

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

Keys that Affect the Data Stream

The data stream is oriented toward a keyboard display, but during implementation, the user has the option of providing or not providing a keyboard. If supported, the key functions and key positions depend on the type of keyboard and the keyboard definition that the user provides. The keyboard description that follows is limited to the functions that have an effect on the data stream.

Cursor

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

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

Alphanumeric Data

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

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

Automatic Skip

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

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

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

Erase to End of Field (Erase EOF) Key

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

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

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

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

Erase Input Key

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

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

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

Insert Mode Key

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

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

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

After all null characters at or beyond the cursor location in the field have been overwritten, or if there were no null characters, pressing an alphanumeric key locks the keyboard and notifies the terminal operator of an input inhibit condition. The field attribute and the extended field attribute remain unchanged.

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

Operation of the Reset key returns the keyboard from insert mode to normal mode.

Delete Key

If the cursor is located in an alphanumeric character location in an unprotected field, pressing the Delete key deletes the character from the character location occupied by the cursor and sets the MDT bit to 1. The cursor does not move. All remaining characters in the unprotected field (to the right of the cursor and on the same row) are shifted one character location to the left. The vacated character location at the end of the row is filled with a null and the default character attributes. If the unprotected field encompasses more than one row, characters in rows other than the row identified by the cursor may or may not be affected.

An implementation may choose to limit the delete to the current row or may choose to allow the delete local function to affect the whole field which may span many rows.

The character attributes of any shifted characters do not change.

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

Duplicate (DUP) Key

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

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

Field Mark Key

Operation of this key causes a X'1E' code to be entered into the active partition buffer and the MDT bit to be set to 1. The field mark character informs the application program of the end of a field in an unformatted buffer or subfield in a formatted buffer. The field mark character transferred to the application program is X'1E' (EBCDIC) and is sent when the data is read. The field mark character, when stored in the character buffer, is displayed as semicolon-overscore. Display devices operating in monocase mode may also display the field mark character as a semicolon.

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

Program Attention Keys

These keys solicit application program action by causing the sending of an AID character to identify the key. The MDT bit is not affected. The program attention keys are Clear, Enter, Cancel, all program function (PF) keys, and the program access (PA) keys.

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

Clear Partition

The effect of invoking this function depends on whether the display is in implicit partition state or in explicit partition state.

The operation of the Clear Partition function does the following:

- Sets all buffer locations for the active partition to nulls and all character attributes to X'00'.
- Sets the reply mode for the active partition to the default (field reply mode).
- Causes an AID (of value X'6A') to be transmitted inbound in a Short Read operation.
- Moves the current cursor position to the top left corner of the presentation space (buffer address 0). In a scrollable partition, this can cause automatic scrolling to occur.

Invoking this function while a previous inbound transmission is being processed locks the input device, and signal an input-inhibited condition to the operator on the indicator row of the display.

Clear

If the display is in implicit partition state, invoking the Clear function results in the following:

- All buffer locations are set to nulls.
- All character attributes are set to their default values.
- The MDT bit is set to 0.
- The reply mode is set to the default (field reply mode).
- An AID of X'60' is transmitted inbound in a short read operation.

- The implicit partition is either set to its default size or remains the same depending on the customization value selected or the option that has been designed into the product.
- The cursor is moved to the current cursor position, buffer address 0.

If the device is in explicit partitioned state, invoking the Clear function results in the following:

- All existing partitions are destroyed.
- A partition with a PID of 0 and default size is created.
- The device is reset to implicit partition state.
- The cursor is moved to the current position, buffer address 0.
- All buffer locations are set to nulls.
- All character attributes are set to their default values.
- The MDT bit is set to 0.
- The reply mode is set to the default (field reply mode).
- An AID of X'6D' is transmitted inbound in a short read operation.

Error Override

Operation of this key when the cursor is in an unprotected field causes a special substitute (SUB) character with a value X'3F' to be entered into the active partition buffer at the current cursor position. The SUB character provides a means of informing the application program that there is an error in the field.

The cursor is advanced to the next position and the MDT bit is set to 1. The character is displayed as a solid circle.

Invoking this function when the cursor is in a protected field locks the keyboard and notifies the terminal operator of an input-inhibit condition.

With the cursor in an unprotected field of an unprotected partition, or in an unprotected partition that is unformatted, invoking this function causes the following actions:

- A special substitute (SUB) character value X'3F' is entered into the active partition buffer at the current cursor position. The SUB character provides a means of informing the host that there is an error in the field.
- The MDT bit for the field containing the cursor is set to 1.
- The cursor position is advanced by one position as for normal data entry.
- The character displays as a solid circle.
- The character attribute is updated according to the highlighting and color currently specified.

Invoking this function when the cursor is in a protected partition at an attribute location or in a protected field raises the Input-Inhibited-Wrong-Place condition.

Invoking this function when the Field Validation display function is not in use raises the Input-Inhibited-Minus Function condition.

Invoking this function while the input device is owned by the SSCP-LU session locks the input device and signal an input-inhibit condition to the operator on the indicator row of the display.

Keyboard Actions with Attribute Selection Keys

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

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

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

Keyboard Actions in Partitions

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

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

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

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

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

The effect of the Delete, Erase EOF, and Erase Input keys is constrained to the active partition; that is, no deletions or storing of nulls occurs outside the active partition. The Clear Partition key clears the active partition. The Clear key clears the entire screen by destroying all partitions, placing the device in implicit partition state, and recreating an implicit partition 0 of default size.

Scrolling Partitions

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

When a partition is created, it can be set up as a scrollable partition by making the presentation space larger than the viewport. The viewport seen by the operator has a one-to-one relationship with a window on the corresponding presentation space. The position of the window on the presentation space determines what data is seen in the viewport. The initial position of the window on the presentation space is specified by the Create Partition structured field. The position of the window on the presentation space is reset using the Set Window Origin structured field. The number of rows moved during scrolling is determined by the RS Value given in the Create Partition structured field.

Row-by-row vertical scrolling within the presentation space is provided in response to the operator's pressing the keys for the scroll up and scroll down. Vertical scrolling is achieved by moving the position of the window within the presentation space in response to the use of the keys for scrolling.

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

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

Vertical Scrolling

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

The previous top row is lost from the window; all other rows are moved up.

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

If the window is already positioned at the bottom of the presentation space, then scrolling up has no effect. There is no input-inhibit condition and no indicator.

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

The previous bottom row is lost from the window; all other rows are moved down.

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

If the window is already positioned at the top of the presentation space, then scrolling down has no effect. 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 Scrolling

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

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

Selector Pen Operation

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

Selector Pen Field Format

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

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

Designator Characters

Designator characters are used to define two types of selector pen fields: selection fields and attention fields. Each type of field performs a different selector pen operation. If the field attribute defines a field as selector pen detectable but the character in the designator character position is not a valid designator, a detect cannot be made on the field.

Selection

The selection field is defined by a question mark (?) designator character (X'6F'). When the selector pen detects on a selection field, the MDT bit in the field attribute for the field is set to 1, and the X'6F' is changed to X'6E'. Also, the designator character is automatically changed in the character buffer and on the display surface to show to the operator that the detection was successful.

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

The field may be deselected by using the same technique as for selection.

If the designator is > (X'6E') and a detect is made on the field, the designator is changed to ? (X'6F') and the MDT bit is set to 0, and the X'6E' is changed back to X'6F'.

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 can support the ampersand (&) designator character to provide an Enter key simulation.

A detection on an attention field designated by a space or null causes an AID X'7E' to be sent to the application program that identifies the source of entry to be the selector pen. Also sent are the addresses of fields with the MDT bit set to 1 and the current cursor position for the partition that contains the selected field. The application program responds, usually by issuing a Read Modified All, to obtain the contents of modified fields and the addresses.

A detection on an attention field designated by an ampersand results in an AID X'7D' being sent that identifies the source of entry to be the Enter key. Also sent are the addresses and contents of all fields with the MDT set to 1 and the current cursor position for the partition that contains the selected field. If the application defines an attention field with an ampersand designator and the implementation does not support the ampersand, the selector pen is inoperative on the field; the operator must then use the Enter key.

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

Selecting Fields in Partitions

The operator can use the selector pen to select fields in any partition, 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. This need not be the active partition.

Chapter 8. Printer Operations

Introduction . . . 8-2
Local Copy Function in an SNA Environment . . . 8-4
Copy Initiation . . . 8-5
Printer Availability . . . 8-5
Display/Printer Compatibility . . . 8-7
APL Mismatch . . . 8-7
Character Attribute/Extended Field Attribute (CA/EFA) Mismatch . . . 8-7
Programmed Symbols (PS) Considerations . . . 8-7
Extended Color Mismatch . . . 8-8
Extended Highlighting Mismatch . . . 8-8
Partition Mode Considerations . . . 8-9
Local Copy Command in the BSC Environment . . . 8-9
Valid LU 1 SCS Control Codes . . . 8-11
Early Print Complete (EPC) Operations . . . 8-13
IPDS Data/Non-SNA . . . 8-15
IPDS Selection . . . 8-15
Local Copy . . . 8-16
Pacing . . . 8-16
Input Transmissions . . . 8-16

Introduction

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

Commands The read commands are not valid in the 3270 data stream received by the printer in an SNA environment. If received, the request is rejected (sense code X'1003'). See also the *Structured Field* exception below. However, the read commands are valid if received in the 3270 data stream in a non-SNA environment for locally attached 3274D models (or 3272 version) or BSC attached devices.

Structured fields Structured fields that are not supported and structured fields with invalid values are rejected, as follows:

- The Create Partition, Activate Partition, and Set Window Origin structured fields are not supported.
- The Read Partition structured field is limited to query operations in an SNA environment. The Read Modified command is supported only for sending Query Replies to an application.

Selector pen Not applicable to printer.

WCC byte The WCC byte for printer use is defined as follows:

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

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.

EM (End of Medium) 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 an 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 they are 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. Therefore, 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 NL 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 NL is not executed, because the printer is already positioned at the left margin for the next printout.

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

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

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

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

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

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

Local Copy Function in an SNA Environment

A hard copy of the display screen or a portion of a display screen can be obtained 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 resulting 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 has the capability for selection of monocase/dualcase without requiring operator action, the printed data must match the case of the displayed data. It is not necessary for the printer to recognize any manual override by the display of uppercase and lowercase.
- Color may not match.

The local copy operation is described on the basis of a nonpartitioned screen; that is, for a display that does not have partitioning capability or for a display with partitioning capability that is in implicit partition state (implicit partition 0). Differences due to partitioning are described under "Partition Mode Considerations" on page 8-9.

Copy Initiation

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

Printer Availability

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

A logical printer is considered available if one or more of the physical printers, configured for the display's use, can immediately execute a display printout. When none of the configured printers is available, the logical printer assumes the status of the most available physical printer. The following are the not-available categories listed in order of best to worst:

1. Short-term busy. A display printout can be executed after *n* queued display printout requests are executed. The maximum delay is 30 minutes. Use by a session is not allowed.
2. Intervention-required condition.
3. Allocation for session use or local use (not copy).
4. Permanent error condition (or an intervention-required situation on an unattended printer).
5. No logical printer configured.

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

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

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

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

0801 No printer configured.

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

- The LUSTAT 0001 (source=printer) reporting the clear of a 082E or 0807 condition is not sent if either of the following is true:
  - After sending 082E/0807 -rsp, the Data Traffic Subtree is reset or the session is terminated.
  - After sending 082E/0807 -rsp, the logical printer develops a permanent error or becomes not configured.

    An LUSTAT 081C/0801 (source=printer) is sent instead of LUSTAT 0001.
- The LUSTAT 0001 (source=printer) does not indicate the host will necessarily find an available or short-term-busy logical printer if one of the following is true:
  - After sending 082E/0807 -rsp, the SLU receives any normal FM data request chain other than a Copy request chain.
  - After sending LUSTAT 0001 (source=printer), the logical printer develops an intervention-required, permanent error, or not-configured condition. The Copy request will be rejected with the appropriate -rsp, that is, 082E, 082F, or 0801.
  - It is not valid for an implementation to allow session contention for a printer prior to honoring outstanding LUSTATs; that is, at least one physical printer must be held after having sent the LUSTAT(s) so that the logical printer is available to execute copy sessions. However, if as a result of an exception condition developing on the held printer the logical printer status of in session results, then 0807 -rsp may also be sent.
  - After sending 082E/0807 -rsp, the display detects a nonprinter exception condition which causes 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 is not inhibited because of a mismatch in display/printer capability. However, a copy operation can be inhibited if the printer is unable to accept all the display data; for example, if the print buffer is too small, or if the printer maximum line length is less than the 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 occurs 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 can send only the required information or can 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 would send only the color information from the CA/EFA or all the CA/EFA information.

Programmed Symbols (PS) Considerations

A copy operation is not 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 can 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.

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 is not 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 is not 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 WSF command contains multiple structured fields, the WCC=SP can be set only in the last structured field. If set in any other structured field, the WSF 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 W, EW (reset off), or an EWA (reset off) is sent with WCC=SP to a partitioned screen, a printout of partition 0 occurs. 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 is transferred. The *from* terminal, the source of the buffer data to be copied, is identified in the second two bytes that follow the Copy command code; the first byte, called the *copy control character* (*CCC*), identifies the type of data to be copied. The CCC can also, at the *to* device, start print operations, specify the printout format for those operations, and, when the terminal is a display station, sound the audible alarm.

The copy data stream is shown in Figure 8-1 on page 8-10.

Table 8-1 on page 8-11 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 an error status.

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

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

Notes:

1. Copy should not be chained from a W, EW, EWA, or EAU command, since it copies the data as modified by the W or E 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 controller 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 causes an operation check.

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

- The source terminal is a display.
- The destination terminal is a printer.
- The source terminal does not have a protected/alphanumeric field attribute in the first buffer position.
- The CCC has bits 4, 6, and 7 (start print and copy entire buffer) set to 1.

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

Byte	
0	Start of text
1	Escape
2	Copy command code
3	CCC (see the following table)
4	Source device address
5	End of text

CCC-Byte Format

*	1	Printout Format	Start Print	Sound Alarm	Type of data to be copied
0	1	2 3	4	5	6 7

* Determined by the configuration of bits 2 through 7

Figure 8-1. The Copy Data Stream

Table 8-1. Copy Control Character (CCC)

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

Valid LU 1 SCS Control Codes

SCS control codes that are valid for the LU 1 SCS Data Stream are defined by the acceptance of LU 1 BIND byte 18 (Data Stream Flags) and by indications in certain Query Replies, such as the descriptors returned in the Device Characteristics Query Reply. The valid SCS control codes are listed in Table 8-2. All other SCS control codes are invalid for LU 1 usage.

Table 8-2 (Page 1 of 2). Valid LU 1 SCS Control Codes

SCS Control Codes for LU 1	EBCDIC Code
Vertical Channel Select (VCS)	X'04'
Horizontal Tab (HT)	X'05'
Graphic Escape (GE)	X'08'
Vertical Tab (VT)	X'0B'
Form Feed (FF)	X'0C'
Carriage Return (CR)	X'0D'
Enable Presentation (ENP)	X'14'
New Line (NL)	X'15'
Backspace (BS)	X'16'
Interchange Record Separator (IRS)	X'1E'
Inhibit Presentation (INP)	X'24'

Table 8-2 (Page 2 of 2). Valid LU 1 SCS Control Codes

SCS Control Codes for LU 1	EBCDIC Code
Line Feed (LF)	X'25'
Set Attribute (SA)	X'28'
Format (FMT)	X'2B'
Set Horizontal Format (SHF)	X'2BC1(L)(P)'
Set Vertical Format (SVF)	X'2BC2(L)(P)'
Set Line Density (SLD)	X'2BC6(L)(P)'
Set Text Orientation (STO)	X'2BD1(L)83(P)'
Set Print Density (SPD)	X'2BD2(L)29(P)'
Page Presentation Media (PPM)	X'2BD2(L)48(P)'
ASCII Transparent (ATRN)	X'2BFE(L)35(P)'
BELL (BEL)	X'2F'
Transparent (TRN)	X'35(L)(P)'

Note:
L = length
P = parameters	

The hierarchy of support allowed by the Data Stream Flags in byte 18 of the LU 1 Bind allows for the selection of the SCS control codes that a device chooses to support. Table 8-3 describes this hierarchy.

Table 8-3 (Page 1 of 2). LU 1 BIND Byte 18 (Data Stream Flags)

Bit	Content Description
0	**B'0'** Base NL (New Line) FF (Form Feed) **B'1'** Full Base includes Base plus: BS (Backspace) CR (Carriage Return) LF (Line Feed) ENP (Enable Presentation) INP (Inhibit Presentation) HT (Horizontal Tab) VT (Vertical Tab)
1	SHF (Set Horizontal Format) **B'0'** SHF must not be used **B'1'** SHF may be used
2	SVF (Set Vertical Format) **B'0'** SVF must not be used **B'1'** SVF may be used
3	VCS (Vertical Channel Select) **B'0'** VCS must not be used **B'1'** VCS may be used

Table 8-3 (Page 2 of 2). LU 1 BIND Byte 18 (Data Stream Flags)

Bit	Content Description	
4	SLD (Set Line Density)	
	B'0'	SLD must not be used
	B'1'	SLD may be used
5	Reserved	
6	BEL (Bell)	
	B'0'	BEL must not be used
	B'1'	BEL may be used
7	TRN (Transparent) and IRS (Interchange Record Separator)	
	B'0'	TRN and IRS must not be used
	B'1'	TRN and IRS may be used

Support for other SCS control codes is indicated as follows:

- Graphic Escape (GE) support is indicated by setting byte 4, bit 1 to B'1' in the Character Sets Query Reply.
- Set Attribute (SA) support is indicated by returning the value, character mode (X'02') in one of bytes 4 to n in the Reply Mode Query Reply which commits support of character attributes.
- Set Line Density, Set Print Density, Page Presentation Media, Set Text Orientation, and ASCII Transparent support is indicated by descriptors in the Device Characteristics Query Reply.

Early Print Complete (EPC) Operations

When page printers wait until the page is printed before returning a positive response, they will not run at their rated speed. The remedy is to allow them to return a positive response when the data has been received without error in the print buffer, but before the data is printed.

There exists a hazard that an error may occur during the printing of the last page of a print file. If this happens, the error will be reported on the next print job, not the one on which the error occurred, because the printer has responded that the print job has been successfully completed.

The architecture for Early Print Complete (EPC) was created to allow printers to run at their rated speed, while avoiding the erroneous response. EPC mode allows overlap of the loading of the printer buffer and the printing operation. Support of EPC mode is limited to the non-SNA environment (BSC and non-SNA local channel). In the SNA environment, LU1 may be used if the serial load and print required by LU3 gives performance problems with high speed printers.

EPC mode is set on or off by the SREPC flag (located in the Early Print Complete self defining parameter of the Settable Printer Characteristics structured field). Optionally, the printer may provide a printer operator control (eg. switch) in addition to the host application control.

When printer operator control for EPC mode on or off is not provided, the default mode is EPC mode off. When printer operator control is provided, the default is printer operator control enabled; i.e., EPC can be on or off depending on operator selection.

With the EPC mode off, the printer operates in a serial mode. The first time the buffer is loaded and when loading completes, the contents are printed. When the printing completes, the application is signalled print complete. The application then reloads the buffer and starts the print, etc.

The EPC mode permits overlap of the loading and printing which can significantly improve throughput in certain *bulk* print applications. When the EPC mode is set on, the printer is allowed to indicate print complete prior to actual completion provided sufficient buffering is available to accept the next buffer load. When the *early print complete* indication is given is implementation dependent.

The user of the EPC mode must be made aware of the effect on the exception condition reporting and the associated recovery. Several messages may be 'in the pipe' at one time. If an error is detected during the remainder of the printing which occurs after the *early print complete* indication is given, the next or a subsequent transmission will be rejected. Therefore, the sender must be aware that when the EPC mode is in effect, the receipt of a negative response may be the result of either the current or a prior transmission.

Prior to the last transmission (i.e., prior to starting the last print operation) of a job, the application should turn the EPC mode off. The print complete indication for the last buffer printed will then be given after the print operation actually completes.

When the printer operator control (eg. switch) is provided, conflict of control between the printer operator and the host application will be resolved in favor of the host application. The printer operator control is effective in turning the EPC mode on or off until a valid Set Printer Characteristics structured field, with an EPC self defining parameter, is received which disables printer operator control (SREPC = bin.01' or B'10'). The printer operator control will then have no effect on setting the EPC mode on or off until one of the following occurs:

- A valid Set Printer Characteristics structured field with an EPC self defining parameter which has the SREPC flag = enable printer operator control (B'00') is received.
- A valid EW or EWA with WCC = reset is received.
- A BIND (SNA only).
- A POR.

Support of the early print complete function is indicated by the EPC self defining parameter on the Settable Printer Characteristics query reply. This self defining parameter also indicates whether the printer provides operator control for EPC mode on or off and if supported, whether the EPC mode has been set on or off by the operator.

IPDS Data/Non-SNA

This section describes the carrying of the IPDS by the 3270 data stream in a non-SNA environment. For the SNA environment, IPDS data stream is available through LU1.

When the 3270 data stream/IPDS is supported, the printer is in either a 3270 data stream mode or IPDS mode. The 3270 type data and controls and IPDS type data and controls cannot be mixed. In other words, you cannot send 3270-type data and IPDS-type data and then print the composite.

The 3270 IPDS Query Reply indicates the support of IPDS and also defines the maximum transmission size allowed outbound in the IPDS mode. However, the mechanisms provided by the IPDS (for example, STM/ACK REPLY) are used to provide IPDS-related printer characteristics. Also, in the IPDS mode, the exception handling functions provided by the IPDS are used.

Changing modes clears the printer buffers of any data associated with the previous mode. Use of the printer for 3270 local copy is considered to be 3270 mode. For example, if the printer buffer was loaded in the 3270 mode and the printer changed to the IPDS mode, the 3270 data would be cleared. Therefore, a return to the 3270 mode would find a cleared buffer. However, the printer must maintain forms sync across modes as is done between 3270 and SCS. See the 3270 IPDS Query Reply for non-SNA systems on page 6-110 and the Data Streams Query Reply for SNA systems on page 6-42 for more information about IPDS selection.

IPDS Selection

The default (for example, power on) mode is the 3270 mode. In the 3270 mode, the structured fields defined in the IPDS are rejected. When a device is in the IPDS mode, any non-IPDS structured field (except a valid Data Chaining or Select IPDS structured field) is invalid (refer to the IPDS manual for action taken on invalid structured fields). Also, in the IPDS mode, 3270 orders and control sequences are treated as IPDS data with unpredictable results.

Support of IPDS requires that data chaining (for example, the Data Chain structured field and the Data Chaining Query Reply) be supported. See "Structured Field Grouping" on page 5-6.

If the Data Chain structured field is not used in the selection sequence (that is, no data chaining), the IPDS mode exists until the end of the transmission. If the Data Chain structured field is used (that is, data chaining), the IPDS mode exists until the end of the data chain.

The IPDS mode is selected by the Select IPDS Mode structured field. The Select IPDS Mode structured field also resets IPDS controls and conditions to the default values. On outbound transmissions (to the device), the Select IPDS Mode structured field must immediately follow a WSF, or immediately follow a WSF Data Chain structured field (GROUP = begin) sequence. On inbound transmissions (from the device), the Select IPDS Mode must immediately follow an AID X'88' or immediately follow an AID X'88'; that is, a Data Chain structured field (GROUP + begin) sequence.

In data chaining, the Select IPDS Mode structured field only appears immediately after the Data Chain structured field of the first transmission of the data chain. That is, the IPDS mode is continued by the subsequent Data Chain structured fields (GROUP = continue) and terminates at the completion of the last Data Chain structured field (GROUP = end). For example, IPDS structured fields and control sequences can span transmissions. If the printer is not in the IPDS mode, a Select IPDS Mode structured field appearing anywhere, except after a WSF command or WSF Data Chain structured field (GROUP = begin) sequence, is rejected. If the printer is in the IPDS mode, a Select IPDS Mode structured field received anywhere except after a Data Chain structured field (GROUP = begin) sequence is treated as an invalid structured field and handled in accordance with the IPDS.

With the exception of the IPDS ACK structured field which reports exception and status conditions, IPDS is an output-only data stream. Sending of the ACK structured field by the device is limited to either of the following:

- Responding to an IPDS outbound structured field that requests a reply
- Reporting exception conditions that were detected while processing an outbound IPDS transmission.

Local Copy

When in the IPDS mode, the printer cannot be used for operator- initiated local copy. If it is in IPDS mode and the BSC Copy command is received, the IPDS mode is terminated without error and the copy executed.

Pacing

The non-SNA protocols do not provide the chaining and pacing functions provided by SNA. Control of data to the printer in the IPDS mode is accomplished by a combination of data chaining, limiting the transmission size, and link and channel controls.

The 3270 IPDS Query Reply indicates the maximum allowed transmission length during the IPDS mode. The data chaining function allows a message to be divided into transmissions of suitable length without regard to structured field or control boundaries, for example.

The IPDS mode uses the same BSC and channel controls as does the 3270 mode to determine when more data can be sent to the printer. In BSC, a wait before transmit positive acknowledgment (WACK) is returned after acceptance of a transmission if the printer cannot accept more data at this time. When the printer is ready for another transmission, a device end (DE) is returned by the device. In a non-SNA local channel, if the printer cannot accept more data at this time, channel end (CE without DE) is returned after acceptance of a transmission. When the printer is ready for another transmission, a DE is returned to the device.

Input Transmissions

Although the IPDS data stream is essentially an output-only data stream, the printer generates some inbound data in the form of an acknowledgment reply. This reply can be generated asynchronously (exception reporting) or synchronously (in reply to a host request).

Chapter 9. Binary Synchronous Communications (BSC) Environment

Introduction 9-2
Transparent Mode 9-2
Write Commands 9-2
Read Commands 9-2
Read Buffer Command 9-3
Read Modified Command 9-3
Test Request Read 9-4
Inbound Transmissions 9-4
Inbound Operation (INOP) 9-5
Read States 9-5
Normal Read State 9-5
Data-Pending States 9-6
Read-State Transitions 9-6
Retry States 9-6
Indicators 9-7
Read Acknowledgments 9-7
Query or Query List 9-7
Operator Enter or RM, RMA, or RB Partition Command 9-7
Processing of Read Commands 9-9
Processing of Read Partition Query Structured Fields 9-10
BSC Copy Command 9-10

Introduction

This chapter describes how the 3270 data stream operates in a binary synchronous communication (BSC). Except for the differences described in this chapter, the data stream operates the same way as in the SNA environment.

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, because all data, including the normally restricted data link line control characters, are treated as bit patterns when transmitted in transparent mode. This method of transmission is required when transmitting binary counts and addresses that can appear in the data stream for the functions of extended highlighting, color, and programmed symbols.

BSC data links may be 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 as follows:

For nontransparent mode:

STX	3270 Data Stream	ETX

For transparent mode:

DLE	STX	3270 Data Stream	DLE	ETX

Write Commands

The EW and EWA commands operate the same in a BSC environment as in the SNA environment. For the W command, the only difference is that the starting buffer location depends upon the following considerations:

- The starting location can be specified by an SBA order that follows the WCC.
- The starting location is the buffer address containing the cursor if the W command is not chained from a Copy or EAU.
- The starting location is the current buffer address if the W command is chained from a read or another write command.

Read Commands

The read commands operate in the same way 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 as follows:

- 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 W, EW, EWA, Read Modified, or another Read Buffer command. Regardless of where the transfer of data begins, data transfer from the buffer terminates 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 way as in an SNA environment except the buffer location at which the search begins for field attributes that define modified fields is a function of command chaining. This location is one of the following:

- Buffer address 0 if the Read Modified command is unchained or is chained from a Copy.
- The current buffer address if the Read Modified command is chained from a W, EW, 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:

- 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).
- If the buffer does not contain a wrapped modified field, the modified data stream is terminated when the last modified field is transferred. At the end of the operation, the buffer address is set to 0.

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

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

This read operation can also be terminated by the channel byte count's reaching 0 before all data is read. In this case, the buffer address after termination is undefined.

Test Request Read

The Read Modified command causes a test request read operation if the TEST REQ or SYS REQ key has been pressed at the selected device. The test request read data stream sent inbound to the application program is as follows:

SOH
%
/
STX
Input Data
ETX

The Test Request Read heading is generated by the display. The remainder of the data stream is the same as described previously for read-modified operations, excluding the 3-byte read heading (AID and cursor address). If the buffer is unformatted, all the alphanumeric data in the buffer is included in the data stream (nulls suppressed), starting at address 0. If the buffer is formatted, each field-attribute byte is examined for a set MDT bit. Each time a set MDT bit is found, the alphanumeric data in the field associated with that bit is sent to main storage (nulls suppressed). If no MDT bits are set, the read data stream consists of the Test Request Read heading only. The buffer location at which the search for the MDT bits begins and the transfer of data ends is the same as described for the read-modified operations.

Note: Usage of the test request read function is determined by the application program. Normally, the operator would clear the display, enter test request data in a predefined format, and 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 unsolicited read operation is an inbound transmission not caused by an operator enter action. No application program acknowledgment is required before a new inbound transmission can occur.

Host retry is a retransmission of the last unacknowledged inbound transmission from the device. The application program must acknowledge receipt of an inbound transmission before a new inbound transmission can take place.

A host retry transmission does not cause read state transitions and is not considered a new inbound transmission requiring acknowledgment by the application program. Host retry occurs until an application program acknowledgment takes place.

The type of inbound transmission is either a Query Reply structured field (the response to the Read Partition Query structured field) or data from the device buffer (for example, modified fields of the display image). An inbound operation device characteristic (INOP), set by the 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 power is on, a device is in one of seven states with respect to read operations. The three primary states are as follows:

- Normal read
- Data pending read
- Read retry.

The data pending and read retry states have the following three substates:

- Enter
- Read
- Stacked enter.

The events that cause transitions between states are shown in Table 9-1 on page 9-6.

Normal Read State

A device is in normal read state when power is on, prior to initiation of a new read operation, and after use of the Reset key in certain instances. (See "Read-State Transitions" on page 9-6.)

When in normal read state an operator enter action or the reception of a Read Partition Query structured field causes the device to prepare to generate the inbound data stream and to go into a data pending state.

When in normal read state, an application program initiated read operation using an RM, RMA, or RB command causes the data to be transmitted with no state transitions occurring. The device remains in normal read state.

Data-Pending States

The three data pending states are as follows:

Enter	The device state after an operator enter action
Read	The device state after reception of a Read Partition Query structured field
Stacked enter	The device state after a Read Partition Query structured field was received while the device was in data pending enter read state or retry enter state (the enter is stacked).

A poll received while the device is in a data pending state causes the data to be transmitted and the device to be placed in the corresponding retry state.

Read-State Transitions

The read state transitions for BSC are summarized in Table 9-1.

Table 9-1. Read State Transitions for BSC

Read States		Data Pending			Retry		
Events	**Normal**	**Enter**	**Read**	**Stacked Enter**	**Enter**	**Read**	**Stacked Enter**
Enter Action	2	R	R	R	R	R	R
Read Command	1	1	1	1	G	G	G
Read Partition Query	3	4	R	R	4	R	R
Poll	—	5	6	7	—	—	—
Host Acknowledge	—	1	1	2	1	1	2
Reset key	—	1	—	3	1	—	6

Legend:

R	Reject, no state transition
G	Retry, no state change
—	No action or state change
1	Normal read state
2	Data pending enter state
3	Data pending read state
4	Data pending stacked enter state
5	Retry enter state
6	Retry read state
7	Retry stacked enter state

Retry States

There are three retry states:

Enter	The device state after enter data was transmitted to the application program
Read	The device state after Query Reply data was transmitted to the application program

Stacked enter — The device state after enter data was stacked and the Query Reply data transmitted to the application program.

While in retry state, the last inbound transmission can be retried by means of a Read Modified command.

A host acknowledgment causes the device to revert from a retry state to the normal read state or, in the case of retry stacked enter, to the data pending stacked enter read state.

Indicators

With reference to Table 9-1 on page 9-6, the indicators displayed in the operator information area of a display are as follows:

State	**Indicator**
Normal read	No indicator or 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.

Read Acknowledgments

Read acknowledgements depend on the inbound operation. The following sections describe the read acknowledgements when the inbound operation is a Query, Query List, Operator Enter, RM, RMA, or RB Partition command.

Query or Query List

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

The following commands acknowledge a Query or Query List:

- A WSF command with or without following structured fields. The transmission is an acknowledgment regardless of an error being detected in the accompanying structured fields, as long as the WSF is accepted.
- An EAU command.
- An EW, EWA, or W with or without a WCC or data. If data is present and an error is detected in the data, the transmission is *not* an acknowledgment.

Operator Enter or RM, RMA, or RB Partition Command

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

Any of the following constitutes an acknowledgment when the display is in any of the retry or data pending states:

- If the inbound partition is 0 (implicit or explicit), either of the following:
  - An EW, EWA, or Write command with WCC = Keyboard Restore (see note 1).
  - An EAU command.
- If the display is in the explicit partition state, an EW or EWA command with the WCC = reset (see note 1).
- A WSF command followed by an outbound 3270DS structured field to the inbound partition with either of the following (see note 2):
  - An EW, EWA, or Write partition command with WCC = Keyboard Restore
  - EAU partition command.
- A WSF command followed by a Destroy Partition structured field to the inbound partition, including explicit and implicit partition 0 (see note 2).
- A WSF command followed by a Create Partition structured field to the inbound partition (see note 2).
- If in implicit partition state, a WSF command followed by a Create Partition structured field (see note 2).
- A WSF command followed by an Erase/Reset structured field. Applies to both implicit and explicit partition state (see note 2).
- A Copy command (BSC only).

In addition, the following constitute a read acknowledgment only when the display is in one of the retry states:

- A WSF command followed by an outbound 3270 DS structured field, with an EW, EWA, W or EAU partition command, directed to any partition. This acknowledgement applies whether or not the EW, EWA, or W partition command is followed by a WCC or data (see note 2).
- An EW, EWA, Write, or EAU command. This acknowledgement applies whether or not a WCC or data follows the EW, EWA or W command (see note 1).

Notes:

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

Processing of Read Commands

In a BSC environment, a read command (Read Buffer, Read Modified, or Read Modified All) as the first byte of the data stream is processed as follows:

- If the device is in normal read state, then the command performs a read and the display data is transmitted inbound as defined by the following:

  1. The Read Modified, Read Modified All, or Read Buffer command
  2. The AID (Read Modified command only)
  3. The reply mode.

  The device remains in normal read state.

- If the device is in a data pending state, then the command performs a read and data is transmitted as defined by:

  1. The Read Modified, Read Modified All, or Read Buffer command
  2. The AID (Read Modified command only)
  3. The inbound reply mode.

  The device is placed in normal read state.

- If the device is in a retry state, then the command performs a retry as follows:

  1. If the command is Read Modified, and INOP specifies Query, then appropriate Query Replies are transmitted inbound.
  2. If the command is Read Modified, and INOP specifies Read Modified, data is transmitted as defined by the following:

     a. The Read Modified command
     b. The AID
     c. The inbound reply mode.

  3. If the command is Read Buffer, data is transmitted inbound as defined by the following:

     a. The Read Buffer command
     b. The inbound reply.

  The device remains in the original retry state.

Processing of Read Partition Query Structured Fields

Read Partition Query and the Query Reply structured fields are processed as follows:

If the device is in normal read state, the following occurs:

1. The TWAIT indicator is displayed.
2. INOP is set to Query.
3. The device prepares to generate the required inbound data stream.
4. The device is placed in data pending read state.
5. A later poll causes the data to be transmitted and the device to be placed in retry read state.

If the device is in data pending enter or retry enter state, the following occurs:

1. The outstanding data is stacked.
2. The TWAIT condition remains in effect.
3. INOP is set to Query.
4. The device prepares to generate the required inbound data stream.
5. The device is placed in data pending stacked enter state.
6. 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 the Local Copy function. See Chapter 8, "Printer Operations" for a detailed description of the operation of the Copy command.